

Advanced Function Presentation Consortium
Data Stream and Object Architectures

Graphics Object Content Architecture for Advanced Function Presentation Reference

AFPC-0008-03



AFPC Consortium
Advanced Function Presentation

Note:

Before using this information, read the information in [“Notices” on page 205](#).

**AFPC-0008–03
Fourth Edition (April 2017)**

This edition applies to the Graphics Object Content Architecture for Advanced Function Presentation™. This is the second edition produced by the AFP Consortium™ (AFPC™) and replaces and makes obsolete the previous edition (AFPC-0008–02). In AFP™ environments, this document makes obsolete the *IBM® Graphics Object Content Architecture Reference*, SC31-6804. This edition remains current until a new edition is published.

Technical changes are indicated in green, with a green vertical bar to the left of the change. Editorial changes that have no technical significance are not noted. For a detailed list of changes, see [“Summary of Changes” on page ix](#).

Internet

Visit our home page: www.afpcinc.org

Preface

This book describes the functions and services associated with the Graphics Object Content Architecture (GOCA) for Advanced Function Presentation (AFP). AFP GOCA defines a version of the GOCA architecture that is used in Advanced Function Presentation environments.

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

Who Should Read This Book

This book is for systems programmers and other developers who develop or adapt products or programs to interoperate with other presentation products in an AFP environment.

AFP Consortium (AFPC)

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

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

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

How to Use This Book

This book is divided into nine chapters and four appendixes:

- [Chapter 1, “A Presentation Architecture Perspective”](#) introduces Graphics Object Content Architecture for Advanced Function Presentation and positions it as a strategic Object Content Architecture.
- [Chapter 2, “Introduction to GOCA and AFP GOCA”](#) contains an introduction to the Graphics Object Content Architecture (GOCA) and to the version of GOCA used in Advanced Function Presentation environments.
- [Chapter 3, “AFP GOCA Overview”](#) includes the following concepts:
 - The graphics processor (GP)
 - The environment interface
 - The drawing processor
 - Graphics coordinate spaces
 - Color
 - Mix
 - Segments
 - Subsetting
 - Exception conditions
- [Chapter 4, “Graphics Primitives and Attributes”](#) describes the various primitive drawing operations and the attributes used to control them.
- [Chapter 5, “Segments”](#) describes the structuring of the object into independent pieces (segments) that can be chained together to form the picture.
- [Chapter 6, “Environment Controls”](#) describes how the environment communicates with the graphics processor to draw the graphics picture. It also describes the drawing process controls that can be set by the environment, and the control instructions used.
- [Chapter 7, “Commands and Drawing Orders”](#) provides a detailed listing of the commands and drawing orders, together with a description of their parameters and the exception conditions that can arise.
- [Chapter 8, “Exception Conditions”](#) provides a detailed listing of exception conditions.
- [Chapter 9, “Compliance”](#) defines the subsets currently supported in AFP GOCA.
- [Appendix A, “Mixed Object Document Content Architecture \(MO:DCA\) Environment”](#) describes how AFP GOCA fits into the Mixed Object Document Content Architecture™ (MO:DCA™) environment.
- [Appendix B, “Intelligent Printer Data Stream \(IPDS\) Environment”](#) describes how AFP GOCA fits into the Intelligent Printer Data Stream™ (IPDS™) environment.
- [Appendix C, “AFP GOCA Migration Functions”](#) provides the syntax and semantics for AFP GOCA retired drawing orders and parameters.
- [Appendix D, “Cross-References”](#) provides tables of AFP GOCA commands, control instructions, and drawing orders sorted by identifier and by name.

This publication also contains a [Glossary](#) and [Index](#).

How to Read the Syntax Diagrams

Throughout this book, syntax is described using the structure defined below. 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

Syntax for Graphics Object Content Architecture (GOCA) is shown in tables.

Offset	Type	Name	Range	Meaning
The field's offset	Data type, if applicable	Name of field, if applicable	Range of valid values, if applicable	Meaning or purpose of the data element

Offset values specify the byte offset of the field in the table, or the bit offset within a field of BITS data type.

The bits are specified with bit 0 meaning the most significant bit.

Multi-byte fields of fixed length are shown as having two offsets, those of the first and last bytes of the field. For example, 4–7 indicates a field of length four bytes.

If a field is an array of varying length, then the offset of the last byte of the field is shown as n , for example, 2– n .

A blank entry in the range column indicates that there are no restrictions on the acceptable values.

Certain fields may be denoted as *reserved*. A reserved field is a parameter that has no functional definition at the current time, but may have at some time in the future. All bytes comprising a field, defined by the AFP GOCA architecture as a reserved field, should be given the defined, reserved value by generating applications, and should be ignored by receiving applications.

The following example shows the syntax of the Begin Image at Given Position (GBIMG) order.

Offset	Type	Name	Range	Meaning
0	CODE		X'D1'	GBIMG order code
1	UBIN	LENGTH	X'0A'	Length of following data
2–3	SBIN	XPOS	X'8000'–X'7FFF'	X _g coordinate of image origin (first image point of first image scan line)
4–5	SBIN	YPOS	X'8000'–X'7FFF'	Y _g coordinate of image origin (first image point of first image scan line)
6	CODE	FORMAT	X'00'	Format of the image data: X'00' Each image point is mapped to a presentation device pel
7		RES	X'00'	Reserved; only valid value
8–9	UBIN	WIDTH	X'0000'–X'FFFF'	Width of the image data, in image points
10–11	UBIN	HEIGHT	X'0000'–X'FFFF'	Height of the image data, in rows, or scan lines

How to Use This Book

Notation Used In Formulas

When formulas are used in this book, \cdot is the notation for multiplication, and \div the notation for division. For example:

$a \cdot b$ indicates a multiplied by b

$a \div b$ indicates a divided by b

Related Publications

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

Table 1. AFP Consortium Architecture References

AFP Architecture Publication	Order Number
<i>AFP Programming Guide and Line Data Reference</i>	S544-3884 (IBM)
<i>Bar Code Object Content Architecture™ Reference</i>	AFPC-0005
<i>Color Management Object Content Architecture Reference</i>	AFPC-0006
<i>Font Object Content Architecture Reference</i>	AFPC-0007
<i>Graphics Object Content Architecture for Advanced Function Presentation Reference</i>	AFPC-0008
<i>Image Object Content Architecture Reference</i>	AFPC-0003
<i>Intelligent Printer Data Stream Reference</i>	AFPC-0001
<i>Metadata Object Content Architecture Reference</i>	AFPC-0013
<i>Mixed Object Document Content Architecture (MO:DCA) Reference</i>	AFPC-0004
<i>Presentation Text Object Content Architecture Reference</i>	AFPC-0009

Table 2. Additional AFP Consortium Documentation

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

Table 3. Additional Graphics Object Documentation

Publication	Order Number
<i>Graphics Object Content Architecture Reference</i>	SC31-6804 (IBM)

Related Publications

Table 4. AFP Font-Related Documentation

Publication	Order Number
<i>Character Data Representation Architecture Reference and Registry</i> ; please refer to the online version for the most current information (http://www-306.ibm.com/software/globalization/cdra/index.jsp)	SC09-2190 (IBM)
<i>Font Summary for AFP Font Collection</i>	S544-5633 (IBM)
<i>Technical Reference for Code Pages</i>	S544-3802 (IBM)

Summary of Changes

This fourth edition of the *Graphics Object Content Architecture for Advanced Function Presentation Reference* contains the following significant architecture extensions:

- New drawing orders:
 - Begin Custom Pattern and End Custom Pattern drawing orders, for defining custom patterns to be used to fill an area; custom patterns are arbitrary patterns and add to the existing set of pre-defined patterns in the default pattern set
 - Set Pattern Reference Point drawing order, for defining the origin for placement of custom patterns in an area
 - Linear Gradient and Radial Gradient drawing orders, for defining gradients to be used to fill an area
 - Delete Pattern drawing order, for deleting previously defined custom patterns or gradients
 - Set Custom Line Type drawing order, for defining custom line types to be used when drawing lines; custom line types are arbitrary sequences of dashes and moves
- Additions to the Set Current Defaults instruction:
 - Marker cell-size attribute
 - Pattern reference point attribute
- New Nonzero Winding Mode method to determine what is considered to be in an area's interior; this is in addition to the existing Alternate Mode
- Reinstatement (from IBM GOCA) of the use of the marker cell-size attribute; using the Set Marker Cell drawing order is now supported for setting the size that markers will appear (the drawing order was treated as a No-Op in previous editions of AFP GOCA)
- Reinstatement (from IBM GOCA) of the ability to draw boxes in a clockwise direction
- Reinstatement (from IBM GOCA) of the ability to set the pattern reference point attribute
- Pattern sets X'01'–X'FD' are now supported for custom patterns or gradients
- The marker precision attribute has been made obsolete, along with the Set Marker Precision drawing order and the EC-C202 exception condition
- Many new exceptions having to do with the new custom pattern and gradient functionality
- Exception condition EC-0008 now has more ways it can occur, due to the first extended format drawing orders in AFP GOCA, the Linear Gradient and Radial Gradient drawing orders
- Retirement of the EC-0002 exception condition
- Many updates to the Glossary, to more closely match other AFP publications
- Many small updates to increase consistency and readability

Technical changes between this edition and the previous edition are marked in green and have a green vertical bar (|) in the left margin.

Contents

Preface	iii
Who Should Read This Book	iii
AFP Consortium (AFPC)	iii
How to Use This Book	iv
How to Read the Syntax Diagrams	v
Notation Used In Formulas	vi
Related Publications	vii
Summary of Changes	ix
Figures	xvii
Tables	xix
Chapter 1. A Presentation Architecture Perspective.	1
The Presentation Environment	1
Architecture Components	2
Data Streams	2
Objects	4
Chapter 2. Introduction to GOCA and AFP GOCA.	7
Background	7
Scope of GOCA	7
Concepts of GOCA	7
Chapter 3. AFP GOCA Overview	9
The Graphics Processor Model	9
The Environment Interface	11
Drawing Processor	11
Primitives	11
Drawing Orders	12
Attributes	12
Graphics Coordinate Spaces	13
Drawing Order Coordinate Space (DOCS)	13
Graphics Presentation Space (GPS)	13
Usable Area (UA)	14
Color	14
Mix	16
Segments	17
Subsetting	17
Exception Conditions	17
Chapter 4. Graphics Primitives and Attributes	19
Output Primitives	19
Current Position	19
Symbols	20
Line Primitives	22
Straight Lines	22
Curved Lines	23
Boxes	28
Line Attributes	29
Areas	36
Valid Area Definitions	38
Patterns	39
Area (Pattern) Attributes	50
Character Strings	51
Device-Specific (Character) Precision	53
Device-Specific (String) Precision	56
Character Attributes	56
Markers	57

Marker Attributes	58
Images.....	59
Image Attributes	59
Output Primitive Overflow.....	60
Chapter 5. Segments	61
Segment Types.....	61
Segment Processing Sequence.....	61
Segment Properties	62
Segment Prolog	62
Segment Prolog Semantics	63
Chapter 6. Environment Controls	65
Control Instructions.....	65
Set Current Defaults (SCD) Instruction.....	66
Drawing Processor Facilities.....	70
Current Attributes.....	70
Drawing Process Controls.....	70
Chapter 7. Commands and Drawing Orders	73
Begin Segment Command	75
Syntax	75
Semantics	75
Exception Conditions	76
Order Formats	77
Fixed 1-Byte Format	78
Fixed 2-Byte Format	78
Long Format	78
Extended Format	78
Order Alignment.....	79
Current Position in Drawing Orders	79
Coordinate Data.....	79
Offset Data	79
Default Values	79
Summary List of Orders	80
Begin Area (GBAR) Order	82
Syntax	82
Semantics	82
Exception Conditions	83
Begin Custom Pattern (GBCP) Order.....	84
Syntax	84
Semantics	84
Exception Conditions	85
Begin Image (GBIMG, GCBIMG) Orders	87
Syntax	87
Semantics	87
Exception Conditions	89
Box (GBOX, GCBOX) Orders.....	90
Syntax	90
Semantics	90
Exception Conditions	91
Character String (GCHST, GCCHST) Orders.....	92
Syntax	92
Semantics	92
Exception Conditions	93
Comment (GCOMT) Order	95
Syntax	95
Semantics	95
Exception Conditions	95
Cubic Bezier Curve (GCBEZ, GCCBEZ) Orders.....	96
Syntax	96
Semantics	97
Exception Conditions	98

Delete Pattern (GDPT) Order	99
Syntax	99
Semantics	99
Exception Conditions	99
End Area (GEAR) Order	101
Syntax	101
Semantics	101
Exception Conditions	101
End Custom Pattern (GECF) Order	102
Syntax	102
Semantics	102
Exception Conditions	102
End Image (GEIMG) Order	103
Syntax	103
Semantics	103
Exception Conditions	103
End Prolog (GEPROL) Order	104
Syntax	104
Semantics	104
Exception Conditions	104
Fillet (GFLT, GCFLT) Orders	105
Syntax	105
Semantics	105
Exception Conditions	106
Full Arc (GFARC, GCFARC) Orders	107
Syntax	107
Semantics	107
Exception Conditions	108
Image Data (GIMD) Order	109
Syntax	109
Semantics	109
Exception Conditions	109
Line (GLINE, GCLINE) Orders	110
Syntax	110
Semantics	110
Exception Conditions	111
Linear Gradient (GLGD) Order	112
Syntax	112
Semantics	113
Exception Conditions	115
Marker (GMRK, GCMRK) Orders	116
Syntax	116
Semantics	116
Exception Conditions	117
No-Operation (GNOP1) Order	118
Syntax	118
Semantics	118
Exception Conditions	118
Partial Arc (GPARC, GCPARC) Orders	119
Syntax	119
Semantics	119
Exception Conditions	120
Radial Gradient (GRGD) Order	122
Syntax	122
Semantics	123
Exception Conditions	126
Relative Line (GRLINE, GCRLINE) Orders	127
Syntax	127
Semantics	127
Exception Conditions	128
Segment Characteristics (GSGCH) Order	129
Syntax	129
Semantics	129

Exception Conditions	129
Set Arc Parameters (GSAP) Order	130
Syntax	130
Semantics	130
Exception Conditions	131
Set Background Mix (GSBMX) Order	132
Syntax	132
Semantics	132
Exception Conditions	132
Set Character Angle (GSCA) Order	133
Syntax	133
Semantics	133
Exception Conditions	133
Set Character Cell (GSCC) Order	134
Syntax	134
Semantics	134
Exception Conditions	135
Set Character Direction (GSCD) Order	136
Syntax	136
Semantics	136
Exception Conditions	137
Set Character Precision (GSCR) Order	138
Syntax	138
Semantics	138
Exception Conditions	139
Set Character Set (GSCS) Order	140
Syntax	140
Semantics	140
Exception Conditions	140
Set Character Shear (GSCH) Order	141
Syntax	141
Semantics	141
Exception Conditions	141
Set Color (GSCOL) Order	142
Syntax	142
Semantics	142
Exception Conditions	142
Set Current Position (GSCP) Order	143
Syntax	143
Semantics	143
Exception Conditions	143
Set Custom Line Type (GSCLT) Order	144
Syntax	144
Semantics	144
Exception Conditions	145
Set Extended Color (GSECOL) Order	146
Syntax	146
Semantics	146
Exception Conditions	146
Set Fractional Line Width (GSFLW) Order	147
Syntax	147
Semantics	147
Exception Conditions	147
Set Line End (GSLE) Order	148
Syntax	148
Semantics	148
Exception Conditions	148
Set Line Join (GSLJ) Order	149
Syntax	149
Semantics	149
Exception Conditions	149
Set Line Type (GSLT) Order	150

Syntax	150
Semantics	150
Exception Conditions	150
Set Line Width (GSLW) Order	151
Syntax	151
Semantics	151
Exception Conditions	151
Set Marker Cell (GSMC) Order	152
Syntax	152
Semantics	152
Exception Conditions	153
Set Marker Set (GSMS) Order	154
Syntax	154
Semantics	154
Exception Conditions	154
Set Marker Symbol (GSMT) Order	155
Syntax	155
Semantics	155
Exception Conditions	155
Set Mix (GSMX) Order	156
Syntax	156
Semantics	156
Exception Conditions	156
Set Pattern Reference Point (GSPRP) Order	157
Syntax	157
Semantics	157
Exception Conditions	157
Set Pattern Set (GSPS) Order	158
Syntax	158
Semantics	158
Exception Conditions	158
Set Pattern Symbol (GSPT) Order	159
Syntax	159
Semantics	159
Exception Conditions	160
Set Process Color (GSPCOL) Order	161
Syntax	161
Semantics	161
Exception Conditions	164
Chapter 8. Exception Conditions	167
Set Current Defaults Instruction Exceptions	167
Begin Segment Command Exceptions	167
Drawing Order Exceptions	168
Exception Conditions without Standard Actions	169
Exception Conditions with Standard Actions	171
Chapter 9. Compliance	175
AFP GOCA Subsets	175
Base (Mandatory) Level (Version 0)	175
Drawing Order Level 2, Version 0 (DR/2V0)	176
Graphics Subset Level 3 (GRS3)	178
Appendix A. Mixed Object Document Content Architecture (MO:DCA) Environment	179
Compliance with MO:DCA Interchange Sets	179
Graphics Structured Fields in the MO:DCA Environment	180
Graphics Data Descriptor (GDD) in the MO:DCA Environment	180
GDD Self-Identifying Parameters	181
Graphics Data (GAD) in the MO:DCA Environment	188
GOCA Subsets within the MO:DCA Environment	188
Appendix B. Intelligent Printer Data Stream (IPDS) Environment	189
Graphics in the IPDS Environment	189

IPDS Graphics Command Set	189
Write Graphics Control Command	189
Write Graphics Command	191
Additional Related Commands	191
IPDS Exceptions	192
Appendix C. AFP GOCA Migration Functions	195
Introduction	195
General	195
Migration Functions	195
Obsolete Functions	195
Obsolete Attributes	195
Obsolete Drawing Orders	196
Obsolete Exception Conditions	197
Retired Functions	197
Retired Parameters	197
Retired Exception Conditions	197
Appendix D. Cross-References	199
AFP GOCA Commands Sorted by Identifier	199
AFP GOCA Commands Sorted by Acronym	199
AFP GOCA Control Instructions Sorted by Identifier	199
AFP GOCA Control Instructions Sorted by Acronym	199
AFP GOCA Drawing Orders Sorted by Identifier	200
AFP GOCA Drawing Orders Sorted by Acronym	202
Notices	205
Trademarks	206
Glossary	207
Index	225

Figures

1. Presentation Environment	1
2. Presentation Model	3
3. Presentation Page	5
4. The Graphics Processor (GP) Within the Controlling Environment	10
5. Coordinate System Used for the GPS	14
6. Arc Parameters	23
7. Full Arc	24
8. Partial Arc	25
9. Fillet	26
10. Cubic Bezier Curve	27
11. Box	28
12. Example of the Dash-dot Line Type	31
13. Custom Line Type with Small Move Value (with Different Line Ends)	31
14. Line End Shapes	34
15. Line Join Shapes	35
16. Miter Line Joins in Combination with Line Types and Line Ends	35
17. Closed and Open Figures	36
18. Determining the Interior of an Area	37
19. Default Pattern Set	40
20. An Example of Both a Linear Gradient and a Radial Gradient	43
21. Linear Gradient Extending to the Edge of the GPS	44
22. Effect of Different Outside Values	45
23. A Yellow Color Stop at Offset 60% Added to the Gradient from Figure 21	46
24. A Simple Radial Gradient	47
25. A Radial Gradient when the Start and End Full Arcs Overlap	47
26. A Radial Gradient when the Start and End Full Arcs Are Outside Each Other	47
27. The Radial Gradient from Figure 25 when Both Outside Values Are Pad	48
28. The Radial Gradient from Figure 24 when Both Outside Values Are Pad	48
29. Methods for Positioning Character Strings	52
30. Font Positioning Method	54
31. Cell Positioning Method	55
32. Default Marker Set	57
33. Attributes and Drawing Process Control	70
34. Fixed 1-Byte Order Format	78
35. Fixed 2-Byte Order Format	78
36. Long Order Format	78
37. Extended Order Format	78

Tables

1. AFP Consortium Architecture References	vii
2. Additional AFP Consortium Documentation	vii
3. Additional Graphics Object Documentation	vii
4. AFP Font-Related Documentation	viii
5. Standard Color Table	15
6. Foreground/Background in Graphics Presentation Space	16
7. Setting Attributes for Character, Marker, and Pattern Symbols	21
8. Attributes Controlling Line Primitives	29
9. Move Type Orders	30
10. Attributes Controlling Area Primitives	50
11. Attributes Controlling Character String Primitives	56
12. Attributes Controlling Marker Primitives	58
13. Attributes Controlling Image Primitives	59
14. Valid Values for UTF-8 First and Second Bytes	93
15. Mapping from GOCA Exception Condition to IPDS Exception	192
16. Commands Sorted by ID	199
17. Commands Sorted by Acronym	199
18. Control Instructions Sorted by ID	199
19. Control Instructions Sorted by Acronym	199
20. Drawing Orders Sorted by ID	200
21. Drawing Orders Sorted by Acronym	202

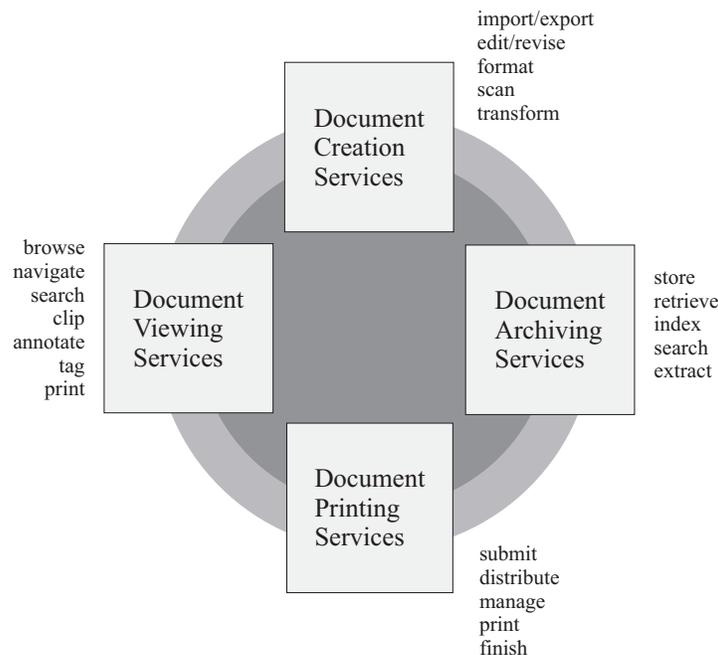
Chapter 1. A Presentation Architecture Perspective

This chapter provides a brief overview of Presentation Architecture.

The Presentation Environment

[Figure 1](#) shows today's presentation environment.

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



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

The solution is a presentation architecture base that is both robust and open ended, and easily adapted to accommodate the growing needs of the open system environment. AFP 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 might be part of an integrated system solution or they might be totally separated from one another in time and space. AFP presentation architectures provide structures that support object-oriented models and client/server environments.

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

Architecture Components

AFP 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 can contain combinations of text, image, graphics, and bar code objects in device-independent and resolution-independent formats. Documents can contain fonts, overlays, and other resource objects required at presentation time to present the data properly. Finally, documents can contain resource objects, such as a document index and tagging elements supporting the search and navigation of document data, for a variety of application purposes.

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

Data Streams

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

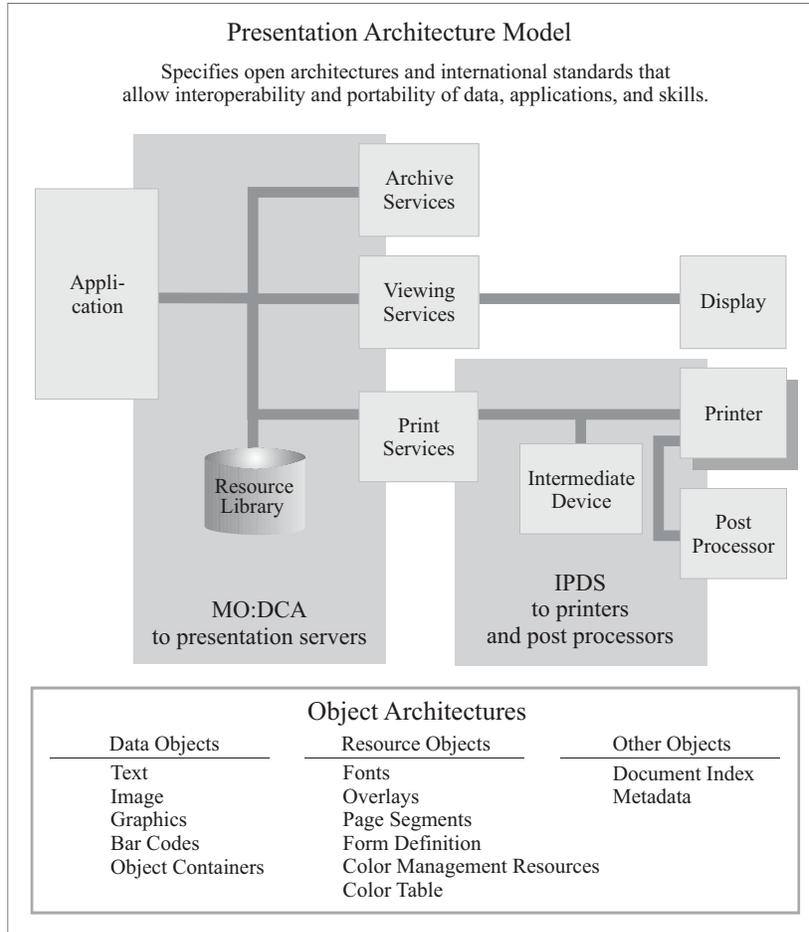
- *Mixed Object Document Content Architecture (MO:DCA)*
- *Intelligent Printer Data Stream (IPDS) Architecture*

The MO:DCA architecture defines the data stream used by applications to describe documents and object envelopes for interchange with other applications and application services. Documents defined in the MO:DCA format can 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.

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

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



Presentation Architecture

Objects

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

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

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

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

The AFPC-defined object content architectures are:

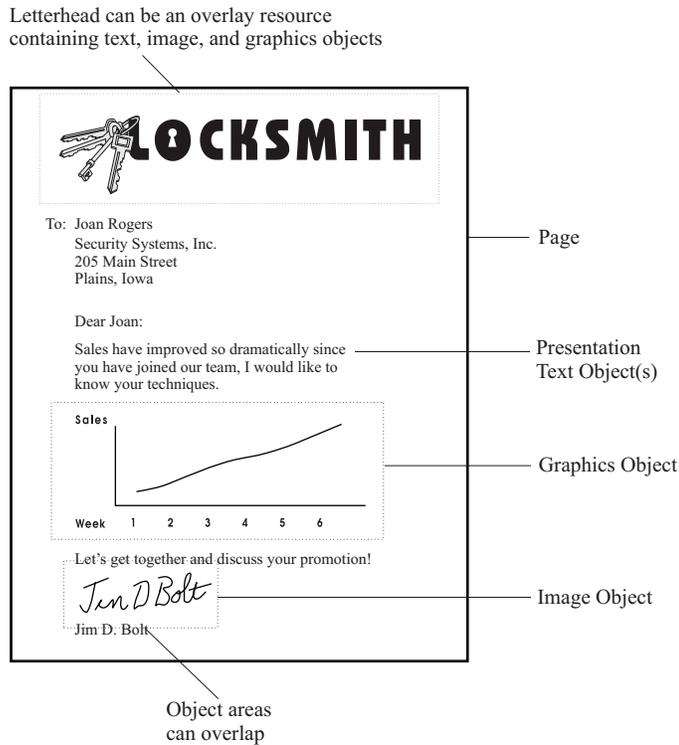
- *Presentation Text Object Content Architecture (PTOCA)*: A data architecture for describing text objects that have been formatted for all-points-addressable presentations. Specifications of fonts, text color, and other visual attributes are included in the architecture definition.
- *Image Object Content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

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

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

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

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



Chapter 2. Introduction to GOCA and AFP GOCA

This chapter covers:

- Background of Computer Graphics
- Scope of GOCA
- Concepts of GOCA

Background

The generation of pictures by computer, called *computer graphics*, has been an application area for many years. However, computer graphics is no longer the specialized concern of large businesses using expensive hardware and consuming vast programming and computing resources. Applications using computer graphics are now readily available for small businesses and the home.

Scope of GOCA

In general, the term *computer graphics* refers to the definition and representation of graphics elements used to build pictures for presentation, either on hard-copy devices such as printers and plotters, or on soft-copy devices such as vector or raster displays. *Interactive computer graphics* refers to the creation and manipulation of these composed pictures using end-user input devices such as a tablet, joystick, or mouse.

GOCA is an object architecture used to represent pictures generated by computer.

This document defines the version of the GOCA architecture that is supported in Advanced Function Presentation (AFP) environments for printing and viewing.

Typically, pictures are built from many different kinds of *primitives*, such as:

- Lines or arcs
- Characters or symbols
- Shaded areas or point arrays

Each of these primitives has its own particular set of *primitive attributes*, such as:

- Line width or line type
- Orientation or direction
- Shading pattern or resolution

In addition, there is usually a set of controls, such as environment-defined defaults, that apply to all primitives.

GOCA is concerned with the creation and manipulation of pictures built by direct invocation of the above primitives and attributes. Thus, GOCA is restricted to the creation and modification of what is generally termed *vector*, or *line-drawn*, graphics. However, additional architectures can be built on top of GOCA for creating and manipulating more complex constructs such as graphs, histograms, and pie charts.

Concepts of GOCA

GOCA effectively defines a graphics subsystem that can exist in, or be invoked by, a number of environments. Each of these controlling environments can be specialized for a particular application area. AFP GOCA is the version of GOCA used to present and interchange graphics pictures in AFP environments. See [Appendix A, “Mixed Object Document Content Architecture \(MO:DCA\) Environment”, on page 179](#), and [Appendix B, “Intelligent Printer Data Stream \(IPDS\) Environment”, on page 189](#) for details of these environments.

Introduction

Presentation of a graphics picture is facilitated by partitioning it into segments, which are effectively independent subpictures. The picture is defined by means of drawing orders that draw primitives, or set attributes and controls that determine the appearance of those primitives.

The picture is defined in a Graphics Presentation Space (GPS) that is independent of the environment. The mapping of this picture onto the presentation surface of a device is defined by a descriptor. The descriptor depends on the environment.

Chapter 3. AFP GOCA Overview

This chapter gives an overview of AFP GOCA, and describes:

- The concept of the graphics processor
- The environment interface
- The drawing processor, including:
 - Primitives
 - Drawing orders
 - Attributes
- Graphics coordinate spaces
- Color
- Mix
- Segments
- Subsetting
- Exception Conditions

The Graphics Processor Model

GOCA is based on the concept of a graphics processor (GP). This processor is embedded into different controlling environments. Some typical controlling environments are:

- Intelligent Printer Data Stream (IPDS) printers with graphics capability
- Mixed Object Document Content Architecture (MO:DCA) data streams for interchange

AFP GOCA deals with GOCA objects that are created, interchanged, archived, and presented within these two controlling environments.

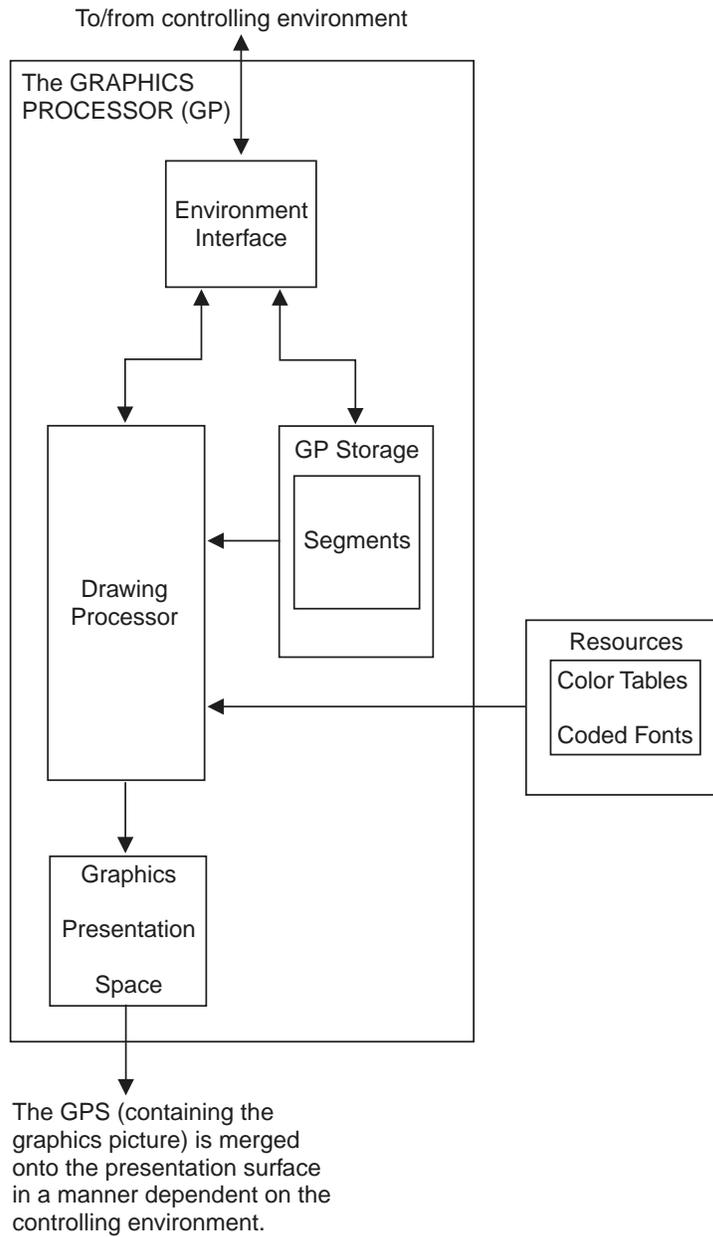
The graphics processor contains the following processing components:

- Environment interface
- Drawing processor

Graphics Processor Model

[Figure 4](#) shows the components and connections of the graphics processor.

Figure 4. The Graphics Processor (GP) Within the Controlling Environment



The Environment Interface

The environment interface performs the functions required to interface the graphics processor with the controlling environment and is responsible for examining the data passed to it from the controlling environment.

Three types of control structures are passed from the controlling environment to the graphics processor via the environment interface:

- *Commands*. The only command supported in AFP GOCA is the Begin Segment (X'70') command, which is used to define a segment.
- *Control Instructions*. The only control instruction supported in AFP GOCA is the Set Current Defaults (X'21') instruction, which sets the current default values of selected attributes.
- *Drawing Orders*. These orders comprise the segment data. They generate graphics primitives in the Graphics Presentation Space (GPS) and set their attributes.

Drawing Processor

The graphics picture is drawn in the GPS by the drawing processor, which executes a sequence of drawing orders. The drawing processor is started by the controlling environment, which in AFP GOCA is the MO:DCA or IPDS environment.

Drawing orders whose execution affects the GPS are called primitive drawing orders. These orders cause the designated primitive to be mixed into the GPS. Additional drawing orders set drawing attributes. All drawing orders are sometimes simply referred to as orders.

Primitives

A primitive is the smallest portion of a picture that can be drawn. There are six types of primitive with their associated set of color and mix attributes:

- Line primitives
- Area primitives
- Character string primitives
- Marker primitives
- Pattern primitives
- Image primitives

A primitive is defined by:

- The parameters of a primitive drawing order
- Modal parameters called attributes
- Control instructions that contain the Set Current Defaults instruction, such as the MO:DCA Graphics Data Descriptor (GDD) and the IPDS Write Graphics Control (WGC)

Modal parameters have values initialized by the environment and can be altered by attribute-setting drawing orders or by control instructions. Modal parameter values persist until they are explicitly altered, or until the end of the graphics object is encountered.

Drawing Orders

Drawing orders are defined for each of the following types of output primitive:

Line	Line primitives: Line and Relative Line One or more straight lines connected together. Full Arc A full circle or ellipse. Partial Arc A line from a point to the start of an arc, then a portion of a full arc. Fillet A curved line drawn tangentially to a specified set of contiguous straight lines. Cubic Bezier Curve A third-order curve defined by a group of four points. Box A rectangle drawn with square or round corners.
Area	One or more closed figures that can be filled. The closed figures can overlap.
Character String	A series of characters drawn along a baseline starting at a specified point.
Marker	A symbol positioned by its center, and drawn at one or more points.
Pattern	A symbol that is repeated to fill an area.
Image	A rectangular area containing a set of foreground and background points.

A summary list of the Drawing Orders is given in ["Summary List of Orders" on page 80](#).

Attributes

Primitive attributes specify the characteristics of the output primitives that define the picture to be drawn.

The following types of attribute are defined:

- Drawing attributes
- Line attributes
- Character attributes
- Marker attributes
- Pattern attributes

Drawing Attributes

Color	The color in which the foreground bits of the output primitive are to be drawn.
Mix	Affects how the foreground of the output primitive that is being drawn is to be merged with the color information already in the GPS.
Background Mix	Affects how the background of the output primitive that is being drawn is to be merged with the color information already in the GPS.

Sets of mix and color attributes are provided for each type of primitive.

Line Attributes

Line Type	The type of line to be drawn; for example, solid or dashed.
Line Width	The width of line to be drawn; for example, normal or wide.
Line End	The type of ending of stroked lines; for example, flat, square, or round.
Line Join	The type of joining of stroked lines; for example, round, bevel, or miter.

Character Attributes

Character Precision	The requested appearance fidelity of a character string.
Character Shear	The amount of slope of a character string. This attribute is not supported in AFP GOCA.
Character Angle	The angle between the character baseline and the GPS X_g axis. Only values of 0° , 90° , 180° , and 270° are supported in AFP GOCA.
Character Cell	The size of the cell in which a character is drawn.
Character Direction	The direction in which characters are drawn.
Character Set	The set of symbols from which characters are obtained.

Marker Attributes

Marker Cell	The size of the cell in which a marker is drawn.
Market Set	The set of symbols from which the marker is obtained.
Marker Symbol	The particular symbol that is to be used to draw markers.

Pattern Attributes

Pattern Set	The set of symbols from which the area fill pattern is obtained.
Pattern Symbol	The particular symbol that is to be used as a fill pattern when filling an area.
Pattern Reference Point	The point that determines the positioning of a custom fill pattern.

Note: The Arc Parameters also specify characteristics of output primitives. In this, they act in a way very similar to attributes, but are conceptually distinct.

Graphics Coordinate Spaces

Two coordinate spaces or presentation spaces are used in AFP GOCA:

- Drawing Order Coordinate Space (DOCS)
- Graphics Presentation Space (GPS)

Drawing Order Coordinate Space (DOCS)

The DOCS is the coordinate space in which the drawing orders specify graphics primitives. Points are described in the drawing orders by specifying the x and y coordinates in the DOCS. Extents and offsets are described in the drawing orders by specifying the x and y extents and offsets in the DOCS. The DOCS is a standard, 2-dimensional Cartesian coordinate system. Units of measure for the DOCS are specified in the Graphics Data Descriptor. In AFP GOCA, there is a one-to-one mapping between the DOCS coordinate system and its units of measure and the GPS coordinate system and its units of measure. Therefore in AFP GOCA, DOCS and GPS are equivalent coordinate systems. All references to coordinate systems in AFP GOCA will be made with respect to GPS.

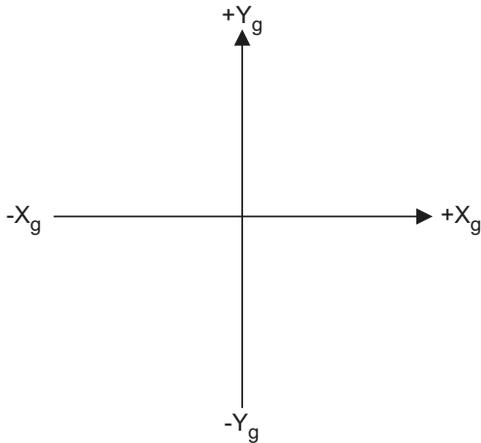
Graphics Presentation Space (GPS)

The GPS is the space in which the application user's view of the specified picture is generated. The GPS is a standard, 2-dimensional Cartesian coordinate system as shown in [Figure 5 on page 14](#). Coordinates in the

Color

GPS coordinate system are denoted by (X_g, Y_g) . Units of measure for the GPS are specified in the Graphics Data Descriptor.

Figure 5. Coordinate System Used for the GPS



AFP GOCA uses 16-bit signed integers to specify GPS coordinates. A point outside GPS is characterized by a 2-byte arithmetic overflow. For a definition of the geometric parameter format used in AFP GOCA, see [“Parameter Type” on page 71](#) and [“Drawing Order Subset” on page 181](#).

Usable Area (UA)

The usable area is a presentation space and coordinate system defined by the controlling environment. It is the space in which the implementation presents the picture to the end user, and merges the GPS with other presentation spaces in the device.

The controlling environment defines a GPS window on the GPS, and a graphics window mapping between the GPS window and the UA. In the AFP environment, the usable area is a MO:DCA or IPDS *object area*, which is merged with other object areas on a *logical page* presentation space.

Color

The color values specified using the Set Color and Set Extended Color drawing orders generate an index into the standard color table defined in [Table 5 on page 15](#). When a primitive is drawn, this color index is mixed with the color index of the GPS using the current mix and background mix attributes. The resulting color index of the GPS can be further modified by drawing another primitive at the same point in the GPS. When drawing is complete, the final color index is used to look up the current color value. The values in the color table control the physical process whereby colors are presented on the presentation surface.

The standard color table is accessed by two-byte color index values. These values are the valid color index values that can be specified in the Set Extended Color order and the Set Color order. The value specified in the Set Color order is prefixed with X'FF' to generate a two-byte color index value. The valid color attribute values and the colors that are drawn when the standard color table is selected are shown in [Table 5 on page 15](#).

[Table 5](#) shows the meanings of the two-byte values. The table also specifies the RGB values that can be used for each named color, assuming that each component is specified with 8 bits and that the component intensity range 0 to 1 is mapped to the binary value range 0 to 255.

Table 5. Standard Color Table

Value	Color	Red (R)	Green (G)	Blue (B)
X'0000' or X'FF00'	Drawing 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 1	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'	Presentation-process default; see Note 2	–	–	–
X'FF08'	Color of medium	–	–	–
All others	Reserved	–	–	–

Notes:

1. The color rendered on presentation devices that do not support white is device dependent. For example, some printers simulate with color of medium, which results in white if white media is used.
2. The presentation-process default specified by X'FF07' is resolved as the presentation device default. This color value is also known in GOCA as neutral white for compatibility with display devices.
3. While the RGB values in the table can be used to render the OCA named colors, many implementations are and have been device dependent. Nevertheless, it is recommended that OCA Black (X'0008') be rendered as C = M = Y = X'00', and K = X'FF'.

The standard color table is equivalent to the Standard OCA Color Value Table defined in the MO:DCA controlling environment; see the *Mixed Object Document Content Architecture (MO:DCA) Reference* for the definition of this table.

Colors may also be specified using the Set Process Color drawing order. This order supports the specification of:

- Process colors, using the RGB, CMYK, and CIELAB color spaces.
- Spot colors, using the highlight color space.

Mix

- Named colors, using the standard OCA color space. This is the color space that is supported by the Set Color and Set Extended Color drawing orders. For definitions of the color values used in this color space, see [Table 5 on page 15](#).

Note: When the standard OCA color space is selected with the Set Process Color drawing order, colors for foreground data are mixed into the GPS in the same manner as described for the Set Color and Set Extended Color orders. However, when any other color space is selected, colors for foreground data always overpaint the GPS.

Mix

If two output primitives drawn into the GPS have a common point, they are mixed at that point to produce a result that is held at that point. The output primitives exist independently in segments, but do *not* exist independently in the GPS. There is no concept of the GPS having layers with the output primitives underlying and overlying one another at points of the space.

Table 6. Foreground/Background in Graphics Presentation Space

Data Type	Foreground	Background
AFP GOCA Graphics	<ul style="list-style-type: none">• Stroked area of lines (including arcs)• Stroked and filled portion of pattern symbols• Stroked and filled portion of marker symbols• Stroked and filled portion of graphic characters• B'1' image points• Entire area with solid fill	Everything else

Mixing applies only to those points of the GPS to which an output primitive is being drawn. The GPS always contains the result of the mixing of the output primitives currently drawn in the GPS. When a new output primitive is drawn into the GPS, each foreground or background point of the output primitive is combined with the corresponding point of the GPS to produce a new result in the GPS. Mixing is always an effect of a foreground or a background value of an output primitive on an existing GPS value.

[Table 6](#) summarizes the definition of foreground and background in the GPS.

Note: A custom pattern or gradient is mixed into the GPS in the same way as are patterns in the default pattern set: only stroked and filled portions of the custom pattern or gradient are foreground. For gradients, however, the entire gradient, as well as any outside areas for which the Outside value was not specified as “None”, is considered stroked and filled, even if the color white occurs.

Implementation Note: If a color fill of an area is simulated with a pattern fill, the complete fill is considered foreground, not just the stroked and filled portion of the pattern symbols.

The attributes of mix and background mix specify the method by which the output-primitive color value is combined with the existing color value of each point of the GPS. These two mixing capabilities are not always the same mixing attribute value. For example, assume that the GPS contains a line on which the controlling environment wants to mix a character A, such that the background of that character does not interfere with the line. The application chooses Overpaint for the foreground mix attribute of the character and Leave Alone for the background mix attribute of the character.

Every point of the GPS is background until points are drawn in GPS. The new color value of the current point of the GPS is obtained by applying the appropriate mix attribute to the existing value for that point with the appropriate, foreground or background, color value for the corresponding point of the output primitive being applied.

The mix attributes are selected by use of the Set Mix or Set Background Mix orders.

In the description that follows, the term *source* means the foreground, or background, of the primitive that is being drawn. The term *destination* means the area of the GPS on which the foreground or background of that primitive is being drawn.

The supported values of the foreground mix attributes are:

- X'00'** *Drawing default.* This resets the mix attribute to its initial value.
- X'02'** *Overpaint.* The color value of the source replaces the color value of the destination. This is also sometimes referred to as *opaque* mixing.

The supported values of the background mix attributes are:

- X'00'** *Drawing default.* This resets the mix attribute to its initial value.
- X'05'** *Leave Alone.* The color value of the destination is unchanged. This is also sometimes referred to as *transparent* mixing.

Segments

Segments are self-contained collections of drawing orders and attributes. They are the basic units from which a picture is constructed. A segment can be given a name defined as a four-byte unsigned integer; however, this name is ignored in AFP GOCA.

Facilities are provided to permit the chaining of segments during the process of describing a complete picture. Chaining is the unidirectional passing of control from one segment to another segment.

Every segment is either *chained* or *unchained*. A collection of one or more chained segments defines the picture to be drawn. Unchained segments are ignored in AFP GOCA. Chaining provides a known and architected initial state for the chained segments. Therefore, chained segments are completely independent pieces of the picture.

Subsetting

GOCA supports the functional requirements of a wide spectrum of graphics devices in a number of different environments. To efficiently support this range of capabilities, GOCA defines subsets of functionality. The subsets described in this manual for AFP GOCA are labeled as follows:

- *Drawing Level 2 Version 0 (DR/2V0).* The DR/2V0 subset is also referred to as “GRS2”.
- *Graphics Subset Level 3 (GRS3).* The GRS3 subset includes additional functionality above DR/2V0.

These subsets are supported in both the MO:DCA and IPDS environments. See [Chapter 9, “Compliance”, on page 175](#) for details of these subsets.

Exception Conditions

Exception conditions are defined by AFP GOCA for detectable errors in the syntax of GOCA constructs. They are reported to the controlling environment in an environment dependent manner.

If the environment determines that processing can proceed, then for some of the exception conditions, AFP GOCA defines a standard action that is to be taken after the error is detected. For the other exception conditions, the environment must determine the continuation procedure.

Chapter 4. Graphics Primitives and Attributes

This chapter describes:

- Output primitives in general
- Current position
- The symbols used to draw characters, markers, and shading patterns in areas
- The following output primitives and their associated attributes:
 - Lines
 - Areas
 - Character strings
 - Markers
 - Images
- Output primitive overflow

Output Primitives

Output primitives are the basic element from which graphics pictures are built. They are drawn by one or more drawing orders containing the parameters that define the primitive.

Primitives also use the modal parameters called *attributes* associated with them, as well as the general drawing process controls.

The architecture defines exception conditions for invalid values of parameters within drawing orders and assigns exception condition codes, EC-xxxx, to these for reporting purposes. See [“Drawing Order Exceptions” on page 168](#) for details.

Current Position

Current position is a position in Graphics Presentation Space (GPS) remembered by the drawing processor. Current position is updated by the drawing processor as each output primitive value is executed. It is maintained as an (X_g, Y_g) coordinate value in GPS. With the drawing orders that are described in [Chapter 7, “Commands and Drawing Orders”, on page 73](#), this updating of current position can, in general, be implemented by replacing the old value of current position by an (X_g, Y_g) coordinate from the order being executed.

Two alternative forms of each output primitive drawing order are provided, each with a different order code:

- With the first form, all coordinates required to draw the output primitive are contained in the order itself.
- With the second form, the current position is used as the first pair of coordinate values of the output primitive.

The second form of drawing order is shorter than the first form. The second form is used when the initial coordinate of an order is the current position as established by the previous order, and effectively connects the primitives together.

The drawing order Set Current Position is provided to manipulate current position.

Current position is set to the origin of GPS—that is, $(X_g=0, Y_g=0)$ —at the beginning of each new segment and at the beginning of each new custom pattern definition.

Symbols

Symbols are used to draw:

- Characters
- Markers
- Shading patterns in areas

A particular symbol can be used as a character, as a marker, or as a pattern.

For character symbols, the controlling environment provides access to sets of symbols by resolving the local identifier of the character set.

When drawing character symbols, the concept of precision exists:

- The minimum degree of accuracy required for the appearance of the symbols is determined by the value of the character precision attribute.
- The method of defining a symbol does not limit the operations that can be applied to that symbol. Therefore, the method of symbol definition does not tie that symbol to a particular level of precision. An implementation can choose to support only certain precisions for particular types of symbol definition. Subsets may define what precision is required to be supported.
- Precision and the method by which symbols are defined are independent of each other.

To draw a symbol, an x,y position, a symbol set, and a code point are specified.

If the requested symbol set does not exist, the appropriate exception condition is raised. The standard action for this exception is to use the appropriate standard default set.

If the code point identifies a symbol that is not valid or not defined, the appropriate exception condition is raised. The standard action for this exception is to use the appropriate standard default symbol.

Markers, patterns, and characters are all examples of symbols. Parts of the loading mechanism and handling facilities are common to all types of symbols.

[Table 7](#) summarizes how attributes are set when symbols are used for characters, markers, and patterns.

Table 7. Setting Attributes for Character, Marker, and Pattern Symbols

Attribute	Symbols		
	Character	Marker	Pattern
Color	Set Color, Set Extended Color, or Set Process Color orders	Set Color, Set Extended Color, or Set Process Color orders	Set Color, Set Extended Color, or Set Process Color orders
(Foreground) Mix	Set Mix order	Set Mix order	Set Mix order
Background Mix	Set Background Mix order	Set Background Mix order	Set Background Mix order
Precision	Set Character Precision order	Reserved	Reserved
Shear	Set Character Shear order	Reserved	Reserved
Angle	Set Character Angle order	Reserved	Reserved
Cell Size	Set Character Cell order	Set Marker Cell order	Reserved
Direction	Set Character Direction order	Not applicable	Not applicable
Set	Set Character Set order	Set Marker Set order	Set Pattern Set order
Code Point	Character String order	Set Marker Symbol order	Set Pattern Symbol order
Reference Position	Character String order	Marker order	Set Pattern Reference Point order for a custom pattern; not applicable for a gradient; otherwise, device default

In raster symbol definitions and in fully described and outline fonts, the foreground color of the symbol is always the current character color attribute value.

Line Primitives

There are three types of line primitives:

- Straight Lines
- Curved Lines
- Boxes

Straight Lines

The following orders can be used to draw straight lines:

- The Line order draws one or more contiguous straight lines by providing the endpoints of each line.
- The Relative Line order draws one or more contiguous straight lines by using offset values.

Line Order

The Line order has two forms:

- Line at Given Position (GLINE) order
- Line at Current Position (GCLINE) order

Straight lines are drawn through the set of points specified as parameters of the order. In general, any number of points can be specified, provided the maximum length count on the order is not exceeded.

The current values of the line attributes are taken into account when the lines are drawn. Current position is set to the last point specified in the order.

Relative Line Order

The Relative Line order has two forms:

- Relative Line at Given Position (GRLINE) order
- Relative Line at Current Position (GCRLINE) order

The parameters of the order include an initial position, (X_0, Y_0) , and a set of offset values, $\{D_1, E_1\}, \dots, \{D_n, E_n\}$. The offsets are one-byte values that give the end point of a line relative to the start of that same line; that is, the differences in the x,y coordinate values of the start and end points of the line. Negative values for these offsets are permitted.

Straight lines are drawn between the points (X_0, Y_0) , $(X_0 + D_1, Y_0 + E_1)$, $(X_0 + D_1 + D_2, Y_0 + E_1 + E_2)$, \dots , $(X_0 + D_1 + \dots + D_n, Y_0 + E_1 + \dots + E_n)$.

The current values of the line attributes are taken into account when the relative lines are drawn. Current position is set to the last point calculated.

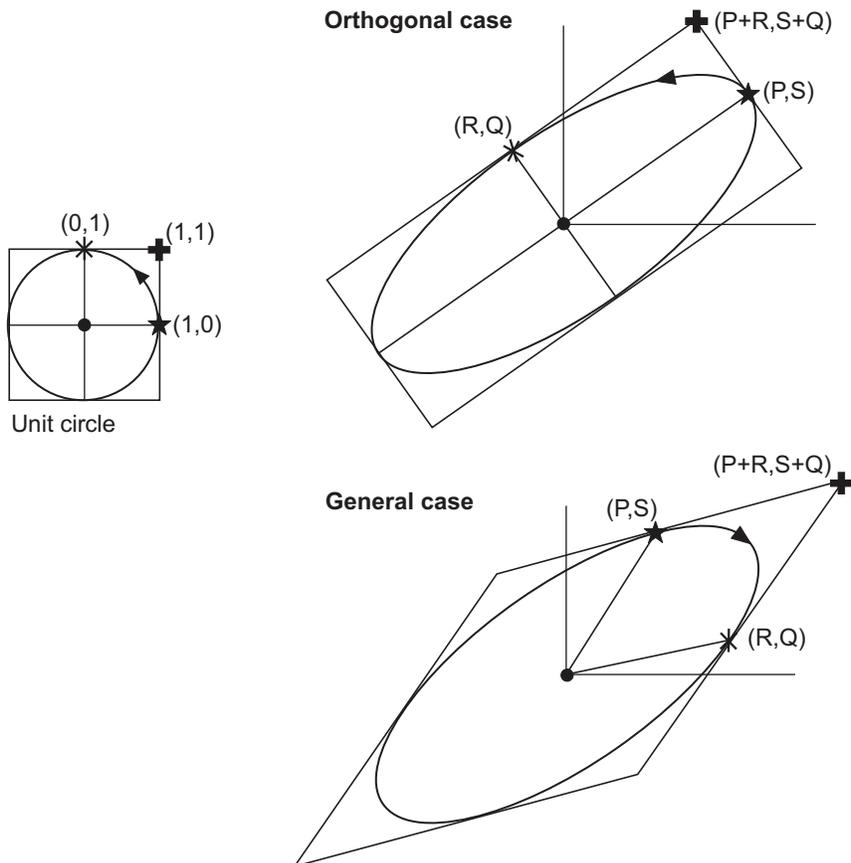
Note that the straight lines are drawn so that the line width is centered on the specified points.

Curved Lines

Curved lines can be drawn using the following orders:

- Full Arc
- Partial Arc
- Fillet
- Cubic Bezier Curve

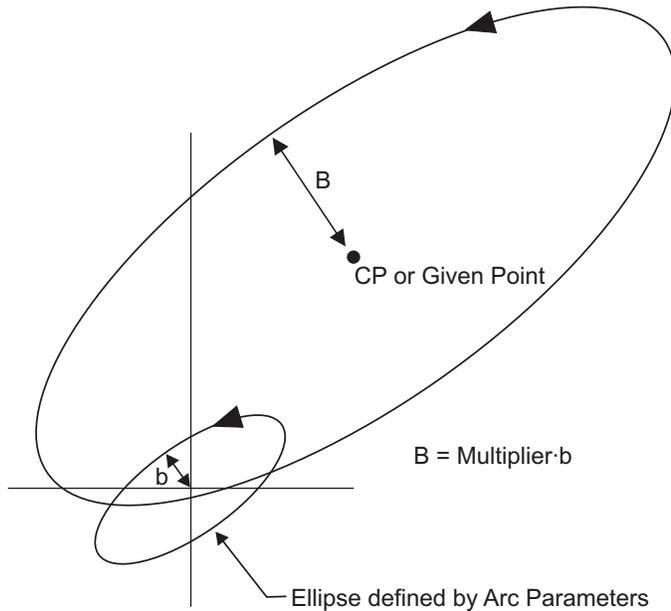
Figure 6. Arc Parameters



Line Primitives

Full Arc

Figure 7. Full Arc



Full Arc orders use the current value of the arc parameters to define the primitive. The arc parameters specify the required *shape* and *size* of an ellipse, which can be a circle. The arc parameters also specify the direction in which the ellipse is drawn: clockwise or counterclockwise.

The Set Arc Parameters order sets the current value of the arc parameters. The arc parameters are shown in [Figure 6 on page 23](#).

The parameters in the Set Arc Parameters order, P, Q, R, and S, define a transformation that maps the unit circle to the required ellipse, placed at the origin (0,0):

$$X' = P \cdot X + R \cdot Y$$

$$Y' = S \cdot X + Q \cdot Y$$

where X and Y are the coordinates of the points on the unit circle, and X' and Y' are the coordinates of the points on the defined ellipse. Note that the unit circle has a radius of 1 GPS unit.

If $P \cdot R + S \cdot Q = 0$, the transform is termed orthogonal and the line from the origin (0,0) to the point (P,S) is either a radius of the circle, or half the major/minor axis of the ellipse. The line from the origin to the point (R,Q) is either the radius of the circle, or half the minor/major axis of the ellipse.

If $P \cdot Q = R \cdot S$, the ellipse degenerates to a straight line or a point. If $P = Q = r$ and $R = S = 0$, the ellipse degenerates to a circle with radius r .

The direction that the circle or ellipse is drawn depends upon the determinant $P \cdot Q - R \cdot S$, as follows:

- If $P \cdot Q - R \cdot S > 0$, the direction is counterclockwise.
- If $P \cdot Q - R \cdot S < 0$, the direction is clockwise.

Implementation Note: For historical reasons, not all output devices have supported the directionality of arcs based on the determinant. However, all output devices that support the Cubic Bezier Curve (GCBEZ, GCCBEZ) drawing orders must support arc directionality based on the determinant.

The Full Arc order draws one complete circle, or a complete ellipse. The parameters of this order are the center point, and a multiplier that specifies by how much the ellipse or circle defined by the Set Arc Parameters order is to be scaled, before being drawn. The Full Arc circle or ellipse is drawn either in a clockwise or counterclockwise direction, depending on the direction of the ellipse as defined by the arc parameters. [Figure 7 on page 24](#) shows the generation of an ellipse. The small ellipse at the origin is defined by the Set Arc Parameters order with minor axis $2b$. The Full Arc drawing order transforms this ellipse into an ellipse with center at current position or a specified point, and with a multiplier such that the new minor axis $2B = \text{Multiplier} \cdot 2b$. The major axis is scaled in the same manner.

The current values of the line attributes are taken into account when each full arc is drawn. Current position is set to the center point of the arc.

In AFP environments, the standard default for the arc parameters is:

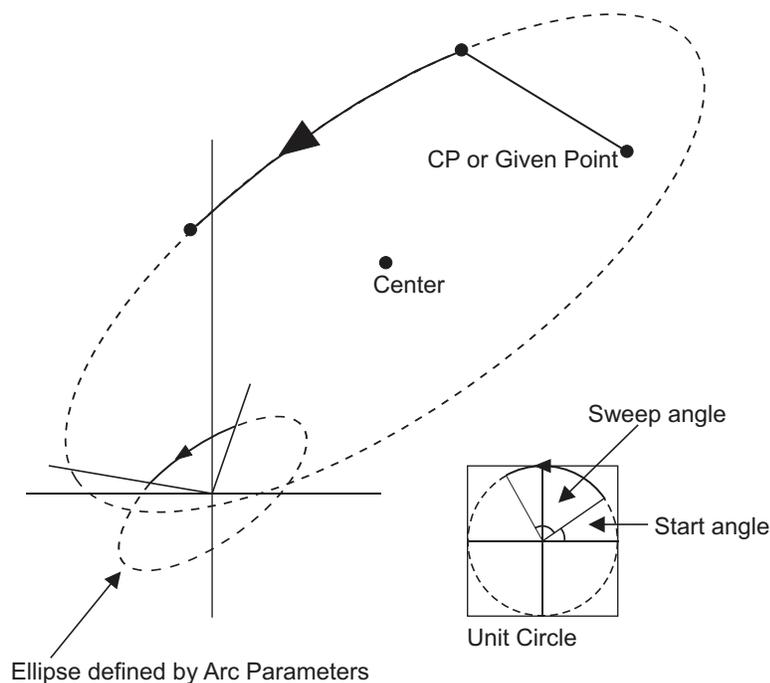
$$P=Q=1$$

$$R=S=0$$

Note that the parameter values are specified in GPS L-units.

Partial Arc

Figure 8. Partial Arc



The Partial Arc primitive draws a line from a specified point or current position to the start of an arc, and then draws the arc.

The arc is part of the full arc defined by the current arc parameters and the multiplier M . The center of the arc is at a point specified within the Partial Arc drawing order. The part of the arc that is drawn is defined by the start-angle and sweep-angle parameters. The direction of drawing is determined by the arc parameters.

The start angle is the angle between the X axis of the unit circle space and the radius drawn from the center of the arc to the start point of the arc. The sweep angle is the angle subtended at the center of the arc by the two radii drawn from the center of the unit circle to the start and end points of the arc; see [Figure 8](#).

Line Primitives

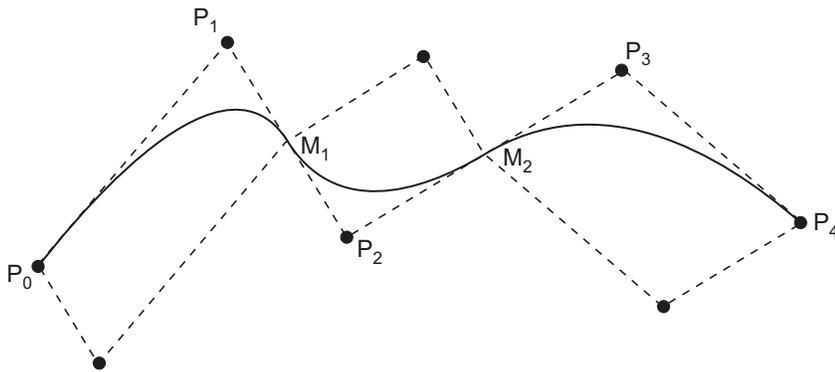
Both angles are specified in the unit-circle space, and hence are transformed by an amount defined by the current arc parameters in the same way that the unit circle is transformed. If the partial arc is part of a circle, the angles following the transform will be the same as the angles on the unit circle. If the partial arc is part of an ellipse, the angles following the transform will, in general, be different than the angles on the unit circle.

The Partial Arc order can draw arcs with sweep angles greater than or equal to 0° and less than 360° ; it cannot draw a complete 360° arc. The Full Arc drawing order can be used to draw a complete arc.

The current values of the line attributes are taken into account when the partial arc is drawn. Current position is set to the endpoint of the arc.

Fillet

Figure 9. Fillet



This primitive is drawn using the Fillet order. The parameters of the order are the (X_g, Y_g) coordinates for a set of points P_0, P_1, \dots, P_n .

The points specified in the order are joined by conceptual straight lines, to which a curve is fitted. The curve is tangential to the first line at its start point, and to the last line at its end point. If there are more than two lines, the curve is tangential to the intermediate lines at their center points. If only two points are supplied, a straight line is drawn between the points.

Architecture Note: The Fillet drawing order does not support specification of a *sharpness* parameter. In [Figure 9](#), this parameter would determine how close the drawn curve comes to the points P_1, P_2 , and P_3 . If a quarter circle or quarter ellipse is used to fit the points, the sharpness parameter is not required since the circle or ellipse is completely defined by completing the parallelogram. If a quarter arc is not used, a sharpness parameter can be used and is defined, in reference to [Figure 9](#), as follows:

1. Generate the virtual line P_0M_1
2. Find the midpoint of this line, V_0
3. Generate V_0P_1
4. Call the point where V_0P_1 intersects the arc D_1
5. The sharpness parameter is defined to be the ratio of $V_0D_1 \div D_1P_1$

The recommended value for the sharpness parameter, when used in AFP GOCA, is 0.7.

The current values of the line attributes are taken into account when the fillet is drawn. Current position is set to the last point specified.

The curve that is drawn is computed as follows (see [Figure 9 on page 26](#)).

1. Let the points specified in the order be known as P_0, P_1, \dots, P_n .
2. Conceptual lines are drawn between the points P_0 to P_1 , P_1 to P_2 , P_2 to P_3 , and so on.
3. The midpoints of the lines from P_1 to P_2 , P_2 to P_3 , \dots , P_{n-2} to P_{n-1} are computed; call these M_1, M_2, \dots, M_{n-2} .
4. The points $P_0, P_1, M_1, P_2, M_2, P_3, \dots, M_{n-2}, P_{n-1}, P_n$ are then considered three at a time, starting with P_0, P_1, M_1 . A quadrant of a circle is scaled, and can be distorted to become a part of an ellipse, in order that the curve be tangential to the line P_0 - P_1 at the point P_0 , and tangential to the line P_1 - M_1 at the point M_1 .

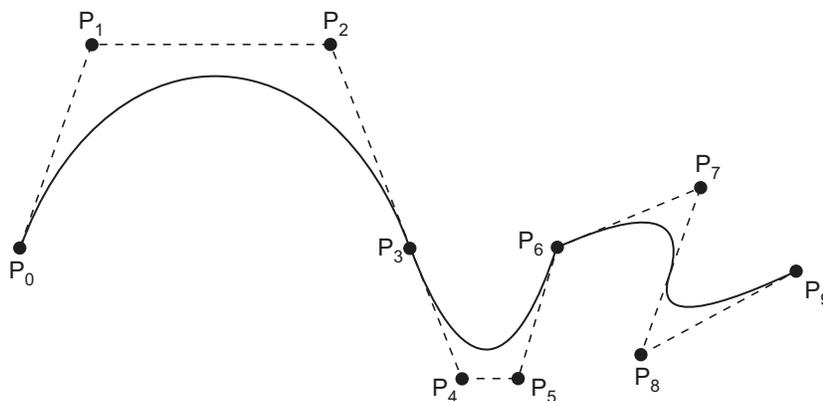
The center point of the ellipse is the point obtained by completing the parallelogram defined by the sides P_0 - P_1 and P_1 - M_1 .

5. The next three points are considered, that is M_1, P_2, M_2 and a quadrant of a circle is transformed into part of an ellipse that is tangential to the line M_1 - P_2 at M_1 , and tangential to the line P_2 - M_2 at M_2 .
6. This process continues, with part of an ellipse being fitted to three points in turn, until the last three points M_{n-2}, P_{n-1}, P_n , have been incorporated; see [Figure 9 on page 26](#).

Note: If all the points P_0 through P_n are within the GPS, the actual fillet does not go outside the GPS.

Cubic Bezier Curve

Figure 10. Cubic Bezier Curve



A Cubic Bezier curve primitive is drawn by executing a Cubic Bezier Curve order. The parameters of the order are a set of points P_0, P_1, \dots, P_n . These points are considered in sets of four, the first being P_0, P_1, P_2 , and P_3 . The second set is P_3, P_4, P_5 , and P_6 ; that is, the set overlaps at P_3 with the first set. This process continues, each set overlapping the previous set by one point, up to the final set, which is $P_{n-3}, P_{n-2}, P_{n-1}$, and P_n .

An exception condition, EC-0003, occurs if the length of the order does not provide a number of points equal to $3m + 1$, where m is the number of sets. The number of points provided includes the current position when the order is Cubic Bezier Curve at Current Position.

A curve is drawn independently for each set of four points. It is computed as follows:

1. Let the first set of points be labeled P_0, P_1, P_2 , and P_3 . The curve is drawn from P_0 to P_3 and is tangential to P_0 - P_1 and P_2 - P_3 ; see [Figure 10](#).

2. The curve drawn is defined by the parametric equations:

$$x(t) = P_x \cdot t^3 + Q_x \cdot t^2 + R_x \cdot t + P_{0x}$$

$$y(t) = P_y \cdot t^3 + Q_y \cdot t^2 + R_y \cdot t + P_{0y}$$

t takes values from 0 to 1.

Line Primitives

P_0x and P_0y are the x and y coordinates of point P_0 .

P_x , Q_x , and R_x are given by the following formulas:

$$P_x = P_3x - 3 \cdot P_2x + 3 \cdot P_1x - P_0x$$

$$Q_x = 3 \cdot P_2x - 6 \cdot P_1x + 3 \cdot P_0x$$

$$R_x = 3 \cdot P_1x - 3 \cdot P_0x$$

Note: P_1x and P_1y are the x and y coordinates of point P_1 .

P_2x and P_2y are the x and y coordinates of point P_2 .

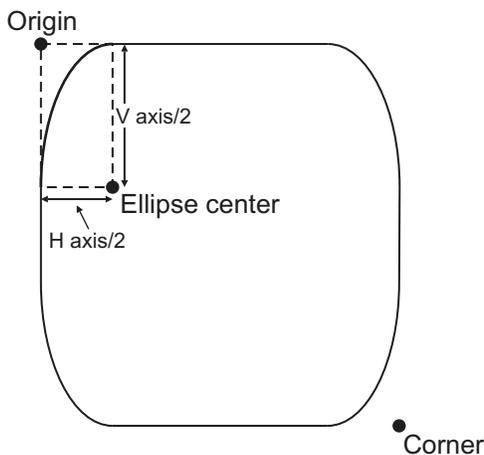
P_3x and P_3y are the x and y coordinates of point P_3 .

- P_y , Q_y , and R_y are given by the same formulas, but using the y coordinates of the points P_0 , P_1 , P_2 , and P_3 instead of the x coordinates.

The choice of control points for the curves determine whether there is a smooth transition from one curve to the next. A smooth transition requires that two curves have the same slope at their connecting point. To ensure that the curves drawn have the same slope at the connecting point, the second control point of the previous curve and the first control point of the new curve must be collinear with the common point of the two curves. In [Figure 10 on page 27](#), for example, P_2 (second control point of the first curve), P_3 (common point), and P_4 (first control point of the new curve) are on the same line resulting in a smooth transition, whereas P_5 (second control point of the second curve), P_6 (common point), and P_7 (first control point of the third curve) are not collinear, resulting in the two curves joining at an angle.

Boxes

Figure 11. Box



The Box order has the two forms:

- Box at Given Position (GBOX)
- Box at Current Position (GCBOX)

The Box order draws a rectangular box with corners that are either square, quadrants of a circle, or quadrants of an ellipse.

The parameters on the order are:

- The corner positions of a rectangle. There are two corners specified; in the case of the GCBOX order, one of the corners is the current position.
- The lengths of the horizontal and vertical axes of an ellipse.

If the lengths of the axes are:

- Both omitted, or either is specified as zero, square corners are drawn.
- Equal and nonzero, each corner is a quadrant of a circle.
- Not equal and nonzero, each corner is a quadrant of an ellipse.

The current values of the line attributes, except for line join, are taken into account when the box is drawn. The line end attribute is used only for the internal ends of dotted or dashed lines.

Current position is set to the first corner specified for the GBOX order, or is unchanged for the GCBOX order.

Line Attributes

[Table 8](#) shows the attributes controlling the drawing of line primitives, that is, straight lines, curved lines, and boxes.

Table 8. Attributes Controlling Line Primitives

Attribute	Standard Default	Length (in bytes)	Meaning
LINE TYPE	Solid (X'07')	1 or 4 <i>n</i> (where <i>n</i> is the number of custom dash/move pairs)	Specification of type of line
LINE WIDTH	Normal (X'0100')	2	Specification of line width as fractional multiplier of normal line width
NORMAL LINE WIDTH	Device dependent	2	Specification of normal line width in 1440ths of an inch
LINE END	Round (X'03')	1	Specification of line end
LINE JOIN	Round (X'02')	1	Specification of line join
COLOR	Device dependent	2	Color value set into GPS for foreground
PROCESS COLOR	Device dependent	12–14	Process color value set into GPS for foreground
MIX	Overpaint (X'02')	1	Specification of Mix mode in GPS for foreground

Line Type

The line type attribute controls the type of line used to draw lines.

The line type is defined as a series of dots and dashes. As lines are drawn into the GPS, the line type is used repetitively to determine which parts of the line are drawn into the GPS:

- The dots and dashes are drawn.
- The spaces between the dots and dashes are not drawn and have no effect on the GPS.

Line Attributes

The sequence of line type dots and dashes is not reset, except by a Move Type order, which is an order that causes current position to be updated to a new value specified in the order before anything is drawn. Move Type orders are defined in [Table 9](#).

Table 9. Move Type Orders

Description	Orders
Any straight or curved line order that explicitly specifies the starting point of the line that is to be drawn	<ul style="list-style-type: none"> • Box at Given Position (GBOX) • Cubic Bezier Curve at Given Position (GCBEZ) • Fillet at Given Position (GFLT) • Full Arc at Given Position (GFARC) • Line at Given Position (GLINE) • Partial Arc at Given Position (GPARC) • Relative Line at Given Position (GRLINE)
Orders that explicitly or implicitly set current position	<ul style="list-style-type: none"> • Set Current Position (GSCP)
Other orders that specify an initial position	<ul style="list-style-type: none"> • Begin Image at Given Position (GBIMG) • Character String at Given Position (GCHST) • Marker at Given Position (GMRK)

The line type attribute can be set to a *standard* or a *custom* value. Standard values are predefined line types and are set with the Set Line Type drawing order. Custom values are arbitrary, user-defined line types and are set with the Set Custom Line Type drawing order.

The *standard* line types are defined as follows:

Attribute	Meaning
X'00'	Drawing Default
X'01'	Dotted Line
X'02'	Short-dashed Line
X'03'	Dash-dot Line
X'04'	Double-dotted Line
X'05'	Long-dashed Line
X'06'	Dash-double-dot Line
X'07'	Solid Line
X'08'	Invisible Line

When the line type attribute is set to invisible, current position is updated, but nothing is drawn into the GPS.

The exact appearance of the standard line types is implementation dependent. For consistent appearance of the standard line types, the following guideline should be used. The guideline defines the line types in terms of drawing the dashes and dots and moving over the spaces between them. The lengths are expressed in units of line width. The first number is the length of the first dash or dot in the sequence, and the second is the length of the move that follows. Further pairs of numbers, defining the dash, dot, and move lengths, are defined for the more complex line types.

The guidelines for generating the *standard* line types are as follows:

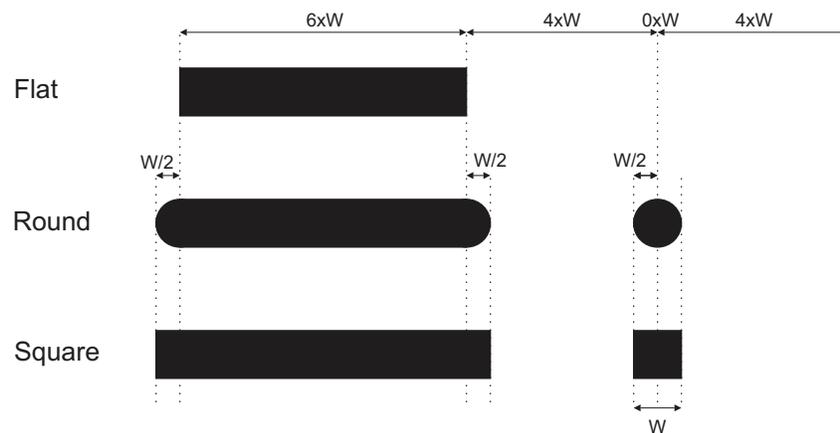
Attribute	Sequence
X'01'	0, 2
X'02'	3, 3
X'03'	6, 4, 0, 4
X'04'	0, 3, 0, 7
X'05'	8, 3
X'06'	6, 3, 0, 3, 0, 3

Note: Because the size of a dot is zero, it is only visible because of the two concatenated line ends, and will disappear when flat endings are chosen.

A Dash-dot Line with different line endings is illustrated by [Figure 12](#).

Figure 12. Example of the Dash-dot Line Type

Dash-dot Line (6,4,0,4)



A custom line type is a sequence of pairs d,m , where d is the length of the dash and m is the length of the move that follows. The lengths are expressed in units of line width. This follows the same format as shown above for the standard line types. A zero value for d indicates a dot. If all values d and m are zero, a solid line is drawn.

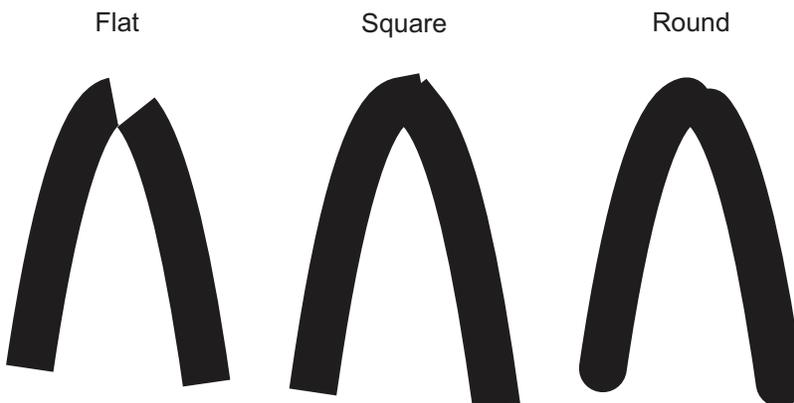
Unusual results can occur when small move values (for example, $m \leq 1$) are specified in custom line types; see [Figure 13](#). A “dashed” line might end up looking partially or completely solid, possibly with “notches” on the edges or “barbs” protruding. Such lines, however, should be drawn exactly as the custom line type directs, and should not be “smoothed” or otherwise made more pleasing.

It is not possible to use the Set Current Defaults instruction to set a custom line type as a default.

There is only one line type attribute. It has a current value that is either a standard value or a custom value, depending on which drawing order (Set Line Type or Set Custom Line Type) was specified last.

Figure 13. Custom Line Type with Small Move Value (with Different Line Ends)

Custom dashed line (6,0.5)



Line Attributes

Line Width

The current line width attribute controls the width of line used to draw lines.

The line width attribute consists of an integral and fractional part. When only the integral part is set by a Set Line Width order, the fractional part is reset to zero.

The value of the Line Width attribute specifies a multiplier of the normal line width.

The normal line width may be set as a default to an absolute value in 1440ths of an inch in the Set Current Defaults instruction in the data descriptor. In this case, to determine the line width in output-medium pels, first the line width multiplier is determined. An implementation then multiplies the line width multiplier by the absolute value normal line width and selects the nearest supported value in output-medium pels.

If not set to an absolute value, the normal line width and all multiples of the normal line width are device dependent (although see the Implementation Note below for a recommendation).

In any case, if the calculated line width exceeds the maximum supported by the device, that maximum is used.

Attribute	Meaning
X'0000'	Drawing Default. The value of the attribute when the graphics processor was invoked. This value was set either by the Set Current Defaults instruction in the Graphics Data Descriptor or by the controlling environment.
X'0100'	Normal line width (multiplier of 1).
X'nnnn'	Multiplier. The high-order byte is an integral multiplier of the normal width, and the low-order byte is a fractional multiplier.

Architecture Note: The line width should be scaled when the controlling environment specifies a scaling mapping of the GPS window into the usable area (object area).

Implementation Note: If a normal line width is not specified with the Set Current Defaults instruction (SET = X'11'), the following algorithm is recommended for generating line widths.

```
Line width (in inches) =  
    MH.MFR x 1/120 inches = MH.MFR x 0.00833 inches
```

```
If resulting line width < 1 pixel,  
    set line width to 1 pixel  
If resulting line width > .13 inches,  
    set line width to .13 inches
```

The term MH.MFR is a decimal number consisting of the integral line width multiplier followed by the fractional line width multiplier. For example, if MH = 6 and MFR = 1/4, MH.MFR = 6.25.

Line End and Line Join

The line end attribute defines the shape used at the start and end of contiguous lines that are drawn by a set of straight or curved lines.

The line join attribute defines the shape used for the junction between contiguous lines. This attribute is used to define the join between lines in the following instances:

- At the intermediate points within output primitives that contain multiple lines.
- At the junction that occurs between the end point of a line primitive and the start point of a following line primitive specifying at *current position*, except when any of the following orders occur between the two primitives:
 - A Move Type order. See [Table 9 on page 30](#) for the definition of a Move Type order.

- A Set of any Line attribute.
- A Begin Area or End Area order.
- At the junction between the start point of a closed figure within an area and the end point of the closed figure. See [“Areas” on page 36](#) for the definition of a closed figure.

The line end attribute is used to define the outline of *all* other line end points.

Notes:

1. Except as detailed in 2 below, the line end attribute is not applicable to Full Arc or Box as they are closed figures.
2. The line end attribute applies only to the extreme ends of a set of contiguous lines, and if the line type is dotted or dashed, the internal ends of the dots and dashes.
3. Line ends are drawn only if the end point is visible, as indicated by the current line type attribute.
4. Line joins are drawn only if the junction point and the adjacent points of *both* lines are visible.
5. Line joins are not applied to the corners of the Box figure.

Line End

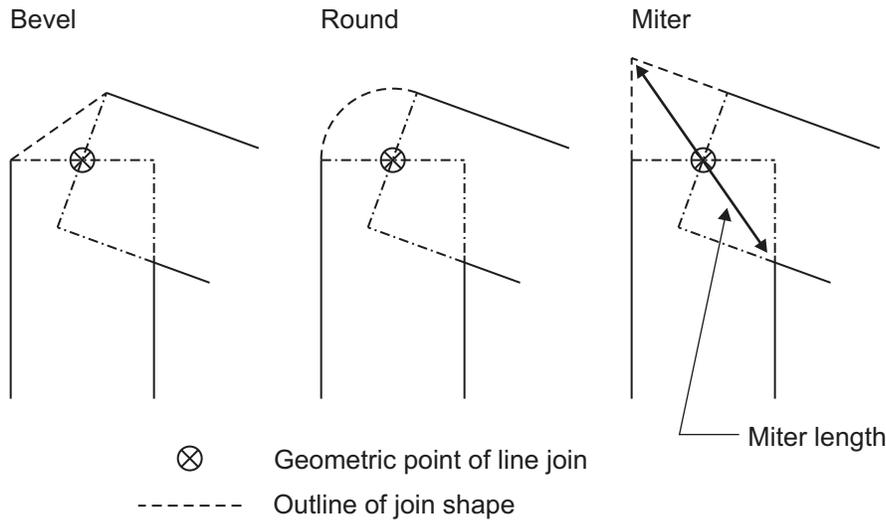
The line end attribute defines the shape used for the outline of the start and end of contiguous lines, drawn by one or more Line primitives. If the line is not solid, it also defines the shape used for the internal ends of the dots and dashes.

The attribute can have the following values:

Attribute	Meaning
X'00'	Drawing Default. The value of the attribute when the graphics processor was invoked. This value was set either by the Set Current Defaults instruction in the Graphics Data Descriptor or by the controlling environment.
X'01'	Flat. The boundary of the end is formed by truncating a line at the end point, perpendicular to the line direction.
X'02'	Square. The boundary of the end is formed by extending the line, in the line direction at the end point, by half the line width. This extension is then truncated with a line perpendicular to the line direction.
X'03'	Note: The effect is to place a rectangle on the end, even if the line is curved. Round. The boundary of the end is formed by a semicircle, centered at the end point, with a radius of half the line width.

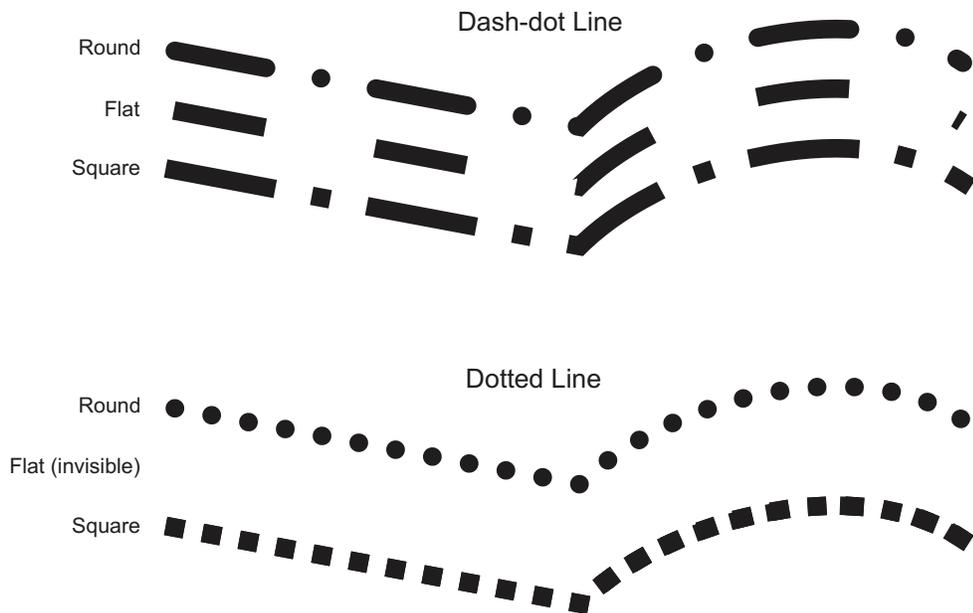
Bevel, round, and miter line-join shapes are illustrated in [Figure 15](#).

Figure 15. Line Join Shapes



Note: When non-solid lines are joined and flat line ends are used, the resulting corner can look strange. For example, with a dashed line that joins another dashed line at an angle, the corner will look different depending on how much of a dash bends around the corner; notches can appear when one of the line segments is shorter than its width, see [Figure 16](#) (Dash-dot line, Flat endings). It is not necessary for an implementation to attempt to compensate for these inherent problems.

Figure 16. Miter Line Joins in Combination with Line Types and Line Ends



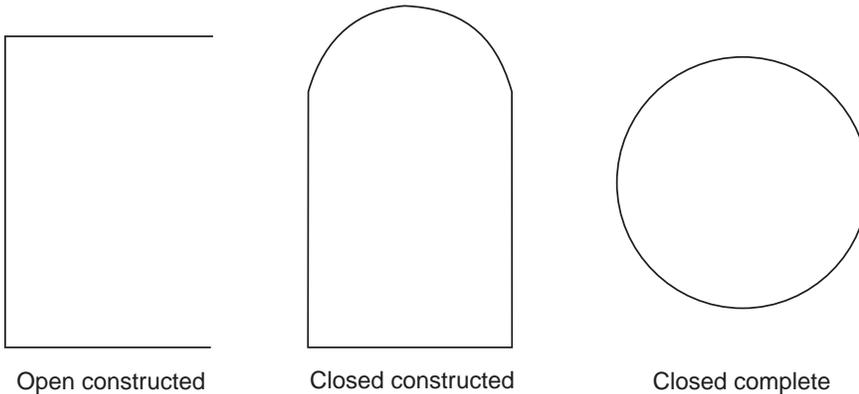
Areas

Areas are two-dimensional, composite primitives defined within a Begin Area/End Area bracket. An area is defined by its boundary, which is filled with a shading pattern.

An area definition may start in one segment and be completed in an appended segment.

The boundary of an area is defined as one or more *closed figures* that are either constructed or complete; see [Figure 17](#). An example of a complete figure is one defined by the Full Arc order. Each constructed figure consists of a set of straight and curved lines connected together. These lines can be drawn if required.

Figure 17. Closed and Open Figures



The following description refers to a *Move Type* order. This term refers to the type of order that causes current position to be updated to a new value specified in the order before anything is drawn. See [Table 9 on page 30](#) for a list of Move Type orders.

The first constructed figure in an area is defined as starting at the Begin Area order. It is delimited either by an End Area order, or by any Move Type order that is valid in an area definition, which implies the start of a new closed figure. See [“Valid Area Definitions” on page 38](#) for a list of orders that are valid in an area definition.

Implementations can choose to allow complete figures, such as Full Arc at current position, within a constructed figure.

Each figure in an area must be properly closed, that is, its start and end points must be identical. If the points are not identical, the figure is closed arbitrarily with a straight line connecting the start and end points. The current position is set to the start point of the figure.

Implementation Note: If the Begin Area order specifies that the boundary is to be drawn, and if the area is not properly closed, the generated line to close the figure may or may not be drawn; this is presentation device dependent. The architecture recommendation is that this line not be drawn¹. If the Begin Area order specifies that the boundary is not to be drawn, the generated line to close the figure is not drawn.

The figures formed in this way jointly define the area boundary. The interior of the area is shaded using the values of the pattern attributes that were current when the Begin Area order was executed.

1. Not drawing the generated line to close the figure, while not required in the AFP GOCA GRS3 subset, is required in the MO:DCA IS/3 interchange set.

The interior of the area can be defined in one of two ways, as specified on the `Begin Area` order (see [Figure 18](#)):

- *Alternate mode*

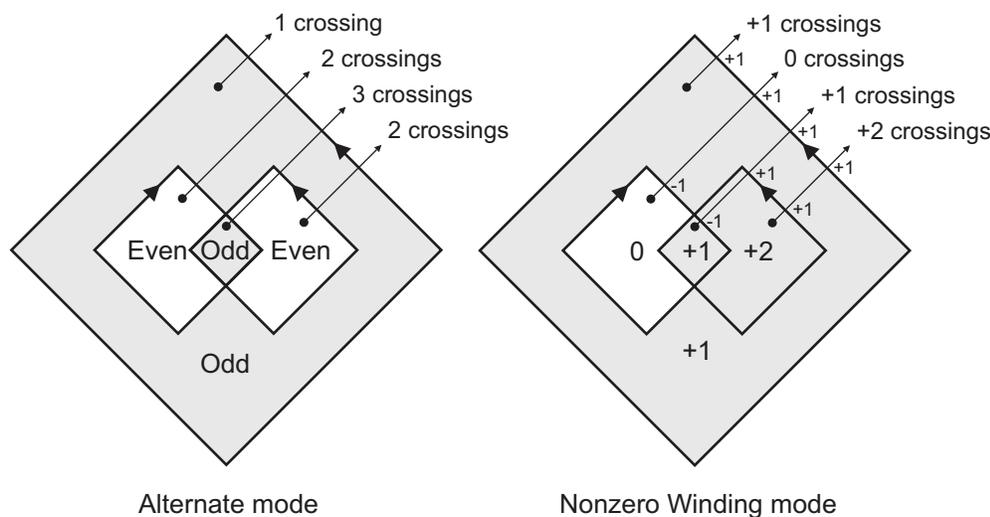
Whether any point is within the area interior is determined by drawing a conceptual line from that point to infinity, without crossing any vertices. If this line crosses the area boundary an *odd* number of times, the point is in the area interior and the region containing that point is shaded. When counting line crossings, coincident boundary lines are all counted. Regions with an *even* number of line crossings from infinity are not shaded.

- *Nonzero Winding mode*

Nonzero Winding mode is similar to alternate mode, in that whether any point is within the area interior is determined by counting the number of times a conceptual line crosses the area boundary. With this mode, however, the direction of the boundary lines is taken into account. Using the same conceptual line, the number of crossings is counted, but boundary lines going in one direction—for example, right-to-left—count as +1, while lines going the other direction—left-to-right—count as -1. The original point is then considered to be inside the interior if the final count is nonzero. Thus a region with a nonzero number of line crossings from infinity is shaded, and a region with a zero number of line crossings from infinity is not shaded.

Implementation Note: It is strongly recommended that implementations that support Nonzero Winding mode also support directionality of full and partial arcs, and directionality of boxes.

Figure 18. Determining the Interior of an Area



The area is filled with the pattern specified by the pattern set and pattern symbol attributes that were current when the `Begin Area` order was executed. If no such set is available, exception condition EC-6803 is raised, the standard action for which is to use the standard default pattern set. If the code point is undefined in the specified pattern set, exception condition EC-6804 is raised, the standard action for which is to use the standard default pattern symbol. In AFP environments, this is X'10'—Solid fill.

Logically, an area is constructed as follows:

1. When an `End Area` order is executed, the closed figures within the area are filled. The values of the pattern set, pattern symbol, pattern mix, and pattern background mix attributes that were current when the `Begin Area` order was executed are used in generating the fill pattern. When using the default pattern set or a bilevel custom pattern (but not when using a full-color custom pattern or gradient), the value of the pattern color attribute that was current when the `Begin Area` order was executed is also used in generating the fill pattern. When using a custom pattern, the value of the pattern reference point attribute that was current when the `Begin Area` order was executed is also used in generating the fill pattern. After the `End Area`

Areas

order is executed, the current pattern color, pattern mix, and pattern background mix attributes are updated to reflect any change in the color, mix, and background mix attributes that may have been specified inside the area definition.

2. If indicated by the Begin Area order, the area boundary is drawn in the GPS in the sequence that the drawing orders that define the boundary are executed. The boundary lines are drawn using the values of the Line attributes that are current at the time the orders defining the boundary are executed.

If no boundary lines are drawn, or the line type specified is invisible, the boundary acts as if it were drawn as a zero-width line.

If two areas are defined that are adjacent to each other, that is, they have at least one boundary line in common, the fill patterns used in the two areas can overlap at the boundary. It is also permissible to overlap the boundary with the fill pattern. Drawing boundaries with solid lines and a mix value of Overpaint ensures consistent results.

The value of the current position is not changed by the Begin Area order itself, but is changed by those orders that are used to define the area boundary.

Valid Area Definitions

An exception condition, EC-6801, occurs if the Begin Area and End Area orders delimiting an area definition are not both in the same segment. Note that an appended segment is part of the segment that it appends. Area orders cannot be nested.

Only the following orders are allowed between the Begin Area order and its corresponding End Area order:

- Box
- Comment
- Cubic Bezier Curve
- Fillet
- Full Arc
- Line
- No-Operation
- Partial Arc
- Relative Line
- Set Arc Parameters
- Set Background Mix
- Set Color
- Set Current Position
- Set Custom Line Type
- Set Extended Color
- Set Fractional Line Width
- Set Line End
- Set Line Join
- Set Line Type
- Set Line Width
- Set Mix
- Set Process Color

Supported orders other than those listed above raise exception condition EC-6802.

Note: The Marker and Character String orders are not permitted as part of an area definition.

Patterns

A pattern is used to fill the interior of an area and is created by selecting a symbol using the pattern set and pattern symbol attributes. For patterns that are not gradients, this symbol is repeated, both in the horizontal and vertical directions, to fill the interior of an area. There are three types of patterns in AFP GOCA:

- Patterns in the default pattern set
- Custom patterns
- Gradients

Note that for all three types of patterns, a Pattern Symbol attribute value of X'00' selects the drawing default symbol. If no drawing default symbol is specified in the graphics data descriptor, the presentation default in AFP GOCA is solid fill, which is pattern symbol X'10' in the default pattern set.

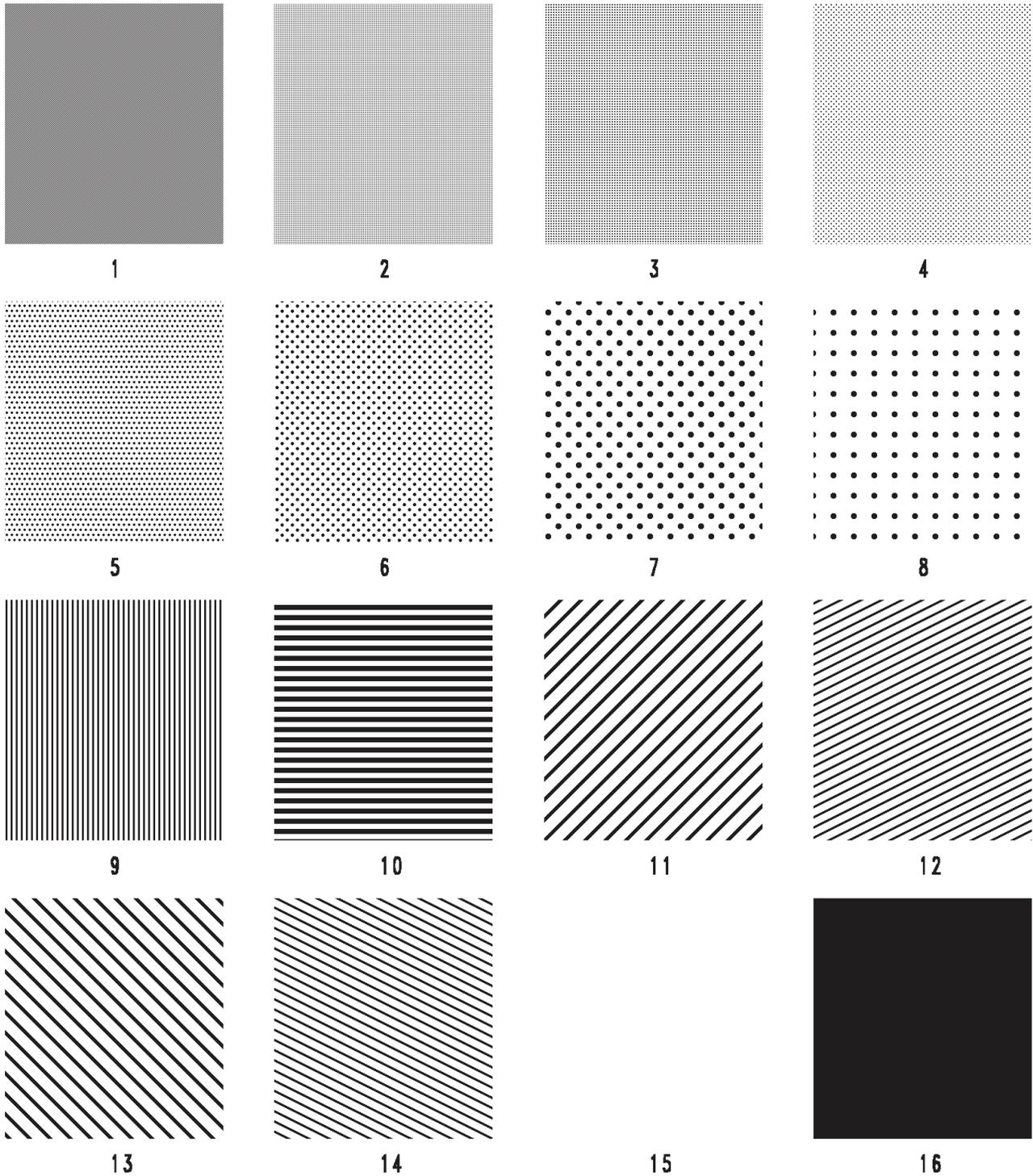
Patterns in the Default Pattern Set

The default pattern set contains predefined patterns. [Figure 19 on page 40](#) shows representative patterns corresponding to attribute values X'01' (1) to X'10' (16) in the default pattern set. A pattern set attribute value of X'00' specifies to use the default pattern set.

Note also that in the default pattern set, a pattern symbol attribute value of X'40' (blank) is treated the same as an attribute value of X'0F' (no fill).

Areas

Figure 19. Default Pattern Set



Implementation Note: Implementation differences with respect to dot size, dot spacing, line thickness, and line spacing exist. In particular, some printers use the same line thickness and line spacing for pattern #10 as for pattern #9. However, all future presentation devices should try to replicate the published patterns as closely as possible.

Architecture Note: The precise appearance of the patterns in [Figure 19 on page 40](#) in this edition of this Reference do not *exactly* match earlier editions of this Reference, although differences are very slight. Any differences are unintentional and do not constitute a changed definition of the patterns.

Custom Patterns

Custom patterns can be used to fill areas with any arbitrary pattern. A custom pattern is defined using GOCA drawing orders such that almost any GOCA picture can be used as a custom pattern.

The Begin Custom Pattern drawing order starts the definition of a custom pattern, with a matching End Custom Pattern drawing order ending the definition. The GOCA drawing orders between the Begin Custom Pattern and End Custom Pattern draw a picture, some window of which is the custom pattern to be used to fill areas. The edges of the window are specified in the Begin Custom Pattern drawing order.

A custom pattern definition may start in one segment and be completed in an appended segment.

Once a custom pattern has been defined in a segment, it can then be used to fill areas within that segment (including subsequent appended segments). Implementations must maintain the custom pattern definition throughout the segment, unless it is deleted using the Delete Pattern drawing order. The custom pattern cannot be used outside of the segment in which it is defined.

Once a custom pattern has been defined, it can be deleted using the Delete Pattern drawing order. After the Delete Pattern executes, the deleted custom pattern is no longer available for area fill.

As with any pattern, a custom pattern can be set as the default pattern using the Set Current Defaults instruction.

Custom Pattern Mode

When a Begin Custom Pattern drawing order is encountered, custom pattern mode is entered. While in custom pattern mode, all drawing is done in a separate, temporary GPS, and no drawing is done in the current GPS containing the picture corresponding to the current GOCA object—for discussion purposes, the GPS containing the picture drawn by the current GOCA object will be called the object GPS. The word “temporary” is used since once the End Custom Pattern drawing order is encountered, there is no way to draw into that GPS anymore. Note, however, that when drawing into the temporary GPS, all normal GOCA rules apply, such as raising an exception condition if drawing is attempted outside the GPS.

When a custom pattern is used to fill an area, the window from the temporary GPS is mapped one-to-one into the object GPS. For example, if the pattern window is of size 200x300 in the temporary GPS, it will be drawn at size 200x300 in the object GPS. The custom pattern will be tiled in all directions with the pattern reference point as origin, using as many tiles or partial tiles as are necessary to fill the area in question. The pattern reference point can be set using the Set Pattern Reference Point drawing order.

Before custom pattern mode is entered, all current attributes and the current position must be saved away by the drawing processor and then later restored when custom pattern mode is exited. In this way, adding a custom pattern definition into an existing GOCA segment will not change the picture drawn by that segment in any way (assuming the custom pattern is not used, of course).

Similarly, when custom pattern mode is entered, the current attributes and current position are set by the drawing processor to the drawing defaults—this is the same processing as is done at the beginning of a new segment. In this way, moving a custom pattern definition from one place in a segment to another will not affect the custom pattern definition in any way, since the custom pattern inherits nothing from the surrounding segment. The initialization when entering custom pattern mode is not affected by the presence of a segment prolog in the segment containing the custom pattern definition.

Custom Pattern Color Aspects

There are two types of custom patterns: bilevel and full-color. The only difference between the two is at which point color is applied to them.

Bilevel custom patterns are uncolored at definition time, then get a single color applied to them when they are used. This behavior is the same as that for the patterns in the default pattern set. In a bilevel custom pattern definition, any attempt to set a color is ignored, and the custom pattern is uncolored in the temporary GPS. When the bilevel custom pattern is used to fill an area, the current value of the pattern color attribute at the time the Begin Area drawing order is executed is used as the color of the custom pattern. Bilevel custom patterns, then, appear in only one color (possibly black).

Full-color custom patterns have all color applied to them during their definition, and can therefore contain any number of colors in them (including the possibility of containing only one color, possibly black). The temporary GPS corresponding to a full-color custom pattern contains all the colors to be used in the pattern. When a full-color custom pattern is used to fill an area, the colors in the pattern appear as in the custom pattern definition, and are not affected by the current value of the pattern color attribute.

The type of a custom pattern is specified in the Begin Custom Pattern drawing order that starts the definition of the custom pattern. Every custom pattern, then, is either a bilevel custom pattern or a full-color custom pattern, but never both, and a given custom pattern cannot change its type.

Valid Custom Pattern Definitions

Custom pattern definitions cannot be nested—if a Begin Custom Pattern drawing order is found in a custom pattern definition, exception condition EC-DE00 is raised. Also, custom patterns cannot be used to fill areas inside the definition of a custom pattern; an attempt to do so will cause exception EC-6806 to be raised.

Similarly, gradient definitions cannot be nested inside custom pattern definitions. If a Linear Gradient or Radial Gradient drawing order is found in a custom pattern definition, exception condition EC-DE07 is raised. As with custom patterns, gradients cannot be used in custom pattern definitions, and exception condition EC-6806 is raised if this is attempted.

Just as nested custom patterns or gradients cannot be created inside a custom pattern definition, custom patterns or gradients cannot be deleted inside a custom pattern definition. The Delete Pattern drawing order, then, is not valid between a Begin Custom Pattern order and its corresponding End Custom Pattern order; exception condition EC-DE07 is raised if such a Delete Pattern drawing order is found.

Other than the Begin Custom Pattern, Delete Pattern, Linear Gradient, and Radial Gradient drawing orders, all GOCA drawing orders are allowed between a Begin Custom Pattern drawing order and its corresponding End Custom Pattern drawing area, although normal GOCA rules apply; for example, if an End Prolog drawing order is found, exception condition EC-3E00 would be raised as it would anytime an End Prolog drawing order is found outside the segment prolog.

If, in custom pattern mode, the End Custom Pattern drawing order is not found before the end of the segment, exception condition EC-DE01 is raised. Note, however, that an appended segment is part of the segment that it appends.

The Set Color, Set Extended Color, and Set Process Color drawing orders are allowed in bilevel custom pattern definitions, but are ignored.

Gradients

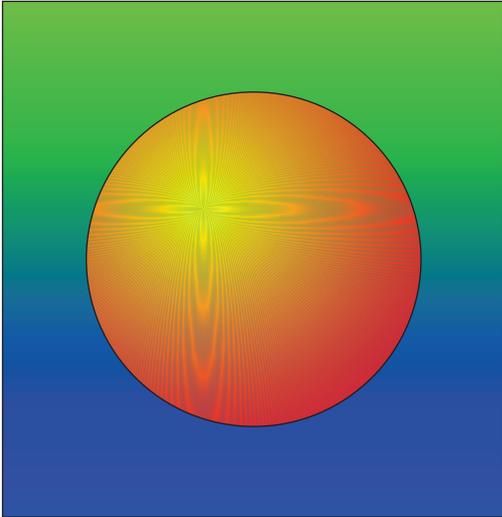
Gradients can be used to fill areas. A gradient is an area fill where one color gradually changes to another.

There are two types of gradients in AFP GOCA:

- Linear gradients
- Radial gradients

An example of a picture with both a linear gradient (from the top to the bottom in the background) and a radial gradient (on the circle) is shown in [Figure 20](#).

Figure 20. An Example of Both a Linear Gradient and a Radial Gradient

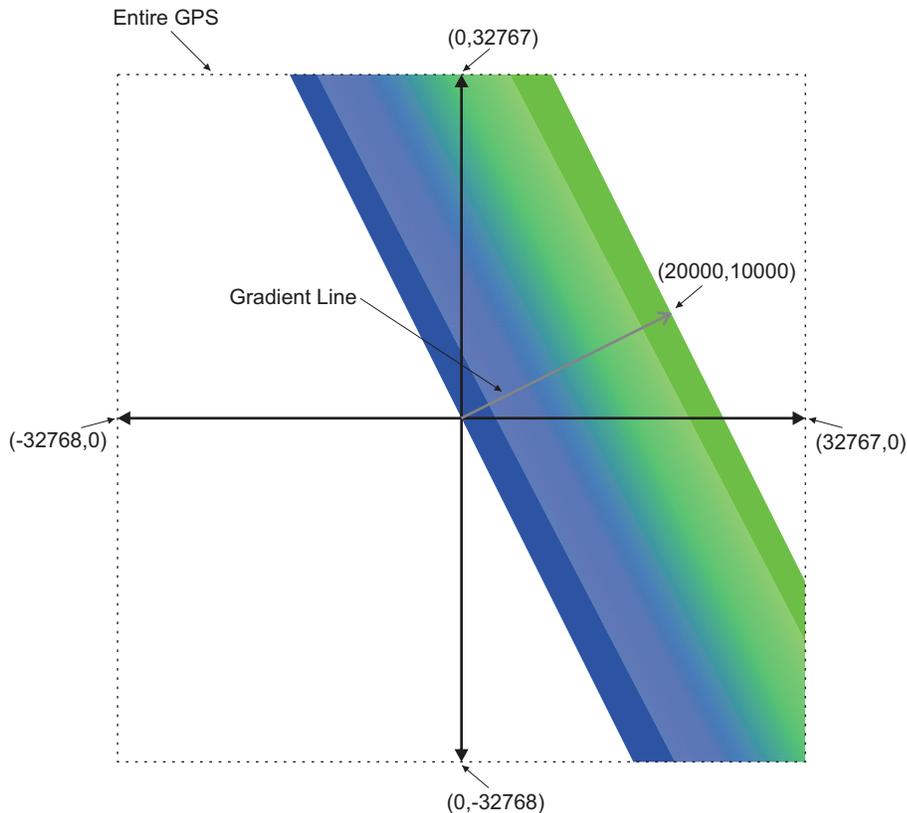


The two types of gradients are defined using the Linear Gradient and Radial Gradient drawing orders.

Linear Gradients

Linear gradients go from a start color located at a start point to an end color located at an end point. The gradient, then, proceeds along a line from the start point to the end point; this line will be called the gradient line. The colors radiate out in a perpendicular fashion from the gradient line such that the gradient becomes 2-dimensional. The color gradually changes, starting at the line that is perpendicular to the gradient line and that goes through the start point, and ending at the line that is perpendicular to the gradient line and that goes through the end point. A linear gradient, then, continues out to the edge of the GPS. As an example, [Figure 21](#) shows the entire GPS and what a gradient defined to go from blue at (0,0) to green at (20000,10000) looks like.

Figure 21. Linear Gradient Extending to the Edge of the GPS



As can be seen in [Figure 21](#), a gradient can be quite large in GPS terms, taking up a significant part of the entire GPS, but nonetheless there can be areas of the GPS that are outside the gradient. In [Figure 21](#), the outside areas are:

- The quadrilateral on the lower left. This area is considered to be on the “start side” of the gradient since it borders the start of the gradient line.
- The triangle on the upper right. This area is considered to be on the “end side” of the gradient.

Also included in the definition of a gradient are indications of how to fill these areas that are outside the gradient—these are the “Outside” values. There is one Outside value for the start side of the gradient and one Outside value for the end side, and the values do not have to be the same. The possible values:

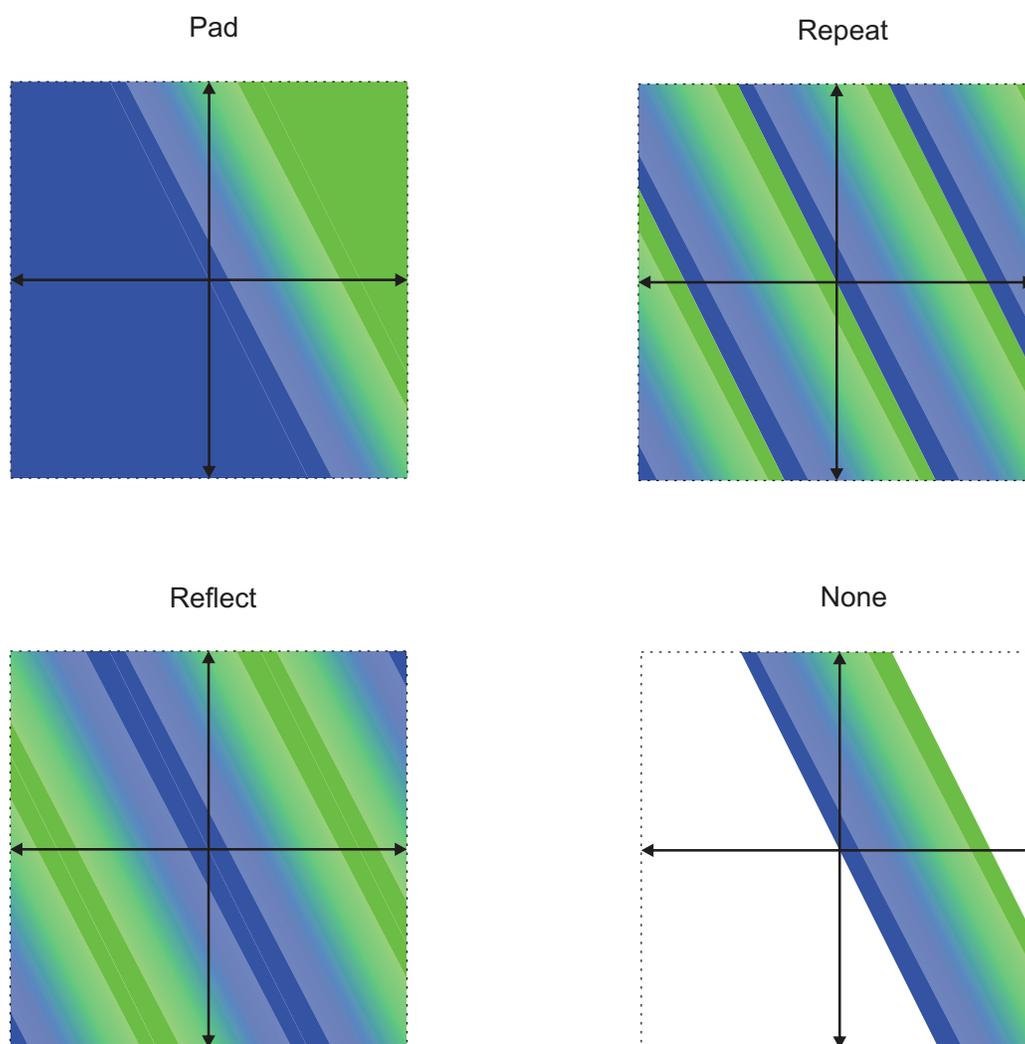
Outside value Meaning

Pad Extend the color on the edge of the gradient as far as necessary to fill the outside area. For example, for a blue to green gradient, the outside area on the start side would be all blue and the outside area on the end side would be all green.

- Repeat** The gradient is repeated as many times as necessary to fill the outside areas, by repeating the gradient line in both directions. For example, for a blue to green gradient, the blue to green gradient would be repeated over and over right next to the previous gradient.
- Reflect** Right next to the gradient, a mirror-image of the gradient is produced. If this does not fill the outside area, the gradient itself is then repeated, followed by another mirror image, followed by the same gradient, and so on. For example, for a blue to green gradient, a blue to green gradient followed immediately by a green to blue gradient would be drawn; continuing this, the end result would be a gradient going from blue to green to blue to green and so on to the edge of the GPS.
- None** Fill the outside areas with no color. This is equivalent to treating these areas as if they had been filled with the X'0F' (no fill) pattern of the default pattern set.

[Figure 22](#) shows the effect of each Outside value on the example gradient from [Figure 21 on page 44](#); in the figure, both the start and end Outside values are set to the same value.

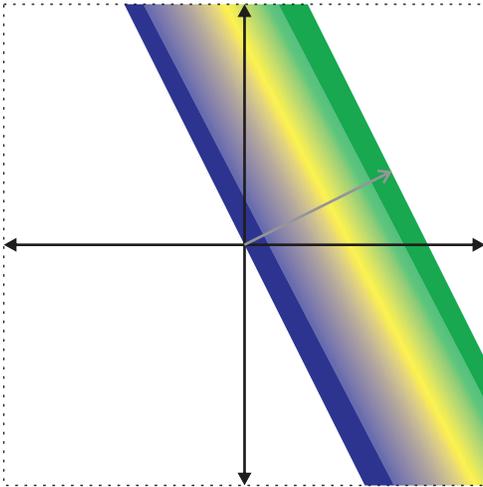
Figure 22. Effect of Different Outside Values



Areas

In addition to the gradient changing from the start color to the end color, additional colors can be specified to occur between the start and end. For example, rather than a gradient simply changing from blue to green, it can be defined that it also be yellow at some point in between. This point is called a color stop. A color stop is defined by the offset along the gradient line where it occurs, and the color to appear at that offset. Multiple color stops can be defined. As an example, [Figure 23](#) shows the example gradient from [Figure 21 on page 44](#) but with the addition of a yellow color stop at an offset of 60% of the way along the gradient line.

Figure 23. A Yellow Color Stop at Offset 60% Added to the Gradient from [Figure 21](#)



Two color stops at the same offset define a discontinuity of the gradient. For example, if two color stops are defined at halfway along the gradient line, the first yellow and the second purple, the gradient will smoothly change toward yellow in the first half of the gradient getting to yellow just before the halfway point, abruptly change to purple at the halfway point, then smoothly change away from purple in the second half of the gradient.

If any part of the gradient is specified to transition from some color *C* to that same color *C*, that part of the gradient will be drawn as solid fill with color *C*.

Radial Gradients

Radial gradients are conceptually very similar to linear gradients. They transition from a start color to an end color with any number of color stops in between. They have two Outside values to define how to fill areas outside the gradient. However, rather than the gradient changing color along a gradient line, radial gradients change color radiating from a start full arc to an end full arc. Conceptually, the color changes through an infinite number of intermediate full arcs that occur in the process of transitioning from the start full arc to the end full arc. For example, if the start full arc is smaller than the end full arc and its center is directly to the left of the end full arc's center, the first intermediate full arc between the start and the end will be a tiny bit larger than, and a tiny bit to the right of, the start full arc.

[Figure 24](#) shows a radial gradient transitioning from yellow at the inside, start full arc to red at the outside, end full arc. [Figure 25](#) shows a similar radial gradient, from yellow to red, but with overlapping start and end full arcs. [Figure 26](#) shows the same radial gradient with full arcs that are completely outside each other. Note that, as can be seen in figures [25](#) and [26](#), conceptually the intermediate full arcs are drawn beginning at the start full arc and ending at the end full arc, which results in a picture where the end full arc looks closer to the viewer.

Figure 24. A Simple Radial Gradient

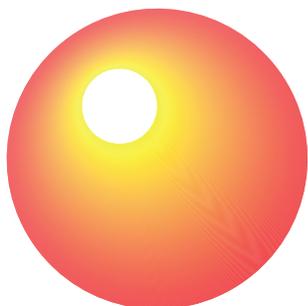


Figure 25. A Radial Gradient when the Start and End Full Arcs Overlap

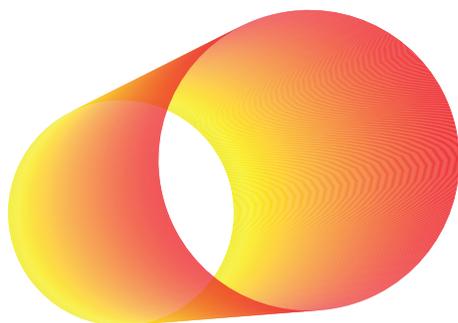
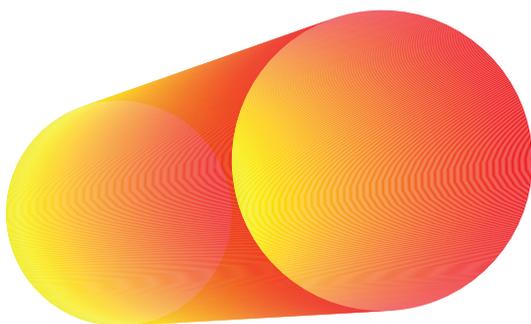


Figure 26. A Radial Gradient when the Start and End Full Arcs Are Outside Each Other



Color stops in radial gradients work the same as those in linear gradients, but note that a color stop at an offset of, for example 60%, defines the color of the intermediate full arc that is 60% of the way between the start full arc and end full arc.

Areas

There is an important difference between linear and radial gradients regarding the filling of areas outside the gradient. In cases where one of the full arcs is not contained inside the other full arc, as in [Figure 25 on page 47](#), a given intermediate full arc does not surround the previous intermediate full arc; instead it overlaps the previous intermediate full arc. The intermediate full arcs move in some specific direction rather than expanding in all directions; in [Figure 25 on page 47](#), the intermediate full arcs are always moving to the right and slightly upward. Thus it can be seen that if `Outside=pad`, the continuation of the gradient would only cause there to be more intermediate full arcs that would fill certain areas to the right and left, but would not fill the area above and below—this is shown in [Figure 27](#). Thus, a radial gradient, even when taking into account the `Outside` values, does not necessarily completely fill the GPS; that is, in some cases, there are parts of the GPS that are outside the gradient but that cannot be filled using the `Outside` values. Note that when one full arc is completely inside the other, as was the case for [Figure 24 on page 47](#), the `Outside` values *can* fill the entire GPS; see [Figure 28](#).

Figure 27. The Radial Gradient from [Figure 25](#) when Both Outside Values Are Pad

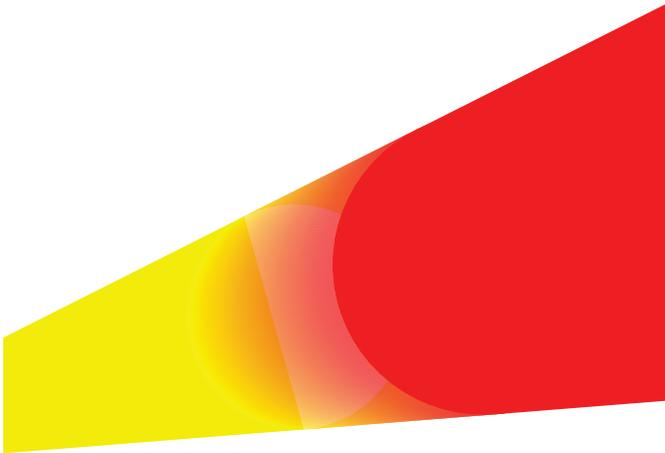
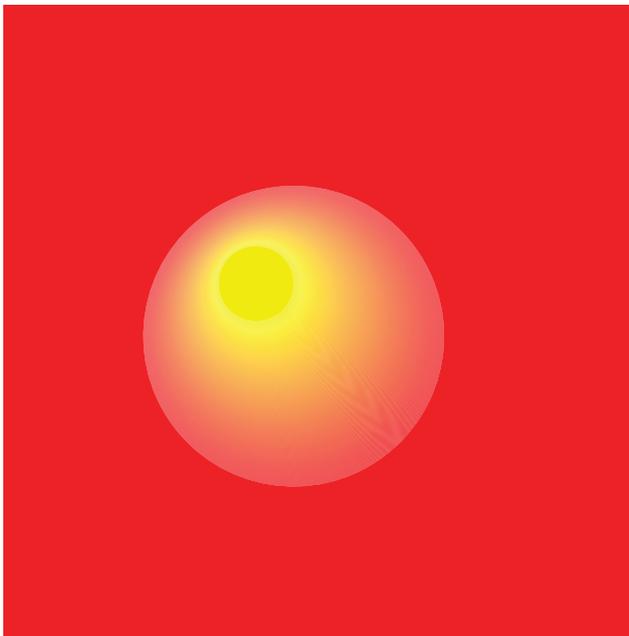


Figure 28. The Radial Gradient from [Figure 24](#) when Both Outside Values Are Pad



Gradients as Patterns

When a gradient is defined, it is assigned a pattern set value and a pattern symbol value that can be used to subsequently identify the gradient.

Once a gradient has been defined in a segment, it can then be used to fill areas within that segment (including subsequent appended segments). Implementations must maintain the gradient definition throughout the segment, unless it is deleted using the Delete Pattern drawing order. The gradient cannot be used outside of the segment in which it is defined.

Once a gradient has been defined, it can be deleted using the Delete Pattern drawing order. After the Delete Pattern executes, the deleted gradient is no longer available for area fill.

As with any pattern, a gradient can be set as the default pattern using the Set Current Defaults instruction.

Areas

Area (Pattern) Attributes

[Table 10](#) shows the attributes controlling the drawing of Area primitives.

Table 10. Attributes Controlling Area Primitives

Attribute	Standard Default	Length (in bytes)	Meaning
PATTERN SET	Default pattern set (X'00')	1	Specification of pattern set identifier
SYMBOL	Solid fill (X'10')	1	Specification of pattern symbol code point
COLOR	Device dependent	2	Color value set into GPS for foreground
PROCESS COLOR	Device dependent	12–14	Process color value set into GPS for foreground
MIX	Overpaint (X'02')	1	Specification of mix mode in GPS for foreground
BACKGROUND MIX	Leave Alone (X'05')	1	Specification of mix mode in GPS for background
REFERENCE POINT	(0,0)	4	Specification of pattern reference point for custom patterns

Character Strings

Two orders are supported for drawing character strings:

- The Character String at Given Position order draws a character string starting at a given point (X_0, Y_0), and sets the current position to (X_0, Y_0).
- The Character String at Current Position order draws a character string starting at the current position, and does not change the current position.

See [“Character String \(GCHST, GCCHST\) Orders” on page 92](#) for details.

The font from which the character definitions are to be obtained is given by the value of the character set attribute.

The color of all characters in the string is given by the value of the color or process color attribute.

The way in which characters in the string are merged with any output primitives that have already been drawn is controlled by the values of the character mix and character background mix attributes.

The current values of the line type, line width, pattern set, and pattern symbol attributes have no effect on the appearance of the characters in the string.

A character string can be defined in which some of its characters need to be drawn outside the boundaries of the GPS. The result of executing a Character String order where this occurs is implementation dependent.

The appearance and relative positions of the characters in the string are dependent on the values of:

- Code points in the order
- Character Attributes (see [“Character Attributes” on page 56](#))

A character string is drawn based upon the specified character precision:

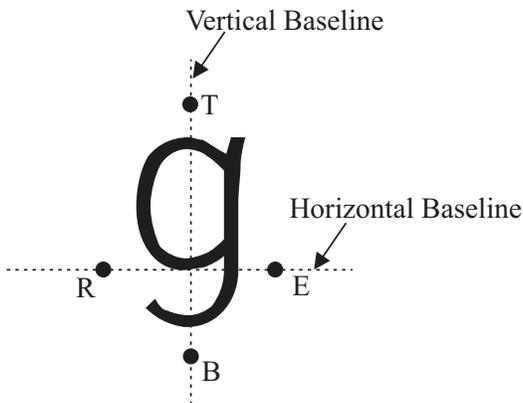
- Precision 1** Device-specific (string) precision
- Precision 2** Device-specific (character) precision
- Precision 3** Stroke precision (not supported in AFP GOCA)

Character Strings

Figure 29 shows two different methods used in AFP environments for positioning GOCA character strings.

Figure 29. Methods for Positioning Character Strings

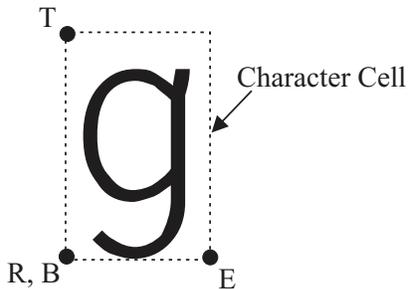
Font Positioning Method



In this case, the character cell is ignored and the size of the selected font determines the size of each character. Characters are placed using the metrics within this font.

R = Reference point for Left to Right, escapement point for Right to Left
T = Reference point for Top to Bottom, escapement point for Bottom to Top
B = Reference point for Bottom to Top, escapement point for Top to Bottom
E = Reference point for Right to Left, escapement point for Left to Right

Cell Positioning Method



In this case, the character cell size determines the size of the characters. Each character is placed within a character cell. Character cells are positioned using the character cell width and height.

Example

• Digits

Font Positioning Method

• D i g i t s

Cell Positioning Method

Device-Specific (Character) Precision

Character precision has been implemented differently on different devices; it is not consistent among implementations. The intent of this precision is that characters are positioned and drawn as follows. Note that the character reference point is not always placed at the current position. Scale and rotation are not necessarily applied when drawing the symbol.

1. The position of the first character is determined by the character direction attribute. Each device uses one of the two methods of locating the points R,E,T,B shown in [Figure 29 on page 52](#); refer to your device documentation for specific implementation information.
 - *When the character direction is left to right*, point R for the first character is positioned at the current or given position from the Character String order.
 - *When the character direction is right to left*, point E for the first character is positioned at the current or given position from the Character String order.
 - *When the character direction is top to bottom*, point T for the first character is positioned at the current or given position from the Character String order.
 - *When the character direction is bottom to top*, point B for the first character is positioned at the current or given position from the Character String order.
2. The character is then drawn taking the following character attributes into account:
 - For devices that scale GOCA characters, the symbol is scaled using the character cell-size attribute. This scaling is independent in the X-axis and Y-axis.
 Note that for some devices, if the character cell size is specified as negative values, a mirror image of the character is generated. That is, if the cell width is negative, the character is mirrored about the Y-axis, and if the cell height is negative, the character is mirrored about the X-axis.
 - The character cell is rotated by the angle given in the character angle attribute.
 - For some devices, the character is rotated within the cell based on the selected font rotation.
3. The next character in the string is positioned.
 - *When the character direction is left to right*, a vector is generated from the left edge of the character box to the right edge, and successive characters are placed in the direction of this vector.
 - *When the character direction is top to bottom*, a vector is generated from the top edge of the character box to the bottom edge, and successive characters are placed in the direction of this vector.
 - *When the character direction is right to left*, a vector is generated from the right edge of the character box to the left edge, and successive characters are placed in the direction of this vector.
 - *When the character direction is bottom to top*, a vector is generated from the bottom edge of the character box to the top edge, and successive characters are placed in the direction of this vector.
4. Subsequent characters in the string are positioned and drawn in the same manner. [Figure 30 on page 54](#) shows the effect of the character direction and character angle attributes when the device uses the font positioning method. [Figure 31 on page 55](#) shows the effect of the character direction and character angle attributes when the device uses the cell positioning method.

Character Strings

Figure 30. Font Positioning Method

Character Direction	Character Angle				Character Rotation
	0°	90°	180°	270°	
Left to Right					0°
Top to Bottom					270°
Right to Left					180°
Bottom to Top					90°

———— Character baseline

• Current graphics position - Character reference point of first character

————> Inline direction

The character rotation, which is specified in a font mapping in the controlling environment, is selected to match the specified Character Direction. For example in the MO:DCA environment, the character rotation is specified in a Map Coded Font structured field. In the IPDS environment, character rotation is determined by the specified Font Inline Sequence. The relationship between these two parameters is as follows:

0 degree character rotation = 0 degree font inline sequence
 90 degree character rotation = 270 degree font inline sequence
 180 degree character rotation = 180 degree font inline sequence
 270 degree character rotation = 90 degree font inline sequence

Figure 31. Cell Positioning Method

Character Direction	Character Angle				Character Rotation
	0°	90°	180°	270°	
Left to Right					0°
Top to Bottom					270°
Right to Left					180°
Bottom to Top					90°

———— Character baseline

• Current graphics position - Character reference point of first character

————> Inline direction

The character rotation, which is specified in a font mapping in the controlling environment, is selected to match the specified Character Direction. For example in the MO:DCA environment, the character rotation is specified in a Map Coded Font structured field. In the IPDS environment, character rotation is determined by the specified Font Inline Sequence. The relationship between these two parameters is as follows:

- 0 degree character rotation = 0 degree font inline sequence
- 90 degree character rotation = 270 degree font inline sequence
- 180 degree character rotation = 180 degree font inline sequence
- 270 degree character rotation = 90 degree font inline sequence

Character Strings

Device-Specific (String) Precision

String precision has been implemented differently on different devices; it is not consistent among implementations. String precision differs from character precision in the following respects:

- The character angle attribute can be ignored.
- The positioning of the first character can be approximate.
- Drawing of the symbol need take no account of scale or rotation.
- The positioning of further characters in the string need not be scaled according to the character cell-size attribute.

Character Attributes

[Table 11](#) shows the attributes controlling the appearance of character strings.

Table 11. Attributes Controlling Character String Primitives

Attribute	Standard Default	Length (in bytes)	Meaning
ANGLE X,Y	No rotation	4	Character rotation x and y values
CELLSIZE	Device dependent	4	Specification of character cell width and height
CELLSIZEF	Device dependent	4	Specification of fractional extension of character cell width and height
DIRECTION	Left to right (X'01')	1	Specification of character direction
PRECISION	Character precision (X'02')	1	Specification of character precision
CHARACTER SET	Device dependent	1	Specification of character set local identifier
SHEAR X,Y	No shear	4	Shear x and y values; not supported in AFP GOCA
SYMBOL	Device dependent	1	Specification of default character code point
COLOR	Device dependent	2	Color value set into GPS for foreground
PROCESS COLOR	Device dependent	12–14	Process color value set into GPS for foreground
MIX	Overpaint (X'02')	1	Specification of mix mode in GPS for foreground
BACKGROUND MIX	Leave Alone (X'05')	1	Specification of mix mode in GPS for background

Markers

A marker is a symbol that is used to indicate a position. It is similar to a character drawn at a specified (X_g , Y_g) position. When a marker is used, however, the center of the cell in which the marker is drawn is placed at the specified position.

Two orders are provided for drawing markers:

- The Marker at Given Position order draws a marker at one or more points.
- The Marker at Current Position order draws a marker at the current position and at one or more further points.

The particular symbol that is drawn is the one identified by the current marker symbol from the current marker set. This symbol is drawn at all the positions specified in the one order. The only marker set supported in AFP GOCA is the default marker set, shown in [Figure 32](#).

Figure 32. Default Marker Set

X'00'	Drawing default
X'01'	×
X'02'	+
X'03'	◇
X'04'	□
X'05'	*
X'06'	*
X'07'	◆
X'08'	■
X'09'	•
X'0A'	○
X'40'	(blank)

The color of all markers drawn by an order is given by the value of the current marker color. The way in which markers are merged with any output primitives that have already been drawn is controlled by the values of the marker mix and marker background mix attributes.

The current values of the line type, line width, pattern set, and pattern symbol attributes have no effect on the appearance of the markers.

A marker symbol whose position is inside the GPS, but is placed such that part of the marker lies outside the GPS, is not an error. The appearance of that marker in the GPS is implementation dependent.

Markers are drawn taking into account all marker attributes, and are centered at the specified position.

Markers are scaled along with the rest of the GPS if scaling is necessary in the mapping from the GPS window into the usable area (object area).

Implementation note: In earlier versions of AFP GOCA, there existed a marker precision attribute. This attribute has been made obsolete; for more details, see [Appendix C, “AFP GOCA Migration Functions”, on page 195](#). The marker precision attribute allowed implementations to draw markers taking into account only the marker symbol and marker set attributes, and the positioning of the marker could be approximate. Some implementations, then, might draw markers in such a way. However, all implementations that do not treat the Set Marker Cell (GSMC) drawing order as a No-Op must:

- Draw markers taking into account all marker attributes.

Markers

- Center the marker at the specified position.
- Treat the marker precision attribute as obsolete.
- Follow the recommendation for the standard default value for marker cell-size specified in [“Set Marker Cell \(GSMC\) Order” on page 152](#).

Marker Attributes

[Table 12](#) shows the attributes controlling the appearance of markers.

Table 12. Attributes Controlling Marker Primitives

Attribute	Standard Default	Length (in bytes)	Meaning
CELLSIZE	Device dependent	4	Specification of marker cell width and height
MARKER SET	Device dependent	1	Specification of marker set local identifier
SYMBOL	Cross (X'01')	1	Specification of marker symbol code point
COLOR	Device dependent	2	Color value set into GPS for foreground
PROCESS COLOR	Device dependent	12–14	Process color value set into GPS for foreground
MIX	Overpaint (X'02')	1	Specification of mix mode in GPS for foreground
BACKGROUND MIX	Leave Alone (X'05')	1	Specification of mix mode in GPS for background

Images

Images are rectangular arrays of points that are included directly in the graphics picture definition. An image is represented by a sequence of orders. The first order is a Begin Image order, which is followed by one or more Image Data orders. The sequence *must* end with an End Image order.

Note: The only other orders permitted within the Begin Image/End Image order bracket are the No-Operation and Comment orders.

Only one format of image data is defined: FORMAT=X'00'. With this format, each Image Data order contains the data for one row of the image.

Image points are mapped to presentation-device pels. The size of the image is given by the WIDTH and HEIGHT parameters in the Begin Image order. There must be as many Image Data orders as the HEIGHT parameter, and each Image Data order must contain the number of bits specified by the WIDTH parameter, plus padding to a byte boundary.

The position of the image in GPS is specified in GPS L-units.

The current values of the image attributes are taken into account when drawing the image. An image must be completely defined in one segment. However, it may start in one segment and be completed in an appended segment.

Image Attributes

[Table 13](#) shows the attributes controlling the appearance of an image.

Table 13. Attributes Controlling Image Primitives

Attribute	Standard Default	Length (in bytes)	Meaning
COLOR	Device dependent	2	Color value set into GPS for foreground
PROCESS COLOR	Device dependent	12–14	Process color value set into GPS for foreground
MIX	Overpaint (X'02')	1	Specification of mix mode in GPS for foreground
BACKGROUND MIX	Leave Alone (X'05')	1	Specification of mix mode in GPS for background

Output Primitive Overflow

It is possible to define an output primitive such that it starts inside the GPS, but some part of it lies outside the GPS.

If a primitive, such as the Full Arc or Partial Arc primitive, starts inside the GPS, ends inside the GPS, leaves the current position in the GPS, but goes outside the GPS for some part of its path, exception EC-000D is raised. The standard action for exception EC-000D is to draw the primitive in an implementation-dependent manner. For presentation devices that cannot maintain a position outside the GPS, such as devices that are limited to 2-byte arithmetic, this exception is mandatory. For devices that can maintain a position outside the GPS, it is optional.

An error is also generated when a primitive, such as the Relative Line or Partial Arc primitive, is specified that causes the current position to fall outside the GPS. In that case a drawing process check is generated and there is no standard action. For the Relative Line primitive, the exception is EC-E100. For the Partial Arc primitive, it is EC-E300. For presentation devices that cannot maintain a position outside the GPS, these exceptions are mandatory. For devices that can maintain a position outside the GPS, they are optional.

A coordinate point that is outside the GPS is characterized by an arithmetic overflow in its X_g or Y_g coordinate. Because AFP GOCA uses 16-bit signed integers to specify GPS coordinates, a point outside the GPS is characterized by a 2-byte arithmetic overflow. Note that this does not mean that AFP GOCA processors are limited to 2-byte arithmetic. It simply means that they need to be able to detect 2-byte arithmetic overflows. For a definition of the geometric parameter format used in AFP GOCA, see [“Parameter Type” on page 71](#) and [“Drawing Order Subset” on page 181](#).

Chapter 5. Segments

Segments are self-contained collections of drawing orders. This chapter describes:

- Segment types
- Segment processing sequence
- Segment properties
- Segment prolog

Segment Types

Segments of the following types can be created, as determined by the CHAIN parameter in the Begin Segment command:

- Chained
- Unchained

In AFP GOCA, all segments are processed in *immediate mode*. In this mode, chained segments define the picture. They are processed by the drawing processor as they are received from the environment interface, and are *not* stored. Unchained segments are ignored.

Segments are transmitted by the controlling environment to the drawing processor with Begin Segment Commands and the drawing orders that follow these commands. The Begin Segment (chained) command invokes the drawing processor to draw the segment.

In the MO:DCA and IPDS environments, a graphics object can contain multiple chained segments. All chained segments within the object are processed independently in the sequence in which they arrive; together they generate the graphics picture. A segment cannot be split across multiple graphics objects.

The Append option indicates that the segment is a continuation of the preceding segment. Unfinished drawing orders, areas, images, and prologs may be completed in appended segment data. See [“Begin Segment Command” on page 75](#) for further details of the functions of the Append option.

Segment Processing Sequence

Segment processing starts at the first segment in the segment chain. The processing of a segment always starts at its first order and proceeds in sequence, order by order, until the last order is processed, at which time the segment is terminated.

When the invocation operates on a single segment, it is complete when the segment is terminated.

When the invocation operates on a chain of segments, the graphics processor sequentially processes and terminates each segment in the chain. When the last specified chained segment is terminated, the invocation is complete.

Segment Properties

Associated with each segment is a set of properties. These properties are specified in the Begin Segment command; see [“Begin Segment Command” on page 75](#). The function of these properties is to provide control information relevant to the processing of the segment.

The properties and their functions are as follows:

Property	Function
Length	2-byte length of segment data.
Chain	Indicates whether or not the segment is to be chained.
Prolog	Indicates that the segment has a prolog section at the beginning of the data. See Segment Prolog for details of prolog processing.
New/Append	Indicates whether this is a new segment or a segment to be appended to the previous segment.

These properties are unique to the segment. They are not inherited between segments. They are defined when a segment is created.

Segment Prolog

Segment prologs provide a defined position where initial attributes and drawing process controls are set. A prolog is optional; its presence is indicated by the PROLOG bit in the Begin Segment command.

If present, the prolog is always at the beginning of the segment, and is ended by an End Prolog order within the same segment. Note that for a segment that is spread over multiple appended segments with multiple Begin Segment commands, the End Prolog drawing order may be specified in any of the appended segments.

The end of a prolog in a segment must be indicated by an End Prolog order. Exception condition EC-000C is raised if the end of the segment is reached while still in the prolog.

Note: Exception condition EC-3E00 is raised if an End Prolog order is found when not in a prolog.

Within the prolog, only the following orders are valid:

- Comment
- No-Operation
- Segment Characteristics
- Set Arc Parameters
- Set Background Mix
- Set Character Angle
- Set Character Cell
- Set Character Direction
- Set Character Precision
- Set Character Set
- Set Character Shear
- Set Color
- Set Current Position
- Set Custom Line Type
- Set Extended Color
- Set Fractional Line Width
- Set Line End

- Set Line Join
- Set Line Type
- Set Line Width
- Set Marker Cell
- Set Marker Precision (obsolete, see [Appendix C, “AFP GOCA Migration Functions”, on page 195](#))
- Set Marker Set
- Set Marker Symbol
- Set Mix
- Set Pattern Reference Point
- Set Pattern Set
- Set Pattern Symbol
- Set Process Color

Implementation Note: Some AFP printers also accept the Set Pick Identifier order in a prolog, and process this order as a No-Op.

The other supported orders, when specified in the prolog, cause exception condition EC-000C to be raised.

Segment Prolog Semantics

The semantics of the segment prolog for chained segments processed in immediate mode are as follows.

The segment data is processed by the graphics processor following processing of a Begin Segment command. For an appended segment—that is, one specified with the APP flag equal to B'11'—the segment data that follows the Begin Segment command is part of the segment, not the whole segment.

For an Immediate mode chained segment that is spread over multiple appended segments with multiple Begin Segment commands, only the PROLOG flag bit in the Begin Segment that is marked as *New* determines whether the segment has a prolog or not. The PROLOG bits in subsequent Begin Segment commands for appended segments are ignored.

For a segment that is spread over multiple appended segments with multiple Begin Segment commands, it is necessary for the graphics processor to check the segment data in all the Begin Segment commands for the segment before it can determine whether the segment is valid or not, that is, whether or not the segment contains an End Prolog order to match the setting of the segment prolog property.

For example, with a segment whose data is spread over three appended segments with three Begin Segment commands, the first command could indicate that the segment has a prolog, but the End Prolog order could well be in data in the third segment. The PROLOG flag in the first Begin Segment command of a multi-part segment does not indicate that an End Prolog order is necessarily contained within the segment data that follows the Begin Segment command.

Chapter 6. Environment Controls

This chapter describes:

- Control instructions
- Drawing processor facilities, including
 - Current attributes
 - Drawing process controls

Control Instructions

The controlling environment communicates with the graphics processor by means of control instructions and drawing process controls. Control instructions are embedded in environment-dependent carriers in AFP GOCA as follows:

- When the graphics object is carried in a MO:DCA data stream, the carrier is a Graphics Data Descriptor (GDD) structured field; for more information, see [Appendix A, “Mixed Object Document Content Architecture \(MO:DCA\) Environment”, on page 179](#).
- When the graphics object is carried in an IPDS data stream, the carrier is a Graphics Data Descriptor self-defining field in the Write Graphics Control (WGC) command; for more information, see [Appendix B, “Intelligent Printer Data Stream \(IPDS\) Environment”, on page 189](#).

Both the GDD and WGC contain the Set Current Defaults control instruction, defined in [“Set Current Defaults \(SCD\) Instruction” on page 66](#).

Note: In the MO:DCA environment, if the drawing defaults contain any invalid bits, the processor optionally raises exception condition EC-000A.

Set Current Defaults Instruction

Set Current Defaults (SCD) Instruction

This control instruction sets the current default values of the selected attributes and drawing process controls. When the graphics object is carried in a MO:DCA data stream, this control instruction is contained in the Graphics Data Descriptor (GDD) structured field. When the graphics object is carried in an IPDS data stream, this control instruction is contained in the Graphics Data Descriptor (GDD) self-defining field of the Write Graphics Control (WGC) command.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4– <i>n</i>	Length of following data
2		SET	X'00'	Drawing attributes
			X'01'	Line attributes
			X'02'	Character attributes
			X'03'	Marker attributes
			X'04'	Pattern attributes
			X'0B'	Arc parameters
			X'10'	Process color attributes
			X'11'	Normal line width attribute
			All others	Reserved
3–4	CODE	MASK	X'0000'–X'FFFF'	Set mask
5	BITS	FLAGS		
Bit 0		DEFAULT	B'0',B'1'	B'0' Set all indicated items to their standard default values B'1' Set the indicated items to the values contained in the source data
Bits 1–3		RES1	B'000'	Reserved; only valid value
Bits 4–7		RES2	B'1111'	Reserved; only valid value
6– <i>n</i>		DATA		Default values; bytes 6 onward are not present if DEFAULT=B'0'

This instruction permits the setting of a variable number of values, under control of the MASK parameter in bytes 3–4, into the attribute set selected by the value of the SET parameter in byte 2. When a MASK bit equals 0, the default does not change and data bytes are not present for that attribute. A B'1' in any bit of MASK indicates that the corresponding item is to be set. If the DEFAULT bit is B'0', these items are set to the standard defaults; if it is B'1', these items are set to the values contained in the data in bytes 6–*n* of the instruction.

If the value of an attribute specifies the drawing default in an attribute setting order, for example the X'00' value of the MODE parameter used in the Set Mix order, it causes the current default to be set to the standard default value.

Bits 0–15 in MASK correspond to items within the selected attribute set, as shown in the following tables. The number of bytes required is set into the item corresponding to each 1 bit in Mask, in ascending numerical order of the MASK bit (0–15). Setting is terminated when all the items requested have been loaded.

The default value of a given attribute should be specified only once. If specified more than once, the results are implementation dependent; it is recommended that future implementations use the last-specified value.

Notes:

1. When the integral part of the line width attribute is set, the fractional part is reset to zero. See [“Line Attributes” on page 29](#) for a description of the Line Width attribute.
2. The format of the DATA field is the same as the corresponding data in the attribute setting drawing orders.

Drawing Attributes (SET=X'00')

Mask bit	Item name	Length (bytes)
0	Color	2
1	Reserved; must be zero	N/A
2	Foreground mix	1
3	Background mix	1
4–15	Reserved; must be zeros	N/A

Note: Setting any of the above attributes to a value is a shorthand way of setting all color, or mix, attributes to the same value.

Line Attributes (SET=X'01')

Mask bit	Item name	Length (bytes)
0	Line type	1
1	Line width	1
2	Line end	1
3	Line join	1
4–15	Reserved; must be zeros	N/A

Note: The line type attribute cannot be set to a custom value by this instruction.

Character Attributes (SET=X'02')

Mask bit	Item name	Length (bytes)
0	Angle X,Y	4
1	Cell-size CW,CH	4
2	Direction	1
3	Precision	1
4	Character Set	1
5	Shear, X,Y	4
6–15	Reserved; must be zeros	N/A

Note: The character symbol default attribute is *not* settable by this instruction.

Set Current Defaults Instruction

Marker Attributes (SET=X'03')

Mask bit	Item name	Length (bytes)
0	Reserved; must be zero	N/A
1	Marker cell-size width, height	4
2	Reserved; must be zero	N/A
3	Marker precision (obsolete, see Appendix C, "AFP GOCA Migration Functions", on page 195)	1
4	Marker set	1
5–6	Reserved; must be zeros	N/A
7	Marker symbol	1
8–15	Reserved; must be zeros	N/A

Pattern Attributes (SET=X'04')

Mask bit	Item name	Length (bytes)
0–3	Reserved; must be zeros	N/A
4	Pattern Set	1
5–6	Reserved; must be zeros	N/A
7	Pattern Symbol	1
8–10	Reserved; must be zeros	N/A
11	Pattern Reference Point	4
12–15	Reserved; must be zeros	N/A

Arc Parameters (SET=X'0B')

Mask bit	Item name	Length (bytes)
0	P value	2
1	Q value	2
2	R value	2
3	S value	2
4–15	Reserved; must be zeros	N/A

Process Color Attributes (SET=X'10')

Mask bit	Item name	Length (bytes)
0	Foreground mix	1
1	Background mix	1
2	Process Color	12-14
3–15	Reserved; must be zeros	N/A

Architecture Note: If the color is specified using Drawing Attributes - Set = X'00', and the process color is also specified using Process Color - Set = X'10', the last-specified color is used.

Normal Line Width Attribute (SET=X'11')

Mask bit	Item name	Length (bytes)
0	Normal line width	2
1–15	Reserved; must be zeros	N/A

Architecture Note: If the normal line width attribute is specified, it establishes the absolute value of the normal line width in 1440ths of an inch. If the line attributes are also specified and define the line width as a multiple of the normal line width, the multiple is calculated based on the absolute value of the normal line width. Furthermore, all Set Line Width and Set Fractional Line Width orders in the object are also calculated based on the absolute value of the normal line width.

Instruction Process Checks

A check condition is set under the following conditions:

- IPC-0002**
 - If the SET parameter (byte 2) is invalid or unsupported
 - If the FLAGS parameter (byte 5) bits 1–3 are not B'000', or bits 4–7 are not B'1111'
 - If an unallocated item is referenced in the MASK parameter (bytes 3–4)
- IPC-0003**
 - If the FLAGS parameter (byte 5) bit 0 is B'0' and LENGTH is not X'04'
 - If the FLAGS parameter (byte 5) bit 0 is B'1' and the length of the immediate data (byte 6 onward) does not exactly match the length implied by the MASK parameter
- IPC-0021**
 - If any values in the data are invalid or unsupported

Drawing Processor Facilities

The following facilities are available to the drawing processor while it is processing segments:

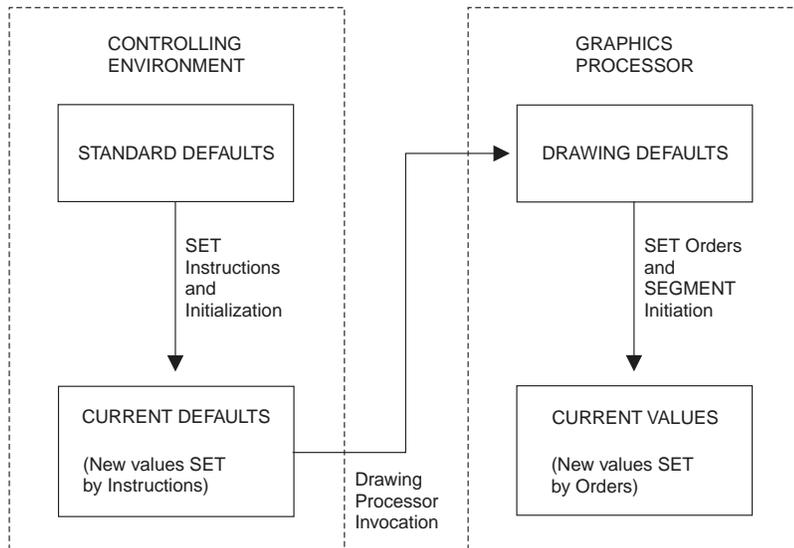
- Current attributes
- Drawing process controls

Current Attributes

As the orders in a segment are processed, the drawing processor maintains the current values of all primitive attribute types in the current attributes. These values are used by the graphics processor to draw output primitives in the GPS.

[Figure 33](#) shows how the controlling environment uses pre-defined standard defaults and the Set Current Defaults control instruction in the GDD and WGC to establish drawing defaults before the drawing processor is invoked to process a segment. At the start of processing of each new segment, the drawing default values are set into the current attributes.

Figure 33. Attributes and Drawing Process Control



Drawing Process Controls

The following controls manage various aspects of the drawing process:

Parameter type The format of the parameters in the drawing orders. These controls are described in [“Parameter Type” on page 71](#).

Arc parameters Values used as parameters when drawing circles or ellipses. These controls are described in [“Set Arc Parameters \(GSAP\) Order” on page 130](#).

Drawing defaults exist for each drawing process control. The defaults are maintained by the processor, and they are set to the standard defaults, or to the current defaults provided by the environment, whenever the processor is invoked.

Parameter Type

The parameter type specifies the format of the parameters within drawing orders, and has two parts:

- Coordinate type
- Geometric parameter format

The format of this control is:

Mnemonic	Standard default	Length (in bytes)	Meaning
COORD	X'00'	1	Coordinate type
GEOM	X'00'	1	Geometric parameter format

The following value of the coordinate type parameter is specified by both the GRS3 and DR/2V0 subsets and is supported in AFP GOCA environments:

Value	Meaning
X'00'	2-D coordinates

The following value of the geometric parameter format is specified by both the GRS3 and DR/2V0 subsets and is supported in AFP GOCA environments:

Value	Meaning
X'00'	16-bit signed integer, "high byte first" format

Chapter 7. Commands and Drawing Orders

This chapter describes:

- The Begin Segment command
- Order formats
- Order alignment
- Current position in drawing orders
- Coordinate data
- Offset data
- Default values for drawing orders and attributes
- The following drawing orders:
 - Begin Area
 - **Begin Custom Pattern**
 - Begin Image at Given Position
 - Begin Image at Current Position
 - Box at Given Position
 - Box at Current Position
 - Character String at Given Position
 - Character String at Current Position
 - Comment
 - Cubic Bezier Curve at Given Position
 - Cubic Bezier Curve at Current Position
 - **Delete Pattern**
 - End Area
 - **End Custom Pattern**
 - End Image
 - End Prolog
 - Fillet at Given Position
 - Fillet at Current Position
 - Full Arc at Given Position
 - Full Arc at Current Position
 - Image Data
 - Line at Given Position
 - Line at Current Position
 - **Linear Gradient**
 - Marker at Given Position
 - Marker at Current Position
 - No-Operation
 - Partial Arc at Given Position
 - Partial Arc at Current Position
 - **Radial Gradient**
 - Relative Line at Given Position
 - Relative Line at Current Position
 - Segment Characteristics
 - Set Arc Parameters
 - Set Background Mix
 - Set Character Angle
 - Set Character Cell
 - Set Character Direction
 - Set Character Precision

- Set Character Set
- Set Character Shear
- Set Color
- Set Current Position
- Set Custom Line Type
- Set Extended Color
- Set Fractional Line Width
- Set Line End
- Set Line Join
- Set Line Type
- Set Line Width
- Set Marker Cell
- Set Marker Set
- Set Marker Symbol
- Set Mix
- Set Pattern Reference Point
- Set Pattern Set
- Set Pattern Symbol
- Set Process Color

Begin Segment Command

This command defines or modifies a segment, its properties, or both. It can be transmitted as part of the picture chain or as an unchained segment. If the segment is transmitted as part of the chain it is executed as received, then discarded. If a segment is transmitted as unchained, it is ignored.

Note: There is no formal End Segment command, and none is needed. However, the reserved fixed two-byte drawing order with order code X'71' is used as an End Segment drawing order by some applications. This order is treated as a No-Op in AFP GOCA.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'70'	Begin Segment command
1	UBIN	LENGTH	X'0C'	Length of following parameters
2–5	UBIN	NAME	Any value	Name of segment to be created; ignored
6	BITS	FLAG1	Any value	Ignored
7	BITS	FLAG2		Segment Properties 2
Bit 0		CHAIN	B'0',B'1'	B'0' Chained B'1' Unchained
Bits 1–2		CHPOS	Any value	Ignored
Bit 3		PROLOG	B'0',B'1'	B'0' No Prolog B'1' Prolog
Bit 4			Any value	Ignored
Bits 5–6		APP	B'00',B'01',B'10', or B'11'	B'00' New segment B'01' Reserved B'10' Reserved B'11' Append the specified data to the end of the existing segment
Bit 7		DATAFL	Any value	Ignored
8–9	UBIN	SEGL	X'0000'–X'FFFF'	Segment data length
10–13	CHAR	P/SNAME	Any value	Predecessor/successor name; ignored

The segment data in the form of drawing orders follows immediately in the following format:

Offset	Type	Name	Range	Meaning
0– <i>n</i>		SEGDATA		Segment data; the number of bytes is equal to the number defined by SEGL

Semantics

This command defines a segment for immediate execution. It consists of a command code, a command length, and a parameters section. The command is followed by a data section that contains the drawing orders for the segment. Bytes 0–13 of this command are often referred to as the *Begin Segment Introducer (BSI)*.

Begin Segment Command

Byte 0	The command code is X'70' for a Begin Segment command.																
LENGTH	Specifies the length of the parameter section, that is, it does not include the length of the command code or its own length. This length must be X'0C'.																
FLAG2	Contains the CHAIN, PROLOG, and APP flags, which are used to set various segment properties for the specified segment. <table><tr><td>CHAIN</td><td>Indicates whether this segment is chained or not. Chained segments define the picture, unchained segments are ignored in AFP GOCA.</td></tr><tr><td>PROLOG</td><td>Indicates whether the segment has a prolog section at the beginning of the segment data.</td></tr><tr><td>APP</td><td>Provides the following functions:<table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>B'00'</td><td><i>New Segment.</i> In chained immediate mode, if the segment starts a chain of appended segments, the prolog property is saved. The segment data in a new segment is processed with the drawing defaults.</td></tr><tr><td>B'01'</td><td><i>Reserved.</i> The command is rejected.</td></tr><tr><td>B'10'</td><td><i>Reserved.</i> The command is rejected.</td></tr><tr><td>B'11'</td><td><i>Append data to segment.</i> In chained immediate mode, if the segment is the first in a graphics object, the command can be rejected; if accepted, it is treated as a <i>new</i> segment. In chained immediate mode, if the segment is not the first in a graphics object, the segment data is considered to be a continuation of the data that followed the Begin Segment command immediately prior to the current Begin Segment command. This data is invoked with the <i>current</i> values of the drawing attributes, not with the drawing defaults, and uses the value of the prolog property that was saved in a previous New Segment command. This means that a prolog that is started in a new segment may be terminated with an End Prolog order in one of the appended segments. The appended segment data may possibly complete any unfinished drawing order, area, image definition, or prolog.</td></tr></tbody></table></td></tr></table>	CHAIN	Indicates whether this segment is chained or not. Chained segments define the picture, unchained segments are ignored in AFP GOCA.	PROLOG	Indicates whether the segment has a prolog section at the beginning of the segment data.	APP	Provides the following functions: <table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>B'00'</td><td><i>New Segment.</i> In chained immediate mode, if the segment starts a chain of appended segments, the prolog property is saved. The segment data in a new segment is processed with the drawing defaults.</td></tr><tr><td>B'01'</td><td><i>Reserved.</i> The command is rejected.</td></tr><tr><td>B'10'</td><td><i>Reserved.</i> The command is rejected.</td></tr><tr><td>B'11'</td><td><i>Append data to segment.</i> In chained immediate mode, if the segment is the first in a graphics object, the command can be rejected; if accepted, it is treated as a <i>new</i> segment. In chained immediate mode, if the segment is not the first in a graphics object, the segment data is considered to be a continuation of the data that followed the Begin Segment command immediately prior to the current Begin Segment command. This data is invoked with the <i>current</i> values of the drawing attributes, not with the drawing defaults, and uses the value of the prolog property that was saved in a previous New Segment command. This means that a prolog that is started in a new segment may be terminated with an End Prolog order in one of the appended segments. The appended segment data may possibly complete any unfinished drawing order, area, image definition, or prolog.</td></tr></tbody></table>	Value	Description	B'00'	<i>New Segment.</i> In chained immediate mode, if the segment starts a chain of appended segments, the prolog property is saved. The segment data in a new segment is processed with the drawing defaults.	B'01'	<i>Reserved.</i> The command is rejected.	B'10'	<i>Reserved.</i> The command is rejected.	B'11'	<i>Append data to segment.</i> In chained immediate mode, if the segment is the first in a graphics object, the command can be rejected; if accepted, it is treated as a <i>new</i> segment. In chained immediate mode, if the segment is not the first in a graphics object, the segment data is considered to be a continuation of the data that followed the Begin Segment command immediately prior to the current Begin Segment command. This data is invoked with the <i>current</i> values of the drawing attributes, not with the drawing defaults, and uses the value of the prolog property that was saved in a previous New Segment command. This means that a prolog that is started in a new segment may be terminated with an End Prolog order in one of the appended segments. The appended segment data may possibly complete any unfinished drawing order, area, image definition, or prolog.
CHAIN	Indicates whether this segment is chained or not. Chained segments define the picture, unchained segments are ignored in AFP GOCA.																
PROLOG	Indicates whether the segment has a prolog section at the beginning of the segment data.																
APP	Provides the following functions: <table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>B'00'</td><td><i>New Segment.</i> In chained immediate mode, if the segment starts a chain of appended segments, the prolog property is saved. The segment data in a new segment is processed with the drawing defaults.</td></tr><tr><td>B'01'</td><td><i>Reserved.</i> The command is rejected.</td></tr><tr><td>B'10'</td><td><i>Reserved.</i> The command is rejected.</td></tr><tr><td>B'11'</td><td><i>Append data to segment.</i> In chained immediate mode, if the segment is the first in a graphics object, the command can be rejected; if accepted, it is treated as a <i>new</i> segment. In chained immediate mode, if the segment is not the first in a graphics object, the segment data is considered to be a continuation of the data that followed the Begin Segment command immediately prior to the current Begin Segment command. This data is invoked with the <i>current</i> values of the drawing attributes, not with the drawing defaults, and uses the value of the prolog property that was saved in a previous New Segment command. This means that a prolog that is started in a new segment may be terminated with an End Prolog order in one of the appended segments. The appended segment data may possibly complete any unfinished drawing order, area, image definition, or prolog.</td></tr></tbody></table>	Value	Description	B'00'	<i>New Segment.</i> In chained immediate mode, if the segment starts a chain of appended segments, the prolog property is saved. The segment data in a new segment is processed with the drawing defaults.	B'01'	<i>Reserved.</i> The command is rejected.	B'10'	<i>Reserved.</i> The command is rejected.	B'11'	<i>Append data to segment.</i> In chained immediate mode, if the segment is the first in a graphics object, the command can be rejected; if accepted, it is treated as a <i>new</i> segment. In chained immediate mode, if the segment is not the first in a graphics object, the segment data is considered to be a continuation of the data that followed the Begin Segment command immediately prior to the current Begin Segment command. This data is invoked with the <i>current</i> values of the drawing attributes, not with the drawing defaults, and uses the value of the prolog property that was saved in a previous New Segment command. This means that a prolog that is started in a new segment may be terminated with an End Prolog order in one of the appended segments. The appended segment data may possibly complete any unfinished drawing order, area, image definition, or prolog.						
Value	Description																
B'00'	<i>New Segment.</i> In chained immediate mode, if the segment starts a chain of appended segments, the prolog property is saved. The segment data in a new segment is processed with the drawing defaults.																
B'01'	<i>Reserved.</i> The command is rejected.																
B'10'	<i>Reserved.</i> The command is rejected.																
B'11'	<i>Append data to segment.</i> In chained immediate mode, if the segment is the first in a graphics object, the command can be rejected; if accepted, it is treated as a <i>new</i> segment. In chained immediate mode, if the segment is not the first in a graphics object, the segment data is considered to be a continuation of the data that followed the Begin Segment command immediately prior to the current Begin Segment command. This data is invoked with the <i>current</i> values of the drawing attributes, not with the drawing defaults, and uses the value of the prolog property that was saved in a previous New Segment command. This means that a prolog that is started in a new segment may be terminated with an End Prolog order in one of the appended segments. The appended segment data may possibly complete any unfinished drawing order, area, image definition, or prolog.																
SEGL	Specifies the length of the data section. A zero length for the data section is invalid when creating a <i>new</i> segment, but this does not cause an exception in AFP GOCA.																
SEGDATA	The data section for the segment. This parameter contains the drawing orders.																

Exception Conditions

The actual mechanism by which these Communication Process Checks are reported is environment dependent.

CPC-0001	Invalid command code specified.
CPC-7001	APP has the value B'10'.
CPC-7082	APP has the value B'01'.
CPC-70C1	Invalid parameter length specified.
CPC-70C5	Insufficient data. Segment data is less than length specified by SEGL.

Order Formats

Drawing orders are represented in one of four formats, depending on the length of the operand data:

- Fixed 1-byte format is used for orders that have no operand data.
- Fixed 2-byte format is used for orders that have a single byte of operand data.
- Long format is used for orders with up to 255 bytes of operand data.
- Extended format is used for orders with up to 65,535 bytes of operand data.

The format of an order is determined by its order code:

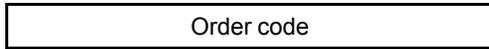
- One fixed 1-byte order has an order code of X'00'.
- For fixed 2-byte orders, bit 0 is set to 0, and bit 4 is set to 1, that is, the first digit of the order code is less than 8, and the second digit is greater than, or equal to, 8.
- Orders that are not **any of the other three formats** are long format orders.
- Extended orders have an order code of X'FE', which introduces the extended format.

Order Formats

Fixed 1-Byte Format

The fixed 1-byte order format has a 1-byte order code only. One order has this format: the No-Operation (GNOP1) order.

Figure 34. Fixed 1-Byte Order Format



Fixed 2-Byte Format

The fixed 2-byte format has a 1-byte order code and one byte of data:

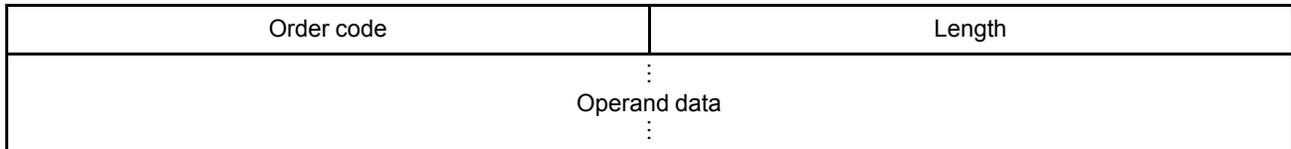
Figure 35. Fixed 2-Byte Order Format



Long Format

The long format has a 1-byte order code and a 1-byte length field followed by the number of data bytes specified by the length field. The value of the length field is the length of the operand data in bytes; that is, the length does not include the order code or length fields.

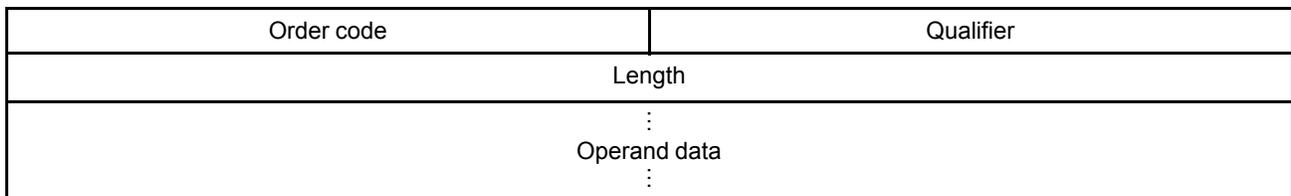
Figure 36. Long Order Format



Extended Format

The extended format has a 1-byte order code, X'FE', and a 1-byte qualifier field, followed by a 2-byte length field, followed by the number of data bytes specified by the length field. The value of the length field is the length of the operand data, in bytes; that is, the length does not include the order code, qualifier, or length fields.

Figure 37. Extended Order Format



Order Alignment

Orders can be followed by any number of No-Operation (GNOP1) or Comment (GCOMT) orders, to align the next order on any convenient byte boundary. Orders can be aligned on any byte boundary, although, depending on the implementation, a performance benefit can be obtained if orders are aligned on even-byte boundaries. Note that a drawing order may start in one segment and be completed in an appended segment.

Current Position in Drawing Orders

Some orders have two forms. One form uses the current position as one of its coordinate values; the other form does not. See [“Current Position” on page 19](#) for more details.

Coordinate Data

Coordinate data is used in orders to specify points in GPS, each point being specified by a set of two parameters, X_g and Y_g . The sequence of parameters in coordinate data is (X_g, Y_g) ; the format of the parameters is 16-bit twos-complement signed binary integers (SBIN). The drawing processor interprets coordinate data and raises an exception condition if the length of the data is not consistent with complete specification of points.

Offset Data

Offset data is used in orders to specify a point in GPS by specifying its offset from a given point. Each point is specified by a set of two parameters. The sequence of parameters in offset data is x,y ; the format of the parameters is 8-bit twos-complement signed binary integers (SBIN).

Default Values

The defaulting mechanism used is as follows. See also [Figure 33 on page 70](#).

- All current attributes and drawing process controls are set to their default values when the environment containing the graphics processor is initialized. These default values are referred to as the *standard defaults*.

The standard defaults are either:

- Environment-dependent values.
 - Architected values, that is, one of the values that can be selected with a nonzero attribute value has been architected as the default.
- The standard defaults are copied into another set of defaults, referred to as the *current defaults*, when the graphics processor is initialized.
 - The current defaults can be changed by the Set Current Defaults control instruction.
 - When a drawing process is initiated, the current defaults are copied into a set of defaults called the *drawing defaults*. These are the defaults that are assumed during the execution of the drawing process.

The drawing defaults apply during the whole of the drawing process. They cannot be changed by a control instruction while the drawing process is being executed.

- The current values of the primitive attributes are either set or propagated at the start of a segment. [“Current Attributes” on page 70](#) describes how the initial values are determined.

Summary List of Orders

- The current values of the drawing process controls are either set or propagated at the start of a segment. [“Drawing Process Controls” on page 70](#) describes how the initial values are determined.

Summary List of Orders

Length	Hex	Order	Meaning
1-byte	X'00'	GNOP1	No-Operation
Long	X'01'	GCOMT	Comment
Long	X'04'	GSGCH	Segment Characteristics
2-byte	X'08'	GSPS	Set Pattern Set
2-byte	X'0A'	GSCOL	Set Color
2-byte	X'0C'	GSMX	Set Mix
2-byte	X'0D'	GSBMX	Set Background Mix
Long	X'11'	GSFLW	Set Fractional Line Width
2-byte	X'18'	GSLT	Set Line Type
2-byte	X'19'	GSLW	Set Line Width
2-byte	X'1A'	GSLE	Set Line End
2-byte	X'1B'	GSLJ	Set Line Join
Long	X'20'	GSCLT	Set Custom Line Type
Long	X'21'	GSCP	Set Current Position
Long	X'22'	GSAP	Set Arc Parameters
Long	X'26'	GSECOL	Set Extended Color
2-byte	X'28'	GSPT	Set Pattern Symbol
2-byte	X'29'	GSMT	Set Marker Symbol
Long	X'33'	GSCC	Set Character Cell
Long	X'34'	GSCA	Set Character Angle
Long	X'35'	GSCH	Set Character Shear
Long	X'37'	GSMC	Set Marker Cell
2-byte	X'38'	GSCS	Set Character Set
2-byte	X'39'	GSCR	Set Character Precision
2-byte	X'3A'	GSCD	Set Character Direction
2-byte	X'3B'	GSMP	Set Marker Precision (<i>obsolete, see Appendix C, “AFP GOCA Migration Functions”, on page 195</i>)
2-byte	X'3C'	GSMS	Set Marker Set
2-byte	X'3E'	GEPROL	End Prolog
2-byte	X'5E'	GECP	End Custom Pattern
Long	X'60'	GEAR	End Area
2-byte	X'68'	GBAR	Begin Area

Length	Hex	Order	Meaning
Long	X'80'	GCBOX	Box at Current Position
Long	X'81'	GCLINE	Line at Current Position
Long	X'82'	GCMRK	Marker at Current Position
Long	X'83'	GCCHST	Character String at Current Position
Long	X'85'	GCFLT	Fillet at Current Position
Long	X'87'	GCFARC	Full Arc at Current Position
Long	X'91'	GCBIMG	Begin Image at Current Position
Long	X'92'	GIMD	Image Data
Long	X'93'	GEIMG	End Image
Long	X'A0'	GSPRP	Set Pattern Reference Point
Long	X'A1'	GCRLINE	Relative Line at Current Position
Long	X'A3'	GCPARC	Partial Arc at Current Position
Long	X'A5'	GCCBEZ	Cubic Bezier Curve at Current Position
Long	X'B2'	GSPCOL	Set Process Color
Long	X'C0'	GBOX	Box at Given Position
Long	X'C1'	GLINE	Line at Given Position
Long	X'C2'	GMRK	Marker at Given Position
Long	X'C3'	GCHST	Character String at Given Position
Long	X'C5'	GFLT	Fillet at Given Position
Long	X'C7'	GFARC	Full Arc at Given Position
Long	X'D1'	GBIMG	Begin Image at Given Position
Long	X'DE'	GBCP	Begin Custom Pattern
Long	X'DF'	GDPT	Delete Pattern
Long	X'E1'	GRLINE	Relative Line at Given Position
Long	X'E3'	GPARC	Partial Arc at Given Position
Long	X'E5'	GCBEZ	Cubic Bezier Curve at Given Position
Extended	X'FEDC'	GLGD	Linear Gradient
Extended	X'FEDD'	GRGD	Radial Gradient

Architecture Note: Some AFP printers accept the following drawing orders and process them as No-Ops:

- Set Pick Identifier (GSPIK, X'43'). This drawing order is in long format.
- End Segment drawing order (X'71'). This drawing order is in fixed 2-byte format, where the second byte is reserved and should be set to X'00'.

Begin Area (GBAR) Order

This order indicates the start of a set of primitives that define an area boundary.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'68'	GBAR order code
1	BITS	FLAGS		Internal flags
Bit 0		RES1	B'1'	Reserved for migration; only valid value
Bit 1		BOUNDARY	B'0', B'1'	Boundary-line draw indicator: B'0' Do not draw boundary lines B'1' Draw boundary lines
Bit 2		INSIDE	B'0', B'1'	Mode to determine inside: B'0' Alternate mode B'1' Nonzero Winding mode
Bits 3–7		RES2	B'00000'	Reserved; only valid value

Semantics

The Begin Area order starts the definition of a filled area. The area definition *must* be terminated by an End Area order. See [“Areas” on page 36](#) for details of the area definition.

The BOUNDARY bit determines whether or not the boundary of the area is drawn. The INSIDE bit determines the method of filling the interior.

The pattern set, pattern symbol, pattern mix, and pattern background mix attributes that *are current when the Begin Area order is executed* are used to fill the area. When using the default pattern set or a bilevel custom pattern (but not when using a full-color custom pattern or gradient), the pattern color attribute that is current when the Begin Area order was executed is also used to fill the area. When using a custom pattern, the pattern reference point attribute that is current when the Begin Area order was executed is also used in generating the fill pattern. The line attributes that are current at the time the orders defining the boundary are executed are used to draw the boundary.

The value of the current position is not changed by the Begin Area order itself, but is changed by those orders used to define the area boundary, including any implicit figure-closing orders that are required.

Area orders must *not* be nested and an area *must* be defined completely within a single segment.

For an area within an immediate-mode segment, temporary storage can be required. An exception condition, EC-6805, is set if this storage overflows.

RES1 is set to B'1' for compatibility with old implementations. Generators must set this value, but receivers can ignore this bit.

Implementation Note: The Nonzero Winding mode value of the INSIDE bit was added to the architecture later than the Alternate mode value. As such, not all GOCA receivers support Nonzero Winding mode, and some receivers will ignore the new value and use Alternate mode instead. Other receivers might report the retired exception condition EC-0002 if passed a Begin Area drawing order with INSIDE=B'1'.

Exception Conditions

The following exception conditions raise a drawing process check:

- EC-6800** A Begin Area order has been executed after another Begin Area order, without an intervening End Area order being executed.
- EC-6801** A Begin Area order has been executed in a segment, and the end of the segment is reached without an End Area order having been executed.
Note: This exception condition is raised when the end of the segment is reached.
- EC-6802** A supported order that is invalid within an area is detected.
Note: This exception condition is raised when the order that is invalid is detected.

The following exception conditions cause a standard action to be taken:

- EC-6803** The pattern set identified by the current pattern set is not supported. This exception condition exists even if the current pattern symbol is set to X'00'.
Standard action: The standard default pattern set is used.
- EC-6804** The current pattern symbol identifies an undefined symbol in the current pattern set.
Standard action: The standard default pattern symbol is used. In AFP environments this is X'10'—Solid fill.
- EC-6805** Temporary storage overflow while drawing an area in an immediate segment.
Standard action: Drawing of the area is completed in an implementation-defined manner.
- EC-6806** While inside the definition of a custom pattern, a Begin Area order has been executed but the current pattern symbol identifies a custom pattern or gradient.
Standard action: The standard default pattern symbol is used. In AFP environments this is X'10'—Solid fill.

Begin Custom Pattern (GBCP) Order

This order indicates the start of a set of primitives that define a custom pattern.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'DE'	GBCP order code
1	UBIN	LENGTH	13	Length of following data
2–3		RES1	X'0000'	Reserved; only valid value
4	BITS	FLAGS		Custom Pattern flags
Bit 0		FULLCOLOR	B'0' B'1'	Bilevel custom pattern Full-color custom pattern
Bits 1–7		RES2	B'0000000'	Reserved; only valid value
5	CODE	PATTSET	X'01' – X'FD'	Pattern set of the custom pattern
6	CODE	PATTSYM	X'01' – X'FF'	Pattern symbol of the custom pattern
7–8	SBIN	XLWIND	X'8000' – X'7FFF'	X _g coordinate for left edge of the pattern window
9–10	SBIN	XRWIND	X'8000' – X'7FFF'	X _g coordinate for right edge of the pattern window
11–12	SBIN	YBWIND	X'8000' – X'7FFF'	Y _g coordinate for bottom edge of the pattern window
13–14	SBIN	YTWIND	X'8000' – X'7FFF'	Y _g coordinate for top edge of the pattern window

Semantics

The Begin Custom Pattern order starts the definition of a custom pattern. The custom pattern *must* be terminated by an End Custom Pattern order.

The FULLCOLOR flag specifies whether the custom pattern being defined is a bilevel custom pattern or full-color custom pattern. Both types must be supported by drawing processors. A custom pattern cannot change from bilevel to full-color or vice versa.

See [“Custom Patterns” on page 41](#) for details of custom patterns.

The PATTSET and PATTSYM values specify the pattern set and pattern symbol where this custom pattern will reside. When the current values of the pattern set and pattern symbol attributes specify these PATTSET and PATTSYM values, respectively, this custom pattern will be used to do area fill.

Custom patterns (defined with this order), linear gradients (defined with the Linear Gradient order), and radial gradients (defined with the Radial Gradient order) share the pattern sets X'01'–X'FD'; the patterns using these pattern sets can be any combination of custom patterns and gradients without restriction.

Definition of a custom pattern using this drawing order does not set the pattern set and pattern symbol attributes to use the custom pattern. To use the custom pattern, the pattern set and pattern symbol attributes must be explicitly set.

All possible valid values of the PATTSET and PATTSYM fields must be supported. There are $253 \cdot 255 = 64,515$ possible combinations of PATTSET and PATTSYM. However, it is not required that implementations

support the creation of 64,515 custom patterns. If insufficient storage is available to correctly handle a custom pattern definition, exception condition EC-DE06 is raised, for which the standard action is to skip the custom pattern definition.

The XLWIND, XRWIND, YBWIND, and YTWIND values specify the coordinates of the edges of the pattern window, in GPS units, in the temporary GPS used to draw the pattern. Only the parts of the picture drawn by the drawing orders in the custom pattern definition that lie within the pattern window make up the custom pattern. Any drawing done outside the window is discarded without raising an exception condition. If the pattern window values are specified in such a way as to define a zero- or negative-sized window, exception condition EC-DE04 is raised, for which the standard action is to skip the custom pattern definition.

Custom pattern definitions must *not* be nested and a custom pattern *must* be defined completely within a single segment. Similarly, a previously defined custom pattern cannot be used in the definition of a new custom pattern; an attempt to do so will cause exception EC-6806 to be raised.

It is not possible to replace a custom pattern or gradient simply by sending a Begin Custom Pattern with the same PATTSET and PATTSYM parameters. If this is attempted, exception EC-DE05 is raised, for which the standard action is to skip the custom pattern definition. To replace a pattern at a given pattern set and pattern symbol, first delete the existing pattern using the Delete Pattern drawing order.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

- EC-DE00** A Begin Custom Pattern order has been executed after another Begin Custom Pattern order, without an intervening End Custom Pattern order being executed.
Standard action: Act as if an End Custom Pattern order had been received just prior to the current Begin Custom Pattern order.
- EC-DE01** A Begin Custom Pattern order has been executed in a segment, and the end of the segment is reached without an End Custom Pattern order having been executed.
Note: This exception condition is raised when the end of the segment is reached.
Standard action: Act as if an End Custom Pattern order had been received just prior to the end of the segment.
- EC-DE02** The value specified for the PATTSET parameter is invalid.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE03** The value specified for the PATTSYM parameter is invalid; a custom pattern cannot be assigned to pattern symbol X'00'.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE04** The pattern window values are ill defined: XLWIND is equal to or greater than XRWIND, or YBWIND is equal to or greater than YTWIND.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE05** The PATTSET and PATTSYM values are within the valid range, but a pattern already resides at that location.

Begin Custom Pattern

Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.

EC-DE06

Insufficient storage available to process and store a custom pattern.

Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.

EC-DE07

A supported order that is invalid within a custom pattern definition is detected.

Note: The exception condition is raised when the order that is invalid is detected.

Standard action: Ignore the invalid order.

Begin Image (GBIMG, GCBIMG) Orders

These orders identify the start of an image definition at a given position or at the current position.

Syntax

Begin Image at Given Position (GBIMG) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'D1'	GBIMG order code
1	UBIN	LENGTH	10	Length of following data
2–3	SBIN	XPOS	X'8000'–X'7FFF'	X _g coordinate of image origin (first image point of first image scan line)
4–5	SBIN	YPOS	X'8000'–X'7FFF'	Y _g coordinate of image origin (first image point of first image scan line)
6	CODE	FORMAT	X'00'	Format of the image data: X'00' Each image point is mapped to a presentation device pel
7		RES	X'00'	Reserved; only valid value
8–9	UBIN	WIDTH	X'0000'–X'FFFF'	Width of the image data, in image points
10–11	UBIN	HEIGHT	X'0000'–X'FFFF'	Height of the image data, in rows, or scan lines

Begin Image at Current Position (GCBIMG) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'91'	GCBIMG order code
1	UBIN	LENGTH	6	Length of following data
2	CODE	FORMAT	X'00'	Format of the image data: X'00' Each image point is mapped to a presentation device pel
3		RES	X'00'	Reserved; only valid value
4–5	UBIN	WIDTH	X'0000'–X'FFFF'	Width of the image data, in image points
6–7	UBIN	HEIGHT	X'0000'–X'FFFF'	Height of the image data, in rows, or scan lines

Semantics

The Begin Image at Given Position (GBIMG) order defines an image at the specified position. The Begin Image at Current Position (GCBIMG) order defines an image at the current position.

An image consists of a rectangular region of points and is defined by a sequence of orders, starting with a Begin Image order and ending with an End Image order. Between these two orders are one or more Image Data, Comment, or No-Operation orders, and these are the only orders permitted.

The XPOS and YPOS parameters define the position of the image origin—that is, the first point of the first row—which is at the top-left corner of the image. This origin is defined in GPS units.

Begin Image

If a particular bit in the image data is B'1', it defines a foreground pel, and the point is drawn using the current values of the image mix and color attributes. If the bit is B'0', it defines a background pel, and the point is drawn using the current value of the image background mix attribute.

There is an Image Data order for every row of the image. That is, for an image n rows high there *must* be n Image Data orders between the Begin Image and End Image orders. Each Image Data order contains sufficient integral bytes of data for the width of the image. If the width of the image is *not* an integral number of bytes, the padding bits at the right-hand end of the last byte in the Image Data order are ignored.

When the image is not contained in a custom pattern definition, each image point is mapped to a presentation device pel, unless the image resolution is explicitly specified in the Graphics Data Descriptor; see [“Window Specification \(Mandatory\)” on page 182](#). If the image resolution is explicitly specified, the mapping may include resolution correction so that the image is presented at its original size.

Implementation Note: In the absence of any other image resolution information, AFP printers map image points to device pels as follows:

- Printers that have a fixed resolution map point-to-pel at that resolution.
- Printers that have an *acceptance mode* for a fixed resolution map point-to-pel at the acceptance-mode resolution and then scale to the device resolution.
- Printers that have a fixed resolution but scale transparently to a different device resolution map point-to-pel at the fixed resolution and then scale to the device resolution.
- Printers that support multiple raster source resolutions map point-to-pel to the single (maximum) device resolution reported in the IPDS XOH-OPC IM-Image and Coded-Font Resolution self-defining field. Such printers normally also provide acceptance modes at lower resolutions, so that if the GOCA image size is too small at the device resolution, the customer can switch to a lower-resolution acceptance mode.
- The image is not scaled when a scale-to-fit or scale-to-fill mapping to the object area is specified for the graphics object.

When the image is contained in a custom pattern definition, each image point is mapped into the temporary GPS in a 1-to-1 manner, such that, for example, an image that is 100x200 image points would appear in the custom pattern definition's temporary GPS at size 100x200 GPS units.

The image is drawn in HEIGHT rows, each row having WIDTH image points. Each row is drawn by taking sequential bits from the corresponding Image Data order, and drawing them in sequential points left-to-right in the horizontal direction. The first row, first Image Data order, starts at the image origin point. Each subsequent row (HEIGHT–1 successive Image Data orders) starts at a point adjacent to the first point of the previous row, in the top-to-bottom direction. The first row of the image is oriented parallel to the GPS X_g axis, and subsequent rows are generated in the negative Y_g direction.

Architecture Notes:

1. The practical limit for the WIDTH parameter range is 2040, which is the maximum number of bits in the Image Data order.
2. Some presentation devices support a smaller range than X'0000'–X'FFFF' for the HEIGHT parameter.

Current position is set to the image origin. The current values of the color, mix, and background mix attributes are taken into account when drawing the image.

An image definition must be completely within one segment, that is, the Begin Image, Image Data, and End Image orders that define a particular image *must* all be in one segment. Note that a segment may consist of a new segment followed by one or more appended segments. Together they are treated as one segment; therefore, an image definition may start in a new or appended segment and finish in an appended segment.

Exception Conditions

The following exception conditions raise a drawing process check:

- EC-0003** The order has an incorrect length.
- EC-D100** A Begin Image order has been executed in a segment, and the end of the segment is reached without an End Image order having been executed.
- Note:** This exception condition is raised when the end of the segment is reached.
- EC-D101** A Begin Image order has been executed in a segment, and a supported order other than a Comment, No-Operation, Image Data, or End Image order is executed.
- Note:** This exception condition is raised when the order that is invalid is detected.
- EC-D102** The value specified for the FORMAT parameter is not supported.

The following exception conditions cause a standard action to be taken:

- EC-D103** The value specified for the WIDTH parameter is too large to allow the environment to completely present the image.
- Standard action:** The width of the image is truncated to allow presentation of the smaller image.
- EC-D104** The value specified for the HEIGHT parameter is too large to allow the environment to completely present the image.
- Standard action:** The height of the image is truncated to allow presentation of the smaller image.

Box (GBOX, GCBOX) Orders

These orders define a box with square or round corners, drawn with its first corner at a given position or at the current position.

Syntax

Box at Given Position (GBOX) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'C0'	GBOX order code
1	UBIN	LENGTH	10, 12, 14	Length of following data
2-3		RES	X'2000'	Reserved; only valid value
4-5	SBIN	XPOS0	X'8000'–X'7FFF'	X _g coordinate of first corner
6-7	SBIN	YPOS0	X'8000'–X'7FFF'	Y _g coordinate of first corner
8-9	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of diagonal corner
10-11	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of diagonal corner
The following parameters are optional:				
12-13	UBIN	HAXIS	0–32,767	Full length of x-direction axis of ellipse for rounded corner
14-15	UBIN	VAXIS	0–32,767	Full length of y-direction axis of ellipse for rounded corner

Box at Current Position (GCBOX) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'80'	GCBOX order code
1	UBIN	LENGTH	6, 8, 10	Length of following data
2-3		RES	X'2000'	Reserved; only valid value
4-5	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of diagonal corner
6-7	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of diagonal corner
The following parameters are optional:				
8-9	UBIN	HAXIS	0–32,767	Full length of x-direction axis of ellipse for rounded corner
10-11	UBIN	VAXIS	0–32,767	Full length of y-direction axis of ellipse for rounded corner

Semantics

The Box at Given Position (GBOX) order defines a rectangular box with square or rounded corners with its first corner specified by the first coordinate pair, and the diagonally-opposite corner specified by the second coordinate pair. The Box at Current Position (GCBOX) order defines a rectangular box with square or rounded

corners with its first corner at the current position, and the diagonally-opposite corner specified by the first coordinate pair. The sides of the rectangle are drawn parallel to the GPS X_g, Y_g axes.

For the GBOX order, the box is drawn in a counterclockwise direction in GPS if the value of $(XPOS1-XPOS0)/(YPOS1-YPOS0)$ is positive, and in a clockwise direction otherwise. For the GCBOX order, the box is drawn in a counterclockwise direction in GPS if the value of $(XPOS1-x_{CP})/(YPOS1-y_{CP})$ is positive, and in a clockwise direction otherwise, where (x_{CP}, y_{CP}) are the coordinates of the current position.

Implementation Note: For historical reasons, not all GOCA receivers support the directionality of boxes; therefore, some receivers will draw all boxes in a counterclockwise direction.

If this drawing order is in an area definition, the box is treated as a closed figure. The BOUNDARY parameter in the Begin Area order determines whether the boundary of the box is drawn.

If HAXIS and VAXIS are omitted, or HAXIS or VAXIS is zero, a rectangular box with square corners is drawn with its first corner at the current position (GCBOX), or at $(XPOS0, YPOS0)$ (GBOX). The diagonally-opposite corner is at $(XPOS1, YPOS1)$. If the values of HAXIS and VAXIS are nonzero, and HAXIS is not equal to VAXIS, a similar rectangle is drawn, except that the corners are replaced by quarter ellipses whose *full* axes are HAXIS and VAXIS. IF $HAXIS = VAXIS$, the corners are quadrants of a circle whose diameter is HAXIS. If VAXIS is omitted, it is assumed to be equal to HAXIS.

Note: HAXIS cannot be omitted without omitting VAXIS as well.

The current values of the line attributes, except for line join, are taken into account when drawing the boundary of the box. The line end attribute is used only for the internal ends of dotted or dashed lines.

Current position is set to $(XPOS0, YPOS0)$ (GBOX), or is unchanged (GCBOX).

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-C000 The HAXIS or VAXIS parameter is too large to fit the indicated corner into the size of the box.
Standard action: Corners with the largest axis that fit the box are drawn.

EC-C001 Either the HAXIS or VAXIS parameter is outside the range.
Standard action: A box with square corners is drawn.

Character String (GCHST, GCCHST) Orders

These orders draw a character string at a given position or at the current position.

Syntax

Character String at Given Position (GCHST) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'C3'	GCHST order code
1	UBIN	LENGTH	4–255	Length of following data
2–3	SBIN	XPOS	X'8000'–X'7FFF'	X _g coordinate of character string origin
4–5	SBIN	YPOS	X'8000'–X'7FFF'	Y _g coordinate of character string origin
6– <i>n</i>	CHAR	CP		Code points of each character in the string

Character String at Current Position (GCCHST) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'83'	GCCHST order code
1	UBIN	LENGTH	0–255	Length of following data
2– <i>n</i>	CHAR	CP		Code points of each character in the string

Semantics

The Character String at Given Position (GCHST) order draws a character string that starts at the specified position. The Character String at Current Position (GCCHST) order draws a character string that starts at the current position.

Note: The current position is changed to (XPOS,YPOS) (GCHST), or is unchanged (GCCHST).

The font from which the character definitions are to be obtained is given by the value of the current character set attribute. If the font identified by the value in the current character set attribute is not available, EC-C300 is raised. The standard action for EC-C300 is to use the standard default character set.

The particular character definitions identified by the current character set are determined by the code points in the Character String order. The length of the code points is determined by the font.

All code points in the Character String order must refer to valid graphic characters. If they do not, EC-C301 is raised. The standard action for EC-C301 is to use the standard default character symbol.

The color of the foreground of all characters in the string is given by the current value of the character color attribute.

The way in which characters in the string are merged with any output primitives that have already been drawn is controlled by the values of the character mix and background mix attributes.

The current values of the line type, line width, pattern set, and pattern symbol attributes have no effect on the appearance of the characters in the string.

A Character String at Given Position (GCHST) order with an initial position (XPOS,YPOS), but with no string of code points, is permitted. This serves only to move the current position. A Character String at Current Position (GCCHST) order with no string of code points is permitted and is treated as a No-Op.

The Set Character Precision, Set Character Set, Set Character Cell, Set Character Angle, and Set Character Direction drawing orders determine the character size, character rotation, and character direction. For more information, see [“Character Strings” on page 51](#).

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-C300 The font identified by the value in the current character set attribute is not available.

Standard action: The standard default character set is used. In AFP environments, this is the presentation device default font.

EC-C301 A code point in the order does not refer to a valid graphic character.

Standard action: The standard default character symbol is used.

EC-C302 The current character set attribute value identifies a character set definition that cannot support the functions implied by the current character precision attribute.

Standard action: The character set identified by the current character set attribute is used with the highest value of precision that the character set can support.

EC-C303 The character encoding is Unicode, but the code points in the drawing order are not valid Unicode data. This can be caused by one of the following:

- A high-order surrogate code value was not immediately followed by a low-order surrogate code value. The high-order surrogate range is U+D800 through U+DBFF.
- A low-order surrogate code value was not immediately preceded by a high-order surrogate code value. The low-order surrogate range is U+DC00 through U+DFFF.
- An illegal UTF-8 byte sequence, as defined in the Unicode Specification, was specified.

Standard action: The remainder of the code points are skipped.

Application Note: For a formal definition of Unicode and UTF-8, consult the Unicode Specification, which is available from the Unicode Consortium at www.unicode.org. The illegal UTF-8 byte sequences can be summarized as follows:

- The value in the 1st byte of the UTF-8 byte sequence was not in the legal UTF-8 range (X'00' - X'7F' and X'C2' - X'F4').
- The value in the 2nd byte of the UTF-8 byte sequence was not in the legal UTF-8 range allowed by the value in the 1st byte. The valid ranges for the 2nd byte are shown in [Table 14](#).

Table 14. Valid Values for UTF-8 First and Second Bytes

First Byte	Second Byte
X'C2' - X'DF'	X'80' - X'BF'
X'E0'	X'A0' - X'BF'
X'E1' - X'EC'	X'80' - X'BF'

Character String

Table 14 Valid Values for UTF-8 First and Second Bytes (cont'd.)

First Byte	Second Byte
X'ED'	X'80' - X'9F'
X'EE' - X'EF'	X'80' - X'BF'
X'F0'	X'90' - X'BF'
X'F1' - X'F3'	X'80' - X'BF'
X'F4'	X'80' - X'8F'

- The value in the 3rd or 4th byte of the UTF-8 byte sequence was not in the legal UTF-8 range for that byte (X'80' - X'BF').

Comment (GCOMT) Order

This order enables data to be stored within a segment.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'01'	GCOMT order code
1	UBIN	LENGTH	0–255	Length of following data
2– <i>n</i>	UNDF	DATA	Any value	Comment data

Semantics

This order is treated as a No-Op. It has no effect on the GPS or on any current attribute or control. The data within the order can be any value and is ignored. The order can appear anywhere within a segment.

Exception Conditions

This order does not raise any exception conditions.

Cubic Bezier Curve (GCBEZ, GCCBEZ) Orders

These orders generate a Cubic Bezier Curve that starts at a given position or at the current position.

Syntax

Cubic Bezier Curve at Given Position (GCBEZ) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'E5'	GCBEZ order code
1	UBIN	LENGTH	4- <i>n</i>	Length of following data; <i>n</i> must be less than 255 and be equal to $12m + 4$, where <i>m</i> is the number of sets of points
2-3	SBIN	XPOS0	X'8000'-X'7FFF'	X _g coordinate of first curve start point
4-5	SBIN	YPOS0	X'8000'-X'7FFF'	Y _g coordinate of first curve start point
6-7	SBIN	XPOS1	X'8000'-X'7FFF'	X _g coordinate of first curve, first control point
8-9	SBIN	YPOS1	X'8000'-X'7FFF'	Y _g coordinate of first curve, first control point
10-11	SBIN	XPOS2	X'8000'-X'7FFF'	X _g coordinate of first curve, second control point
12-13	SBIN	YPOS2	X'8000'-X'7FFF'	Y _g coordinate of first curve, second control point
14-15	SBIN	XPOS3	X'8000'-X'7FFF'	X _g coordinate of first curve endpoint, second curve start point
16-17	SBIN	YPOS3	X'8000'-X'7FFF'	Y _g coordinate of first curve endpoint, second curve start point
18-19	SBIN	XPOS4	X'8000'-X'7FFF'	X _g coordinate of second curve, first control point
20-21	SBIN	YPOS4	X'8000'-X'7FFF'	Y _g coordinate of second curve, first control point
22-23	SBIN	XPOS5	X'8000'-X'7FFF'	X _g coordinate of second curve, second control point
24-25	SBIN	YPOS5	X'8000'-X'7FFF'	Y _g coordinate of second curve, second control point
⋮		⋮		Coordinate data of further points
	SBIN	XPOSF-2	X'8000'-X'7FFF'	X _g coordinate of final curve, first control point
	SBIN	YPOSF-2	X'8000'-X'7FFF'	Y _g coordinate of final curve, first control point
	SBIN	XPOSF-1	X'8000'-X'7FFF'	X _g coordinate of final curve, second control point
	SBIN	YPOSF-1	X'8000'-X'7FFF'	Y _g coordinate of final curve, second control point
	SBIN	XPOSF	X'8000'-X'7FFF'	X _g coordinate of final curve endpoint
	SBIN	YPOSF	X'8000'-X'7FFF'	Y _g coordinate of final curve endpoint

Cubic Bezier Curve at Current Position (GCCBEZ) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'A5'	GCCBEZ order code
1	UBIN	LENGTH	0- <i>n</i>	Length of following data; <i>n</i> must be less than 255 and be equal to 12 <i>m</i> , where <i>m</i> is the number of sets of points
2-3	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of first curve, first control point
4-5	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of first curve, first control point
6-7	SBIN	XPOS2	X'8000'–X'7FFF'	X _g coordinate of first curve, second control point
8-9	SBIN	YPOS2	X'8000'–X'7FFF'	Y _g coordinate of first curve, second control point
10–11	SBIN	XPOS3	X'8000'–X'7FFF'	X _g coordinate of first curve endpoint, second curve start point
12-13	SBIN	YPOS3	X'8000'–X'7FFF'	Y _g coordinate of first curve endpoint, second curve start point
14-15	SBIN	XPOS4	X'8000'–X'7FFF'	X _g coordinate of second curve, first control point
16-17	SBIN	YPOS4	X'8000'–X'7FFF'	Y _g coordinate of second curve, first control point
18-19	SBIN	XPOS5	X'8000'–X'7FFF'	X _g coordinate of second curve, second control point
20-21	SBIN	YPOS5	X'8000'–X'7FFF'	Y _g coordinate of second curve, second control point
⋮		⋮		Coordinate data of further points
	SBIN	XPOSF-2	X'8000'–X'7FFF'	X _g coordinate of final curve, first control point
	SBIN	YPOSF-2	X'8000'–X'7FFF'	Y _g coordinate of final curve, first control point
	SBIN	XPOSF-1	X'8000'–X'7FFF'	X _g coordinate of final curve, second control point
	SBIN	YPOSF-1	X'8000'–X'7FFF'	Y _g coordinate of final curve, second control point
	SBIN	XPOSF	X'8000'–X'7FFF'	X _g coordinate of final curve endpoint
	SBIN	YPOSF	X'8000'–X'7FFF'	Y _g coordinate of final curve endpoint

Semantics

The Cubic Bezier Curve at Given Position (GCBEZ) order generates a curve that starts at P₀ and uses points P₁, P₂, and P₃. The Cubic Bezier Curve at Current Position (GCCBEZ) order generates a curve that starts at the current position and uses points P₁, P₂, and P₃.

Further points are used in groups of three to form a polycurve. Each group of points, together with the last point of the previous curve, generates a new curve, every curve being drawn independently for the set of four points.

See [“Cubic Bezier Curve” on page 27](#) for details of curve drawing.

The length of the order, LENGTH, must be consistent with the two-byte x-coordinates and two-byte y-coordinates and the requirement for sets of points, three at a time after the initial curve.

The current values of the line attributes are taken into account when drawing the curve.

Cubic Bezier Curve

A Cubic Bezier Curve at Given Position (GCBEZ) order with only one point is permitted. This serves only to move the current position, which is set to the specified point. A Cubic Bezier Curve at Current Position (GCCBEZ) order with only one point (the current position) is permitted and is treated as a No-Op.

Exception Conditions

The following exception condition raises a drawing process check:

- EC-0003** The order has an incorrect length.
The number of points, including the current position for the GCCBEZ drawing order, must equal $3m + 1$, where m is the number of curves. Each point requires a length of 4 bytes.

Delete Pattern (GDPT) Order

This order deletes a previously defined custom pattern or gradient, or deletes all previously defined custom patterns or gradients in a given pattern set.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'DF'	GDPT order code
1	UBIN	LENGTH	3, 4	Length of following data
2–3		RES	X'0000'	Reserved; only valid value
4	CODE	PATTSET	X'01' – X'FD'	Pattern set of the pattern(s) to be deleted
The following parameter is optional:				
5	CODE	PATTSYM	X'01' – X'FF'	Pattern symbol of the pattern to be deleted

Semantics

The Delete Pattern order, when it specifies a pattern symbol value, deletes one single custom pattern or gradient that was previously defined using the Begin Custom Pattern, Linear Gradient, or Radial Gradient orders. When the Delete Pattern order does not specify a pattern symbol value, it deletes all previously defined patterns in the specified pattern set.

The PATTSET value specifies the pattern set of the pattern(s) to be deleted. The PATTSYM value, if included, specifies the pattern symbol of the pattern to be deleted.

A request to delete all patterns in a given pattern set does not raise an exception condition if that pattern set has no patterns defined in it. However, a request to delete a specific pattern that has not been defined raises exception condition EC-DF00, for which the standard action is to ignore the Delete Pattern order.

Patterns in the default pattern set cannot be deleted. An attempt to do so will raise exception condition EC-DF01, for which the standard action is to ignore the Delete Pattern order.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-DF00 The PATTSET and PATTSYM parameters are within the valid range, but no pattern exists with the pattern set and pattern symbol specified.

Standard action: Ignore the Delete Pattern order.

EC-DF01 The value specified for the PATTSET parameter is invalid.

Standard action: Ignore the Delete Pattern order.

EC-DF02 The value specified for the PATTSYM parameter is invalid; pattern symbol X'00' cannot be deleted.

Delete Pattern

Standard action: Ignore the Delete Pattern order.

End Area (GEAR) Order

This order indicates the end of a set of primitives that define an area boundary.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'60'	GEAR order code
1	UBIN	LENGTH	0–255	Length of following data
2– <i>n</i>		DATA	X'00'...	Reserved; only valid value

Semantics

The End Area order identifies the end of an area.

The bytes of data on this order must all be X'00'. LENGTH is the number of bytes of zeros, and can be zero.

Exception Conditions

The following exception condition raises a drawing process check:

EC-6000 An End Area order has been executed without an unmatched Begin Area order having previously been executed.

End Custom Pattern (GECP) Order

This order indicates the end of a set of primitives that define a custom pattern.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'5E'	GECP order code
1		RES	X'00'	Reserved; only valid value

Semantics

The End Custom Pattern order identifies the end of the definition of a custom pattern.

Exception Conditions

The following exception condition causes a standard action to be taken:

EC-5E00 An End Custom Pattern order has been executed without an unmatched Begin Custom Pattern order having previously been executed.

Standard action: Ignore the End Custom Pattern order.

End Image (GEIMG) Order

This order identifies the end of an image definition.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'93'	GEIMG order code
1	UBIN	LENGTH	0–255	Length of following data
2– <i>n</i>		DATA	X'00'...	Reserved; only valid value

Semantics

The End Image order identifies the end of an image. The bytes of data on this order must all be X'00'. LENGTH is the number of bytes of zeros, and can be zero.

Exception Conditions

The following exception conditions raise a drawing process check:

- EC-9300** An End Image order is executed without an unmatched Begin Image order having been executed previously.
- EC-9301** The number of Image Data orders between the Begin Image and End Image orders is not equal to the number of rows in the image, as given by the value of HEIGHT in the Begin Image order.

End Prolog (GEPROL) Order

This order indicates the end of the prolog of a segment.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'3E'	GEPROL order code
1		RES	X'00'	Reserved; only valid value

Semantics

The End Prolog order indicates the end of the prolog section of a segment.

See [“Segment Prolog” on page 62](#) for details of the processing of segment prologs.

Exception Conditions

The following exception conditions raise a drawing process check:

- EC-000C** One of the following conditions has occurred within the prolog section of a segment:
- A supported order that is not valid within a prolog is specified.
 - The end of the segment has been reached without an End Prolog order.
- EC-3E00** An End Prolog order has occurred outside the prolog section of a segment.

Fillet (GFLT, GCFLT) Orders

These orders draw a curved line tangential to a specified set of straight lines, at the given position or at the current position.

Syntax

Fillet at Given Position (GFLT) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'C5'	GFLT order code
1	UBIN	LENGTH	4– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 4
2–3	SBIN	XPOS0	X'8000'–X'7FFF'	X _g coordinate of first line start point
4–5	SBIN	YPOS0	X'8000'–X'7FFF'	Y _g coordinate of first line start point
6–7	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of first line endpoint
8–9	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of first line endpoint
10–11	SBIN	XPOS2	X'8000'–X'7FFF'	X _g coordinate of second line endpoint
12–13	SBIN	YPOS2	X'8000'–X'7FFF'	Y _g coordinate of second line endpoint
⋮		⋮		Coordinate data of further line endpoints
	SBIN	XPOSF	X'8000'–X'7FFF'	X _g coordinate of final line endpoint
	SBIN	YPOSF	X'8000'–X'7FFF'	Y _g coordinate of final line endpoint

Fillet at Current Position (GCFLT) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'85'	GCFLT order code
1	UBIN	LENGTH	0– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 4
2–3	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of first line endpoint
4–5	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of first line endpoint
6–7	SBIN	XPOS2	X'8000'–X'7FFF'	X _g coordinate of second line endpoint
8–9	SBIN	YPOS2	X'8000'–X'7FFF'	Y _g coordinate of second line endpoint
⋮		⋮		Coordinate data of further line endpoints
	SBIN	XPOSF	X'8000'–X'7FFF'	X _g coordinate of final line endpoint
	SBIN	YPOSF	X'8000'–X'7FFF'	Y _g coordinate of final line endpoint

Semantics

The Fillet at Given Position (GFLT) order generates a single curve that starts at a specified position. The Fillet at Current Position (GCFLT) order generates a single curve that starts at the current position. Additional points can be added to form a polycurve.

Fillet

The points specified in the order are joined by imaginary straight lines and a curve is then fitted to the lines. The curve is tangential to the first line at its start point and to the last line at its end point. If there are intermediate lines, the curve is tangential to these lines at their center points. See ["Fillet" on page 26](#) for the definition of the curves drawn.

A Fillet at Given Position (GFLT) order with only an initial position is permitted. This serves only to move the current position. A Fillet at Current Position (GCFLT) order with only an initial position (the current position) is permitted and is treated as a No-Op.

When only two points are supplied, a straight line results.

The current values of the line attributes are taken into account when drawing the fillet, and the current position is set to the last point specified.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Full Arc (GFARC, GCFARC) Orders

These orders construct a full circle or an ellipse with the center at a specified point or at the current position.

Syntax

Full Arc at Given Position (GFARC) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'C7'	GFARC order code
1	UBIN	LENGTH	6	Length of following data
2–3	SBIN	XPOS	X'8000'–X'7FFF'	X _g coordinate of the center of the circle or ellipse
4–5	SBIN	YPOS	X'8000'–X'7FFF'	Y _g coordinate of the center of the circle or ellipse
6	UBIN	MH	X'00'–X'FF'	Integer Portion of Multiplier
7	UBIN	MFR	X'00'–X'FF'	Fractional Portion of Multiplier

Full Arc at Current Position (GCFARC) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'87'	GCFARC order code
1	UBIN	LENGTH	2	Length of following data
2	UBIN	MH	X'00'–X'FF'	Integer Portion of Multiplier
3	UBIN	MFR	X'00'–X'FF'	Fractional Portion of Multiplier

Semantics

The Full Arc at Given Position (GFARC) order constructs a circle or an ellipse with its center at the specified position. The Full Arc at Current Position (GCFARC) order constructs a circle or an ellipse with its center at the current position. A previous Set Arc Parameters drawing order determines the shape and orientation of the arc. If no Set Arc Parameters drawing order has been received, the presentation process draws an arc using the drawing default values of the arc parameters. The drawing direction is defined by the determinant of the transform, which is defined by the arc parameters. For details, see page [24](#).

Note: The current position is set to (XPOS,YPOS) (GFARC), or is unchanged (GCFARC).

The current values of the line attributes, except for line join, are taken into account when drawing the full arc. The line end attribute is used only for the internal ends of dotted or dashed lines.

If this drawing order is in an area definition, the arc is treated as a closed figure. The BOUNDARY parameter in the Begin Area order determines whether the boundary of the arc is drawn.

MH specifies the integer portion of a scale factor; MFR specifies the fractional portion of the scale factor. A combined value of X'0000' specifies a scale factor of 0. A decimal point is implied between MH and MFR. The fractional portion of the scale factor is calculated by dividing MFR by 256. For example, if MFR=X'40', its decimal value is 64, which, divided by 256 results in a fractional component for the scale factor of 1/4.

Full Arc

For a circle, the radius is $(MH \cdot R + MFR \cdot R)$ where R is the radius of the circle defined by the current arc parameters.

For an ellipse, the major and minor axes are $(MH \cdot MAJ + MFR \cdot MAJ)$ and $(MH \cdot MIN + MFR \cdot MIN)$, where MAJ and MIN are the major and minor axes of the ellipse defined by the current arc parameters.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception condition causes a standard action to be taken:

EC-C601 The drawing processor has detected an exceptional condition that can prevent the drawing of the arc within the normal limits of pel accuracy.

Standard action: The arc is drawn as accurately as the implementation can define. This action might produce straight lines.

Image Data (GIMD) Order

This order specifies the raster data for one scan line or row of an image.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'92'	GIMD order code
1	UBIN	LENGTH	0–255	Length of following data
2– <i>n</i>	BITS	DATA	Any value	Image Data

Semantics

The Image Data order contains the data for one scan line or row of an image. Each Image Data order can contain any number of bytes of data, from zero up to a maximum of 255 bytes.

The current position is not changed by the order.

If the LENGTH parameter is not equal to the rounded-up quotient of image WIDTH divided by 8, there are too few or too many data bytes, and exception EC-9201 exists.

See [“Begin Image \(GBIMG, GCBIMG\) Orders” on page 87](#) for details of the image construct.

Exception Conditions

The following exception conditions raise a drawing process check:

- EC-9200** A Begin Image order was not executed before the Image Data order in this segment.
- EC-9201** There are insufficient, or too many, bytes of data in the Image Data order.
- EC-9301** The number of Image Data orders between the Begin Image and End Image orders is not equal to the number of rows in the image, as specified by the HEIGHT parameter in the Begin Image order.

Line

Line (GLINE, GCLINE) Orders

These orders define one or more connected straight lines, drawn from the given position or from the current position.

Syntax

Line at Given Position (GLINE) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'C1'	GLINE order code
1	UBIN	LENGTH	4– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 4
2–3	SBIN	XPOS0	X'8000'–X'7FFF'	X _g coordinate of first line start point
4–5	SBIN	YPOS0	X'8000'–X'7FFF'	Y _g coordinate of first line start point
6–7	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of first line endpoint
8–9	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of first line endpoint
10–11	SBIN	XPOS2	X'8000'–X'7FFF'	X _g coordinate of second line endpoint
12–13	SBIN	YPOS2	X'8000'–X'7FFF'	Y _g coordinate of second line endpoint
⋮		⋮		Coordinate data of further line endpoints
	SBIN	XPOSF	X'8000'–X'7FFF'	X _g coordinate of final line endpoint
	SBIN	YPOSF	X'8000'–X'7FFF'	Y _g coordinate of final line endpoint

Line at Current Position (GCLINE) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'81'	GCLINE order code
1	UBIN	LENGTH	0– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 4
2–3	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of first line endpoint
4–5	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of first line endpoint
6–7	SBIN	XPOS2	X'8000'–X'7FFF'	X _g coordinate of second line endpoint
8–9	SBIN	YPOS2	X'8000'–X'7FFF'	Y _g coordinate of second line endpoint
⋮		⋮		Coordinate data of further line endpoints
	SBIN	XPOSF	X'8000'–X'7FFF'	X _g coordinate of final line endpoint
	SBIN	YPOSF	X'8000'–X'7FFF'	Y _g coordinate of final line endpoint

Semantics

The Line at Given Position (GLINE) order draws a line from the point specified by the first pair of coordinates to the point specified by the second pair of coordinates. If additional coordinate pairs are specified, the presentation process draws a line from the previous endpoint to the next coordinate pair. The Line at Current

Position (GCLINE) order draws a line from the current position to the point specified by the first coordinate pair. If additional coordinate pairs are specified, the presentation process draws a line from the previous endpoint to the next coordinate pair. Consecutive points in the orders are joined by straight lines.

The current values of the line attributes are taken into account when drawing the line.

The current position is set to the last point specified.

A Line at Given Position (GLINE) order with only an initial position is permitted. This form of GLINE moves the current position. A Line at Current Position (GCLINE) order with only an initial position (the current position) is permitted and is treated as a No-Op.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Linear Gradient (GLGD) Order

This order defines a linear gradient to be used to fill an area.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'FE'	Extended format order code
1	CODE		X'DC'	GLGD qualifier code
2–3	UBIN	LENGTH	29–65,535	Length of following data
4–5		RES	X'0000'	Reserved; only valid value
6	CODE	PATTSET	X'01' – X'FD'	Pattern set of the gradient
7	CODE	PATTSYM	X'01' – X'FF'	Pattern symbol of the gradient
8–9	SBIN	X_S	X'8000'–X'7FFF'	X _g coordinate of the start of the gradient line
10–11	SBIN	Y_S	X'8000'–X'7FFF'	Y _g coordinate of the start of the gradient line
12–13	SBIN	X_E	X'8000'–X'7FFF'	X _g coordinate of the end of the gradient line
14–15	SBIN	Y_E	X'8000'–X'7FFF'	Y _g coordinate of the end of the gradient line
16– <i>n</i>		COLSPEC_S	See Semantics	Color specification of the start color (13–15 bytes)
(<i>n</i> +1)– <i>m</i>		COLVALUE_E	See Semantics	Color value of the end color (2–4 bytes)
<i>m</i> +1	CODE	OUTSIDE_S	X'00'–X'03'	Value for how to fill areas outside the start side of the gradient: X'00' None X'01' Pad X'02' Repeat X'03' Reflect All other values Reserved
<i>m</i> +2	CODE	OUTSIDE_E	X'00'–X'03'	Value for how to fill areas outside the end side of the gradient: X'00' None X'01' Pad X'02' Repeat X'03' Reflect All other values Reserved
The following parameters are optional:				
	UBIN	OFFSET_1	X'0000'–X'2710' (0–10,000)	Offset along the gradient line of the first optional color stop (2 bytes)
		COLVALUE_1	See Semantics	Color value of the color of the first color stop (2–4 bytes)
	UBIN	OFFSET_2	X'0000'–X'2710' (0–10,000)	Offset along the gradient line of the second optional color stop (2 bytes)
		COLVALUE_2	See Semantics	Color value of the color of the second color stop (2–4 bytes)
⋮		⋮		Further color stop information

Offset	Type	Name	Range	Meaning
	UBIN	OFFSET_F	X'0000'–X'2710' (0–10,000)	Offset along the gradient line of the final optional color stop (2 bytes)
		COLVALUE_F	See Semantics	Color value of the color of the final color stop (2–4 bytes)

Semantics

The Linear Gradient order defines a linear gradient to be used later to fill an area. See [“Gradients” on page 43](#) for details of gradients.

The gradient goes from the start point (X_S,Y_S) to the end point (X_E,Y_E), with the color gradually changing from the start color (COLSPEC_S) to the end color (COLVALUE_E). Areas outside the gradient are filled based on the OUTSIDE_S and OUTSIDE_E parameters. Any number of color stops can be defined along the gradient line from the start point to the end point, which define offsets along the line where a specific color is to be found.

If the start point and end point are the same point, usage of the gradient will result in no fill, no matter the value of the OUTSIDE_S and OUTSIDE_E parameters and no matter how many color stops have been specified.

The offset fields in the color stops have values that can range from 0 to 10,000. This value is then divided by 10,000 to produce a number from 0.0 to 1.0, with 0.0 meaning the start point, 1.0 the end point, 0.5 the halfway point, and so on.

The color stops must be in increasing order of offset; that is, each color stop offset value must be greater than or equal to the previous color stop offset value. If a color stop has an offset value that is smaller than the offset value of any previous color stop, or is otherwise invalid, exception condition EC-DC05 is raised, for which the standard action is to ignore the color stop.

The color specification of the start color, COLSPEC_S, has the same format as bytes 1–end of the Set Process Color (GSPCOL) drawing order; see [“Set Process Color \(GSPCOL\) Order” on page 161](#) for information on how to process the color specification. Included in the color specification is a length field, a color space field, and four fields indicating how many bits are in each color component, as well as a color value field. The color value field specifies the start color and is interpreted using the other fields in the color specification. For all other colors in this order—that is, for the end color and for all color stop colors—only the color value field is specified. These color values are all the same length as the color value contained in COLSPEC_S, and are interpreted in the same way. As an example, if the start color is an RGB color encoded in three bytes, one for each component (R, G, and B), then all other colors in this order will also be three-byte values, one byte for each component.

For problems with the colors specified in this order, exception conditions EC-0E02, EC-0E03, EC-0E04, and EC-0E05 are reported as described in the Set Process Color order. Note, however, that the standard action for the EC-0E02, EC-0E03, and EC-0E05 exceptions is different for this order than for the Set Process Color drawing order. For all three exception conditions, the standard action is to ignore this Linear Gradient order.

If the length field in COLSPEC_S (the first byte) is invalid, exception condition EC-DC06 is raised, for which the standard action is to ignore the Linear Gradient order.

In addition, there are some rules about the colors specified in this order:

- The Standard OCA color space (X'40') cannot be used.
- If the Highlight color space (X'06') is used, all color values must resolve to Indexed CMR Color Palette tags.

If the color specifications do not follow these rules, exception condition EC-DC07 is raised, for which the standard action is to ignore the Linear Gradient order.

Linear Gradient

The smooth transition from one color to another requires interpolation calculations to be performed. For consistency between implementations:

- Linear interpolation is done.
- Interpolation is done in the specified color space. If the specified color space is the Highlight color space:
 - If all colors resolve to Color Palette tags of the same type, interpolation is done in the color space corresponding to that type; for example, if all colors resolve to Color Palette CMYK tags, interpolation is done in the CMYK color space.
 - A special case of the above is if all colors resolve to Color Palette Named Colorants tags. In this case, if all named colorants required for all the colors in the gradient are available in the device, interpolation is done in the intensity of the named colorants; otherwise, interpolation is done in the CIELAB color space, using the CIELABValue field of the Color Palette Named Colorants tags.
 - If all colors do not resolve to Color Palette tags of the same type, interpolation is done in the CIELAB color space, using the CIELABValue field of the Color Palette tags.

If the LENGTH field of this drawing order is not a valid length, given the expected color value sizes, exception EC-0003 is raised. The valid values for the LENGTH field, where n is the number of color stops, are as follows:

First byte of COLSPEC_S	Valid values of the LENGTH field
12	$29 + (n * 4)$
13	$31 + (n * 5)$
14	$33 + (n * 6)$

The PATTSET and PATTSYM values specify the pattern set and pattern symbol where this gradient will reside. When the current values of the pattern set and pattern symbol attributes specify these PATTSET and PATTSYM values, respectively, this gradient will be used to do area fill.

Linear gradients (defined with this order), radial gradients (defined with the Radial Gradient order), and custom patterns (defined with the Begin Custom Pattern order) share the pattern sets X'01'–X'FD'; the patterns using these pattern sets can be any combination of gradients and custom patterns without restriction.

Definition of a gradient using this drawing order does not set the pattern set and pattern symbol attributes to use the gradient. To use the gradient, the pattern set and pattern symbol attributes must be explicitly set.

All possible valid values of the PATTSET and PATTSYM fields must be supported. There are $253 \cdot 255 = 64,515$ possible combinations of PATTSET and PATTSYM. However, it is not required that implementations support the creation of 64,515 gradients. If insufficient storage is available to correctly handle a linear gradient definition, exception condition EC-DC03 is raised, for which the standard action is to ignore the Linear Gradient order.

It is not possible to replace a gradient or custom pattern simply by sending a Linear Gradient order with the same PATTSET and PATTSYM parameters. If this is attempted, exception EC-DC02 is raised, for which the standard action is to ignore the Linear Gradient order. To replace a pattern at a given pattern set and pattern symbol, first delete the existing pattern using the Delete Pattern drawing order.

The current position is not used nor changed by this drawing order.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-0E02 The color space specified in the order is invalid or unsupported.

Standard action: Ignore the Linear Gradient order.

EC-0E03 A color value specified in the order is invalid or unsupported.

Standard action: Ignore the Linear Gradient order.

EC-0E04 A highlight color percent value specified in the order is invalid.

Standard action: The maximum valid percent value is used.

EC-0E05 The number of bits for a color component specified in the order is invalid or unsupported.

Standard action: Ignore the Linear Gradient order.

EC-DC00 The value specified for the PATTSET parameter is invalid.

Standard action: Ignore the Linear Gradient order.

EC-DC01 The value specified for the PATTSYM parameter is invalid; a gradient cannot be assigned to pattern symbol X'00'.

Standard action: Ignore the Linear Gradient order.

EC-DC02 The PATTSET and PATTSYM values are within the valid range, but a pattern already resides at that location.

Standard action: Ignore the Linear Gradient order.

EC-DC03 Insufficient storage available to process and store a linear gradient.

Standard action: Ignore the Linear Gradient order.

EC-DC04 A value for one or both of the OUTSIDE_S or OUTSIDE_E parameters is invalid.

Standard action: Use the value X'00' – None.

EC-DC05 The value specified for a color stop offset is invalid or is less than a previous color stop offset.

Standard action: The color stop is ignored.

EC-DC06 Invalid length value in the color specification.

Standard action: Ignore the Linear Gradient order.

EC-DC07 Color space or color values in the order are valid, but do not follow the rules for colors in a gradient.

Standard action: Ignore the Linear Gradient order.

Marker (GMRK, GCMRK) Orders

These orders draw the current marker symbol at one or more positions starting from the given position or from the current position.

Syntax

Marker at Given Position (GMRK) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'C2'	GMRK order code
1	UBIN	LENGTH	4– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 4
2–3	SBIN	XPOS0	X'8000'–X'7FFF'	X _g coordinate of first marker
4–5	SBIN	YPOS0	X'8000'–X'7FFF'	Y _g coordinate of first marker
6–7	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of second marker
8–9	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of second marker
⋮		⋮		Coordinate data of further markers
	SBIN	XPOSF	X'8000'–X'7FFF'	X _g coordinate of final marker
	SBIN	YPOSF	X'8000'–X'7FFF'	Y _g coordinate of final marker

Marker at Current Position (GCMRK) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'82'	GCMRK order code
1	UBIN	LENGTH	0– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 4
2–3	SBIN	XPOS1	X'8000'–X'7FFF'	X _g coordinate of second marker
4–5	SBIN	YPOS1	X'8000'–X'7FFF'	Y _g coordinate of second marker
⋮		⋮		Coordinate data of further markers
	SBIN	XPOSF	X'8000'–X'7FFF'	X _g coordinate of final marker
	SBIN	YPOSF	X'8000'–X'7FFF'	Y _g coordinate of final marker

Semantics

The Marker at Given Position (GMRK) order draws an initial marker symbol at the point specified by the first coordinate pair, and draws additional marker symbols at all the points specified by the remaining coordinate pairs. The Marker at Current Position (GCMRK) order draws an initial marker symbol at the current position and draws additional marker symbols at all the points specified by the remaining coordinate pairs.

Markers are positioned in GPS. The specified points define the position of the center of the marker. The current position is set to the last coordinate specified. If no coordinate has been specified, the current position remains unchanged.

A Marker at Current Position (GCMRK) order with no coordinate values specified—that is, the value of LENGTH is zero—draws a marker at the current position.

The markers are drawn at a size determined by the marker cell-size attribute.

The marker set from which the marker symbol is obtained is given by the value of the marker set attribute. If this marker set is not available, EC-C200 is raised, the standard action for which is to use the standard default marker set. In AFP environments, this is the default marker set.

The particular marker symbol that is drawn is given by the value of the current marker symbol attribute. If the code point is undefined in the marker set identified by the current marker set attribute, EC-C201 is raised, the standard action for which is to use the standard default marker symbol. In AFP environments, this is X'01'—Cross.

The color of the markers is given by the value of the current marker color. The way in which the markers are merged with any output primitives that have already been drawn is controlled by the values of the marker mix and marker background mix attributes.

Note: It is not an error if a marker symbol is placed inside the GPS such that part of the marker lies outside the GPS. However, the appearance of such a marker in the GPS is implementation defined.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-C200 The marker set identified by the value in the current marker set attribute is not available.

Standard action: The standard default marker set is used. In AFP environments, this is the default marker set.

EC-C201 The code point in the current marker symbol attribute is not defined in the current marker set.

Standard action: The standard default marker symbol is used. In AFP environments, this is X'01'—Cross.

No-Operation

No-Operation (GNOP1) Order

This order is a No-Operation.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'00'	GNOP1 order code

Semantics

The No-Operation order is a null operation. It has no effect on the GPS, or any current attribute or control.

Exception Conditions

This order does not raise any exception conditions.

Partial Arc (GPARC, GCPARC) Orders

These orders draw a line from the given position or the current position to the start of an arc, and then construct a partial arc. The start point of the arc is specified by the start angle, and the length of the arc is specified by the sweep angle.

Syntax

Partial Arc at Given Position (GPARC) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'E3'	GPARC order code
1	UBIN	LENGTH	18	Length of following data
2–3	SBIN	XPOS	X'8000'–X'7FFF'	X_g coordinate of line start point
4–5	SBIN	YPOS	X'8000'–X'7FFF'	Y_g coordinate of line start point
6–7	SBIN	XCENT	X'8000'–X'7FFF'	X_g coordinate of the center of the arc
8–9	SBIN	YCENT	X'8000'–X'7FFF'	Y_g coordinate of the center of the arc
10	UBIN	MH	X'00'–X'FF'	Integer Portion of Multiplier
11	UBIN	MFR	X'00'–X'FF'	Fractional Portion of Multiplier
12–15	SBIN	START	X'00000000'–X'7FFFFFFF'	Start angle of arc, modulo 360
16–19	SBIN	SWEEP	X'00000000'–X'7FFFFFFF'	Sweep angle of arc, modulo 360

Partial Arc at Current Position (GCPARC) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'A3'	GCPARC order code
1	UBIN	LENGTH	14	Length of following data
2–3	SBIN	XCENT	X'8000'–X'7FFF'	X_g coordinate of the center of the arc
4–5	SBIN	YCENT	X'8000'–X'7FFF'	Y_g coordinate of the center of the arc
6	UBIN	MH	X'00'–X'FF'	Integer Portion of Multiplier
7	UBIN	MFR	X'00'–X'FF'	Fractional Portion of Multiplier
8–11	SBIN	START	X'00000000'–X'7FFFFFFF'	Start angle of arc, modulo 360
12–15	SBIN	SWEEP	X'00000000'–X'7FFFFFFF'	Sweep angle of arc, modulo 360

Semantics

The Partial Arc at Given Position (GPARC) order draws a line from point (XPOS,YPOS) to the start of an arc, then draws the arc with its center at point (XCENT,YCENT). The Partial Arc at Current Position (GCPARC) order draws a line from the current position to the start of an arc, then draws the arc with its center at point (XCENT,YCENT). The arc is part of the full arc defined by the current arc parameters and the multiplier

Partial Arc

specified by MH and MFR. The part of the arc that is drawn is defined by the starting angle, START, and the sweep angle, SWEEP. Both angles are defined on the unit circle space and are transformed by an amount defined by the current arc parameters in the same way that the unit circle is transformed. See [“Partial Arc” on page 25](#) for details.

A previous Set Arc Parameters drawing order determines the shape and orientation of the arc. If no Set Arc Parameters drawing order has been received, the presentation process draws an arc using the drawing default values of the arc parameters.

The drawing direction is defined by the determinant of the transform, which is defined by the arc parameters. For details, see [page 24](#).

MH specifies the integer portion of a scale factor; MFR specifies the fractional portion of the scale factor. A decimal point is implied between MH and MFR. The fractional portion of the scale factor is calculated by dividing MFR by 256. For example, if MFR=X'40', its decimal value is 64, which, divided by 256 results in a fractional component for the scale factor of 1/4.

For a circle, the radius is $(MH \cdot R + MFR \cdot R)$ where R is the radius of the circle defined by the current arc parameters.

For an ellipse, the major and minor axes are $(MH \cdot MAJ + MFR \cdot MAJ)$ and $(MH \cdot MIN + MFR \cdot MIN)$, where MAJ and MIN are the major and minor axes of the ellipse defined by the current arc parameters.

The START and SWEEP parameters are defined as signed 32-bit integers, whose range is restricted to positive values, that is, X'00000000' to X'7FFFFFFF'. The START and SWEEP angles are the numbers, in degrees, that result from dividing the integers by 65,536 (2^{16}) and interpreting the result as a modulo 360 number. The effective range of the angles is therefore greater than or equal to 0° and less than 360° . For example, if the sweep angle is specified to be X'00007FFF', its value is $32,767 \div 65,536 \text{ modulo } 360 = .5^\circ$.

Note that since a sweep angle of any integer multiple of 360° results in a 0° arc, this drawing order cannot be used to draw a complete arc. The Full Arc drawing order can be used to draw a complete arc.

The current values of the line attributes are taken into account when drawing the partial arc.

The current position is moved to the endpoint of the arc.

Exception Conditions

The following exception conditions raise a drawing process check:

- | | |
|----------------|--|
| EC-0003 | The order has incorrect length. |
| EC-E300 | The partial arc started inside GPS but then finished outside. Therefore, the calculated new current position is outside GPS. |
| EC-E302 | A negative value is specified for the SWEEP angle. |
| EC-E303 | A negative value is specified for the START angle. |

The following exception conditions cause a standard action to be taken:

- EC-000D** The start and end points of a partial arc are inside GPS, but a portion of the arc is outside GPS.
Standard action: All drawing outside the GPS is suppressed. The portion of the arc that is inside the GPS is drawn.
- EC-C601** The drawing processor has detected an exceptional condition that can prevent the drawing of the arc within the normal limits of pel accuracy.
Standard action: The arc is drawn as accurately as the implementation can define. This action might produce straight lines.

Radial Gradient (GRGD) Order

This order defines a radial gradient to be used to fill an area.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'FE'	Extended format order code
1	CODE		X'DD'	GRGD qualifier code
2–3	UBIN	LENGTH	33–65,535	Length of following data
4–5		RES	X'0000'	Reserved; only valid value
6	CODE	PATTSET	X'01' – X'FD'	Pattern set of the gradient
7	CODE	PATTSYM	X'01' – X'FF'	Pattern symbol of the gradient
8–9	SBIN	XPOS_S	X'8000'–X'7FFF'	X _g coordinate of the center of the start full arc
10–11	SBIN	YPOS_S	X'8000'–X'7FFF'	Y _g coordinate of the center of the start full arc
12	UBIN	MH_S	X'00'–X'FF'	Integer portion of the multiplier for the start full arc
13	UBIN	MFR_S	X'00'–X'FF'	Fractional portion of the multiplier for the start full arc
14–15	SBIN	XPOS_E	X'8000'–X'7FFF'	X _g coordinate of the center of the end full arc
16–17	SBIN	YPOS_E	X'8000'–X'7FFF'	Y _g coordinate of the center of the end full arc
18	UBIN	MH_E	X'00'–X'FF'	Integer portion of the multiplier for the end full arc
19	UBIN	MFR_E	X'00'–X'FF'	Fractional portion of the multiplier for end full arc
20– <i>n</i>		COLSPEC_S	See Semantics	Color specification of the start color (13–15 bytes)
(<i>n</i> +1)– <i>m</i>		COLVALUE_E	See Semantics	Color value of the end color (2–4 bytes)
<i>m</i> +1	CODE	OUTSIDE_S	X'00'–X'03'	Value for how to fill areas outside the start side of the gradient: X'00' None X'01' Pad X'02' Repeat X'03' Reflect All other values Reserved
<i>m</i> +2	CODE	OUTSIDE_E	X'00'–X'03'	Value for how to fill areas outside the end side of the gradient: X'00' None X'01' Pad X'02' Repeat X'03' Reflect All other values Reserved
The following parameters are optional:				
	UBIN	OFFSET_1	X'0000'–X'2710' (0–10,000)	Offset of the intermediate full arc of the first optional color stop (2 bytes)

Offset	Type	Name	Range	Meaning
		COLVALUE_1	See Semantics	Color value of the color of the first color stop (2–4 bytes)
	UBIN	OFFSET_2	X'0000'–X'2710' (0–10,000)	Offset of the intermediate full arc of the second optional color stop (2 bytes)
		COLVALUE_2	See Semantics	Color value of the color of the second color stop (2–4 bytes)
⋮		⋮		Further color stop information
	UBIN	OFFSET_F	X'0000'–X'2710' (0–10,000)	Offset of the intermediate full arc of the final optional color stop (2 bytes)
		COLVALUE_F	See Semantics	Color value of the color of the final color stop (2–4 bytes)

Semantics

The Radial Gradient order defines a radial gradient to be used later to fill an area. See [“Gradients” on page 43](#) for details of gradients.

The gradient goes from the start full arc to the end full arc. The color changes gradually from the start color (COLSPEC_S) to the end color (COLVALUE_E). Areas outside the gradient are filled based on the OUTSIDE_S and OUTSIDE_E parameters. Any number of color stops can be defined along the gradient lines from the start full arc to the end full arc, which define intermediate full arcs between the start and end full arcs where a specific color is to be found.

The start and end full arcs are defined in the same way as in the Full Arc at Given Position (GFARC) drawing order, using the appropriate XPOS, YPOS, MH, and MFR values in this drawing order. See [“Full Arc \(GFARC, GCFARC\) Orders” on page 107](#) for more information. Note that since both full arcs use the same arc parameters, the two will have the same shape (as will all intermediate full arcs along the gradient).

Either multiplier value can be zero (that is, either MH_S=MFR_S=0 or MH_E=MFR_E=0), in which case the gradient starts or ends at a point instead of a full arc. If both multiplier values are zero (MH_S=MFR_S=MH_E=MFR_E=0), however, usage of the gradient will result in no fill, no matter the value of the OUTSIDE_S and OUTSIDE_E parameters and no matter how many color stops have been specified. In addition, if the start and end full arc have the same center and multiplier, usage of the gradient will result in no fill.

If part or all of either full arc is outside the GPS, this is not an error. This functionality can be used to get radial gradients that completely fill the GPS. Implementations that can maintain a position outside the GPS should produce a gradient as expected—gradually changing from the start color at the start full arc toward the end color at the end full arc, even though some parts of the intermediate full arcs might be outside the GPS. For implementations that cannot maintain a position outside GPS, the results are implementation dependent.

The offset fields in the color stops have values that can range from 0 to 10,000. This value is then divided by 10,000 to produce a number from 0.0 to 1.0, with 0.0 meaning the start full arc, 1.0 the end full arc, 0.5 the intermediate full arc halfway between the two, and so on.

The color stops must be in increasing order of offset; that is, each color stop offset value must be greater than or equal to the previous color stop offset value. If a color stop has an offset value that is smaller than the offset value of any previous color stop, or is otherwise invalid, exception condition EC-DD05 is raised, for which the standard action is to ignore the color stop.

The color specification of the start color, COLSPEC_S, has the same format as bytes 1–end of the Set Process Color (GSPCOL) drawing order; see [“Set Process Color \(GSPCOL\) Order” on page 161](#) for information on how

Radial Gradient

to process the color specification. Included in the color specification is a length field, a color space field, and four fields indicating how many bits are in each color component, as well as a color value field. The color value field specifies the start color and is interpreted using the other fields in the color specification. For all other colors in this order—that is, for the end color and for all color stop colors—only the color value field is specified. These color values are all the same length as the color value contained in COLSPEC_S, and are interpreted in the same way. As an example, if the start color is an RGB color encoded in three bytes, one for each component (R, G, and B), then all other colors in this order will also be three-byte values, one byte for each component.

For problems with the colors specified in this order, exception conditions EC-0E02, EC-0E03, EC-0E04, and EC-0E05 are reported as described in the Set Process Color order. Note, however, that the standard action for the EC-0E02, EC-0E03, and EC-0E05 exceptions is different for this order than for the Set Process Color drawing order. For all three exception conditions, the standard action is to ignore this Radial Gradient order.

If the length field in COLSPEC_S (the first byte) is invalid, exception condition EC-DD06 is raised, for which the standard action is to ignore the Radial Gradient order.

In addition, there are some rules about the colors specified in this order:

- The Standard OCA color space (X'40') cannot be used.
- If the Highlight color space (X'06') is used, all color values must resolve to Indexed CMR Color Palette tags.

If the color specifications do not follow these rules, exception condition EC-DD07 is raised, for which the standard action is to ignore the Radial Gradient order.

The smooth transition from one color to another requires interpolation calculations to be performed. For consistency between implementations:

- Linear interpolation is done.
- Interpolation is done in the specified color space. If the specified color space is the Highlight color space:
 - If all colors resolve to Color Palette tags of the same type, interpolation is done in the color space corresponding to that type; for example, if all colors resolve to Color Palette CMYK tags, interpolation is done in the CMYK color space.
 - A special case of the above is if all colors resolve to Color Palette Named Colorants tags. In this case, if all named colorants required for all the colors in the gradient are available in the device, interpolation is done in the intensity of the named colorants; otherwise, interpolation is done in the CIELAB color space, using the CIELABValue field of the Color Palette Named Colorants tags.
 - If all colors do not resolve to Color Palette tags of the same type, interpolation is done in the CIELAB color space, using the CIELABValue field of the Color Palette tags.

If the LENGTH field of this drawing order is not a valid length, given the expected color value sizes, exception EC-0003 is raised. The valid values for the LENGTH field, where n is the number of color stops, are as follows:

First byte of COLSPEC_S	Valid values of the LENGTH field
12	$33+ (n * 4)$
13	$35+ (n * 5)$
14	$37+ (n * 6)$

The PATTSET and PATTSYM values specify the pattern set and pattern symbol where this gradient will reside. When the current values of the pattern set and pattern symbol attributes specify these PATTSET and PATTSYM values, respectively, this gradient will be used to do area fill.

Linear gradients (defined with the Linear Gradient order), radial gradients (defined with this order), and custom patterns (defined with the Begin Custom Pattern order) share the pattern sets X'01'–X'FD'; the patterns using these pattern sets can be any combination of gradients and custom patterns without restriction.

Definition of a gradient using this drawing order does not set the pattern set and pattern symbol attributes to use the gradient. To use the gradient, the pattern set and pattern symbol attributes must be explicitly set.

All possible valid values of the PATTSET and PATTSYM fields must be supported. There are $253 \cdot 255 = 64,515$ possible combinations of PATTSET and PATTSYM. However, it is not required that implementations support the creation of 64,515 gradients. If insufficient storage is available to correctly handle a radial gradient definition, exception condition EC-DD03 is raised, for which the standard action is to ignore the Radial Gradient order.

It is not possible to replace a gradient or custom pattern simply by sending a Radial Gradient order with the same PATTSET and PATTSYM parameters. If this is attempted, exception EC-DD02 is raised, for which the standard action is to ignore the Radial Gradient order. To replace a pattern at a given pattern set and pattern symbol, first delete the existing pattern using the Delete Pattern drawing order.

The current position is not used nor changed by this drawing order.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-0E02 The color space specified in the order is invalid or unsupported.

Standard action: Ignore the Radial Gradient order.

EC-0E03 A color value specified in the order is invalid or unsupported.

Standard action: Ignore the Radial Gradient order.

EC-0E04 A highlight color percent value specified in the order is invalid.

Standard action: The maximum valid percent value is used.

EC-0E05 The number of bits for a color component specified in the order is invalid or unsupported.

Standard action: Ignore the Radial Gradient order.

EC-C601 The drawing processor has detected an exceptional condition that can prevent the drawing of one of the full arcs within the normal limits of pel accuracy.

Standard action: The arc is drawn as accurately as the implementation can define. This action might produce straight lines.

EC-DD00 The value specified for the PATTSET parameter is invalid.

Standard action: Ignore the Radial Gradient order.

EC-DD01 The value specified for the PATTSYM parameter is invalid; a gradient cannot be assigned to pattern symbol X'00'.

Standard action: Ignore the Radial Gradient order.

EC-DD02 The PATTSET and PATTSYM values are within the valid range, but a pattern already resides at that location.

Standard action: Ignore the Radial Gradient order.

EC-DD03 Insufficient storage available to process and store a radial gradient.

Standard action: Ignore the Radial Gradient order.

EC-DD04 A value for one or both of the OUTSIDE_S or OUTSIDE_E parameters is invalid.

Standard action: Use the value X'00' – None.

EC-DD05 The value specified for a color stop offset is invalid or is less than a previous color stop offset.

Standard action: The color stop is ignored.

EC-DD06 Invalid length value in the color specification.

Standard action: Ignore the Radial Gradient order.

EC-DD07 Color space or color values in the order are valid, but do not follow the rules for colors in a gradient.

Standard action: Ignore the Radial Gradient order.

Relative Line (GRLINE, GCRLINE) Orders

These orders define one or more connected straight lines, at the given position or at the current position. For these drawing orders, the endpoint of each line is specified as an offset from the previous endpoint rather than as an absolute value.

Syntax

Relative Line at Given Position (GRLINE) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'E1'	GRLINE order code
1	UBIN	LENGTH	4– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 2
2–3	SBIN	XPOS0	X'8000'–X'7FFF'	X _g coordinate of line start point
4–5	SBIN	YPOS0	X'8000'–X'7FFF'	Y _g coordinate of line start point
6	SBIN	XOFFS0	X'80'–X'7F'	Offset in X _g direction for first endpoint
7	SBIN	YOFFS0	X'80'–X'7F'	Offset in Y _g direction for first endpoint
8	SBIN	XOFFS1	X'80'–X'7F'	Offset in X _g direction for second endpoint
9	SBIN	YOFFS1	X'80'–X'7F'	Offset in Y _g direction for second endpoint
⋮		⋮		Offset data for further points
	SBIN	XOFFSF	X'80'–X'7F'	Offset in X _g direction for final endpoint
	SBIN	YOFFSF	X'80'–X'7F'	Offset in Y _g direction for final endpoint

Relative Line at Current Position (GCRLINE) Order

Offset	Type	Name	Range	Meaning
0	CODE		X'A1'	GCRLINE order code
1	UBIN	LENGTH	0– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 2
2	SBIN	XOFFS0	X'80'–X'7F'	Offset in X _g direction for first endpoint
3	SBIN	YOFFS0	X'80'–X'7F'	Offset in Y _g direction for first endpoint
4	SBIN	XOFFS1	X'80'–X'7F'	Offset in X _g direction for second endpoint
5	SBIN	YOFFS1	X'80'–X'7F'	Offset in Y _g direction for second endpoint
⋮		⋮		Offset data for further points
	SBIN	XOFFSF	X'80'–X'7F'	Offset in X _g direction for final endpoint
	SBIN	YOFFSF	X'80'–X'7F'	Offset in Y _g direction for final endpoint

Semantics

The Relative Line at Given Position (GRLINE) order adds the offset of each line endpoint cumulatively to the line start point (specified by XPOS0,YPOS0) to generate a sequence of points P₁, P₂, ... P_f, where P_f is the

Relative Line

final endpoint specified. The Relative Line at Current Position (GCRLINE) order adds the offset of each line endpoint cumulatively to the current position to generate a sequence of points P_1, P_2, \dots, P_f , where P_f is the final endpoint specified.

Straight lines are drawn connecting the points in sequence, that is, from the starting point to P_1 , from P_1 to P_2 , ... , and from $P_{(f-1)}$ to P_f .

Any number of offsets can be included within the limits of the length specifications.

A Relative Line at Given Position (GRLINE) order with only an initial position is permitted. This serves only to move the current position, which is set to the specified point. A Relative Line at Current Position (GCRLINE) order with only an initial position (the current position) is permitted and is treated as a No-Op.

A relative line that starts inside GPS, but has values of offset specified that accumulate to be outside GPS, is an error.

The current values of the line attributes are taken into account when drawing the relative lines.

Exception Conditions

The following exception conditions raise a drawing process check:

- EC-0003** The order has an incorrect length.
- EC-E100** A relative line starts inside GPS, but goes outside.

Segment Characteristics (GSGCH) Order

This order is processed as a No-Op in AFP GOCA. It is valid only in the prolog of a segment.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'04'	GSGCH order code
1	UBIN	LENGTH	0–255	Length of following data
2	CODE	CHID	X'00'	Identification code for characteristics
3– <i>n</i>	UNDF	PARMS		Parameters of characteristics

Semantics

Not applicable in AFP GOCA.

Exception Conditions

The following exception condition may optionally raise a drawing process check:

EC-0400 The Segment Characteristics order is detected outside the prolog section of a segment.

Set Arc Parameters (GSAP) Order

This order sets the values of the current arc parameters.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'22'	GSAP order code
1	UBIN	LENGTH	8	Length of following data
2-3	SBIN	P	X'8000'-X'7FFF'	P parameter of arc transform
4-5	SBIN	Q	X'8000'-X'7FFF'	Q parameter of arc transform
6-7	SBIN	R	X'8000'-X'7FFF'	R parameter of arc transform
8-9	SBIN	S	X'8000'-X'7FFF'	S parameter of arc transform

Semantics

The Set Arc Parameters order specifies the shape and orientation of a circle or an ellipse. Subsequent Full Arc orders specify the size and location of the circle or ellipse. Subsequent Partial Arc orders specify the size and location of the circle or ellipse that the partial arc is part of. For details, see [“Full Arc” on page 24](#) and [“Partial Arc” on page 25](#).

The parameters P, Q, R, and S define a transformation that maps the unit circle at the GPS origin ($X_g=0, Y_g=0$) to the required circle or ellipse, also at the GPS origin, such that:

$$\begin{aligned} X' &= P \cdot X + R \cdot Y \\ Y' &= S \cdot X + Q \cdot Y \end{aligned}$$

where X and Y are the coordinates of points on the unit circle, and X' and Y' are the coordinates of points on the arc. The drawing direction is defined by the determinant of the transform, which is defined by the arc parameters. For details, see page [24](#).

For a circle of radius r the parameters are:

$$\begin{aligned} P &= r \\ Q &= r \\ R &= 0 \\ S &= 0 \end{aligned}$$

For an ellipse with major axis $2a$ and minor axis $2b$ the parameters are:

$$\begin{aligned} P &= a \\ Q &= b \\ R &= 0 \\ S &= 0 \end{aligned}$$

For the same ellipse, but rotated A degrees counterclockwise with respect to the X_g axis the parameters are:

$$\begin{aligned} P &= a \cdot \cos(A) \\ Q &= b \cdot \cos(A) \\ R &= -b \cdot \sin(A) \\ S &= a \cdot \sin(A) \end{aligned}$$

This drawing order does not change the current position.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Set Background Mix (GSBMX) Order

This order provides a shorthand way of setting the following background mix attributes to the same value:

- Character background mix
- Image background mix
- Marker background mix
- Pattern background mix

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'0D'	GSBMX order code
1	CODE	MODE	X'00'–X'05'	Mix-mode value: X'00' Drawing default X'01'–X'04' Not supported in AFP GOCA X'05' Leave Alone All other values Reserved

Semantics

The Set Background Mix order sets the current value of all four background mix attributes to the value specified in the order.

The standard default in AFP environments is X'05'—Leave Alone.

Background mix attributes control the way in which the color of the background of a primitive is combined with the color of the GPS.

With MODE set to X'05', the background pels are transparent and do not affect the color of underlying pels in the GPS. Since this is the only background mix mode supported in AFP GOCA, selecting the drawing default (MODE X'00') will also default to MODE X'05'.

For a description of the meaning of the various mix modes, see [“Mix” on page 16](#).

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'05'—Leave Alone.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'05'—Leave Alone.

Set Character Angle (GSCA) Order

This order sets the value of the current character angle attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'34'	GSCA order code
1	UBIN	LENGTH	4	Length of following data
2–3	SBIN	XPOS	X'8000'–X'7FFF'	X _g coordinate of point
4–5	SBIN	YPOS	X'8000'–X'7FFF'	Y _g coordinate of point

Semantics

The Set Character Angle order sets the value of the current character angle attribute to the value specified in the order.

The character angle attribute controls the angle of the character baseline with respect to the GPS X_g axis for subsequent character strings. This angle is specified using the values XPOS and YPOS, where the character baseline is a line parallel to the line that passes through the points (X_g=0, Y_g=0) and (X_g=XPOS, Y_g=YPOS). The angle of the baseline relative to the X_g-axis of GPS is then the angle whose tangent is YPOS/XPOS. The values of YPOS and XPOS are not required to be the sine and cosine of the angle.

- If YPOS is zero, and XPOS is positive, the character angle is 0°.
- If XPOS is zero, and YPOS is positive, the character angle is 90°.
- If YPOS is zero, and XPOS is negative, the character angle is 180°.
- If XPOS is zero, and YPOS is negative, the character angle is 270°.

In AFP GOCA, the only supported values for character angle are 0°, 90°, 180°, and 270°. If XPOS is zero and YPOS is zero, the character angle is set to the drawing default. The standard default in AFP environments is 0°.

The application of this attribute is dependent on the value of the character precision attribute; see [“Character Strings” on page 51](#) for details. This drawing order does not change the attributes of any other drawing order.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception condition causes a standard action to be taken:

EC-3400 The specific character angle requested is not supported.

Standard action: The closest character angle supported is used.

Set Character Cell (GSCC) Order

This order sets the value of the current character cell-size attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'33'	GSCC order code
1	UBIN	LENGTH	4, 8	Length of following data
2–3	SBIN	CELLWI	X'8000'–X'7FFF'	Width of character cell, integer part
4–5	SBIN	CELLHI	X'8000'–X'7FFF'	Height of character cell, integer part
The following parameters are optional:				
6–7	UBIN	CELLWFR	X'0000'–X'FFFF'	Width of character cell, fractional part
8–9	UBIN	CELLHFR	X'0000'–X'FFFF'	Height of character cell, fractional part

Semantics

The Set Character Cell order sets the value of the current character cell-size attribute to the value specified in the order. Depending on the device implementation of the specified precision, this attribute is used to scale characters specified in subsequent Character String drawing orders. Devices that use the font positioning method ignore the character cell.

The application of this attribute is dependent on the value of the character precision attribute. See [“Character Strings” on page 51](#) for details.

Two formats exist for this drawing order:

- *Four-byte format*
 - CELLWI specifies the width of the character cell in GPS units.
 - CELLHI specifies the height of the character cell in GPS units.
- *Eight-byte format*

In this format, both integer and fractional values are specified for the character cell width and height. The width and width-fraction fields form a 4-byte signed value, and the height and height-fraction fields form a 4-byte signed value. A decimal point is implied between the integer part and the fractional part of each parameter.

- CELLWI specifies the width of the character cell in GPS units.
- CELLWFR specifies the fractional part of the width of the character cell in GPS units.
- CELLHI specifies the height of the character cell in GPS units.
- CELLHFR specifies the fractional part of the height of the character cell in GPS units.

The fractional parts do not exist in the drawing defaults and are set to zero when the drawing default is set into the current attribute.

For device implementations that do not ignore the character cell, when the width or height in the current character cell-size attribute is 0, Character String drawing orders will draw nothing.

This drawing order does not change the current position. Note that, for precisions 1 and 2 for some implementations, if the character cell size is specified as negative values, a mirror image of the character is generated. That is, if the cell width is negative, the character is mirrored about the Y-axis, and if the cell height is negative, the character is mirrored about the X-axis. Refer to [“Character Strings” on page 51](#) for a description of how the character cell is used on various AFP devices.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Set Character Direction (GSCD) Order

This order sets the value of the current character direction attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'3A'	GSCD order code
1	CODE	DIRECTION	X'00'–X'04'	Value for character direction: X'00' Drawing default X'01' Left to right X'02' Top to bottom X'03' Right to left X'04' Bottom to top All other values Reserved

Semantics

The Set Character Direction order sets the value of the current character direction attribute to the value specified in the order.

The character direction attribute controls the placement of the first character in the string and each succeeding character relative to the previous character.

Value Description

- X'00'** *Drawing Default.* The standard default in AFP environments is left to right (X'01').
- X'01'** *Left to right.* Characters are positioned so that, at a 0° character angle, the character reference point, which is point R in [Figure 29 on page 52](#), is coincident with the current position. A vector is then drawn from the left edge of the character box to the right edge, and successive characters are placed in the direction of this vector.
- X'02'** *Top to bottom.* Characters are positioned so that, at a 0° character angle, the character reference point, which is point T in [Figure 29 on page 52](#), is coincident with the current position. A vector is then drawn from the top edge of the character box to the bottom edge, and successive characters are placed in the direction of this vector.
- X'03'** *Right to left.* Characters are positioned so that, at a 0° character angle, the character reference point, which is point E in [Figure 29 on page 52](#), is coincident with the current position. A vector is then drawn from the right edge of the character box to the left edge, and successive characters are placed in the direction of this vector.
- X'04'** *Bottom to top.* Characters are positioned so that, at a 0° character angle, the character reference point, which is point B in [Figure 29 on page 52](#), is coincident with the current position. A vector is then drawn from the bottom edge of the character box to the top edge, and successive characters are placed in the direction of this vector.

Architecture Note: This graphics drawing order defines a function that is analogous to part of the text orientation function in presentation text, which defines an *inline* direction and the development of characters along this direction.

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'01'—Left to right.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'01'—Left to right.

Set Character Precision (GSCR) Order

This order sets the value of the current character precision attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'39'	GSCR order code
1	CODE	PREC	X'00'–X'03'	Value for character precision attribute: X'00' Drawing default X'01' String precision X'02' Character precision X'03' Stroke precision (not supported in AFP GOCA) All other values Reserved

Semantics

The Set Character Precision order sets the value of the current character precision attribute to the value specified in the order. The character precision attribute controls the type of character that is to be used for drawing character strings. Refer to [“Character Strings” on page 51](#) for a description of how character precision is defined.

Value	Description
X'00'	<i>Drawing Default.</i> This value sets the current character precision attribute to the value of the drawing default. The standard default in AFP environments is precision X'02'.
X'01'	<i>Precision 1—Device-Specific (String) Precision.</i> This precision has been implemented differently on different devices; it is not consistent among implementations. The characters are drawn using the quickest, simplest mode possible for the implementation. In this mode, the only attributes that <i>must</i> be considered when the characters are drawn are the character code point, character set, and character direction attributes. The character angle and character cell-size attributes are not guaranteed to have any effect on the appearance of characters in the string.
X'02'	<i>Precision 2—Device-Specific (Character) Precision.</i> This precision has been implemented differently on different devices; it is not consistent among implementations. The character string is drawn taking into account all the attributes to determine the positioning of the characters. The character attributes are not guaranteed to affect the appearance of the characters in the string.
X'03'	<i>Precision 3—Stroke Precision.</i> This value is not supported in AFP GOCA. If it is specified, exception EC-000E exists.

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Device-Specific (Character) Precision.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Device-Specific (Character) Precision.

Set Character Set (GSCS) Order

This order sets the value of the current character set attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'38'	GSCS order code
1	CODE	LCID	X'00'–X'FF'	Local identifier (LCID) for the character set: X'00' Drawing default X'01'–X'FE' Local identifier for the character set X'FF' Special character set

Semantics

The Set Character Set order sets the value of the current character set attribute to the value specified in the order.

When the current character set attribute is X'00', it identifies the drawing default, that is, the default from the GDD, or if not specified, the standard default character set. In AFP environments, this is the presentation device default font.

When the current character set attribute is X'01' to X'FE', it identifies the character set that is to be used to draw characters in subsequent Character String orders. The controlling environment maps the LCID to a specific font.

When the current character set attribute is X'FF', it identifies the special character set, which is implementation-defined.

Architecture Note: In AFP environments, the special character set is the presentation device default font.

Character definitions from the character set identified by the current character set attribute are modified under control of the current character precision attribute before being drawn.

Architecture Note: In MO:DCA and IPDS environments, the MO:DCA character rotation (IPDS font inline sequence) associated with the font is ignored when determining character direction and angle. However, when the font positioning method is used, the selected character direction should match the selected character rotation (font inline sequence) so that appropriate font metrics, such as character increment and A-space, are available. If a font with the required character rotation is not available to the presentation device, the spacing and positioning of characters is unpredictable.

Exception Conditions

The following exception condition causes a standard action to be taken:

EC-C300 The font identified by the value in the current character set attribute is not available.

Standard action: The standard default character set is used. In AFP environments, this is the presentation device default font.

Set Character Shear (GSCH) Order

This order sets the value of the current character shear attribute. This order is processed as a No-Op in AFP GOCA.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'35'	GSCH order code
1	UBIN	LENGTH	4	Length of following data
2-3	SBIN	HX	X'8000'–X'7FFF'	Dividend of shear ratio
4-5	SBIN	HY	X'8000'–X'7FFF'	Divisor of shear ratio

Semantics

Not applicable in AFP GOCA.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Set Color (GSCOL) Order

This order provides a shorthand way of setting the following foreground color attributes to the same value:

- Character color
- Image color
- Line color
- Marker color
- Pattern color

Architecture Note: To fill an area with the color specified by this drawing order, select the drawing default with the Set Pattern Set order, and either the drawing default or solid fill with the Set Pattern Symbol order.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'0A'	GSCOL order code
1	CODE	COL	X'00'–X'08'	Value for color attribute: X'00' Drawing default X'01' Blue X'02' Red X'03' Magenta (Pink) X'04' Green X'05' Cyan (Turquoise) X'06' Yellow X'07' Device default X'08' Color of medium All other values Reserved

Semantics

The Set Color order sets the current value of all five color attributes to the value specified in the order. Color attributes control the color of the foreground of the output primitives as they are drawn.

The standard default in AFP environments is the presentation device default color.

The color value specified by this order is prefixed with X'FF' to generate a two-byte color index value into the standard color table. See [“Color” on page 14](#).

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The action is implementation dependent.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The action is implementation dependent.

Architecture Note: If colors are simulated in AFP environments, color exceptions need not be generated.

Set Current Position (GSCP) Order

This order sets the value of the current position in the Graphics Presentation Space (GPS).

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	GSCP order code
1	UBIN	LENGTH	4	Length of following data
2-3	SBIN	XPOS	X'8000'–X'7FFF'	X _g coordinate of point
4-5	SBIN	YPOS	X'8000'–X'7FFF'	Y _g coordinate of point

Semantics

The Set Current Position order sets the value of current position to the value specified in the order.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Set Custom Line Type (GSCLT) Order

This order sets the value of the current line type attribute to a custom value.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'20'	GSCLT order code
1	UBIN	LENGTH	0– <i>n</i>	Length of following data; <i>n</i> must be less than 255 and a multiple of 4
2	UBIN	DASH0H	X'00'–X'FF'	Integer portion of length of first dash/dot
3	UBIN	DASH0FR	X'00'–X'FF'	Fractional portion of length of first dash/dot
4	UBIN	MOVE0H	X'00'–X'FF'	Integer portion of length of first move
5	UBIN	MOVE0FR	X'00'–X'FF'	Fractional portion of length of first move
6	UBIN	DASH1H	X'00'–X'FF'	Integer portion of length of second dash/dot
7	UBIN	DASH1FR	X'00'–X'FF'	Fractional portion of length of second dash/dot
8	UBIN	MOVE1H	X'00'–X'FF'	Integer portion of length of second move
9	UBIN	MOVE1FR	X'00'–X'FF'	Fractional portion of length of second move
:		:		
	UBIN	DASHFH	X'00'–X'FF'	Integer portion of length of final dash/dot
	UBIN	DASHFFR	X'00'–X'FF'	Fractional portion of length of final dash/dot
	UBIN	MOVEFH	X'00'–X'FF'	Integer portion of length of final move
	UBIN	MOVEFFR	X'00'–X'FF'	Fractional portion of length of final move

Semantics

The Set Custom Line Type order sets the value of the current line type attribute to the custom value specified in the order. The current line type attribute controls the type of line used to draw line primitives.

When setting the line type attribute, this drawing order will set it to a custom value. The Set Line Type drawing order will set the attribute to a standard value.

The first byte of the length of a dash/dot or move—this will be referred to as the H byte—specifies the integer portion of the length; the second byte—referred to as the FR byte—specifies the fractional portion. A combined value of X'0000' for a dash/dot length indicates a dash of length 0—that is, a dot. A decimal point is implied between H and FR. The fractional portion of the length is calculated by dividing FR by 256. For example, if FR= X'40', its decimal value is 64, which, divided by 256 results in a fractional component for the length of 1/4.

All dash and move lengths are expressed in units of line width.

If no dash and move lengths are provided (that is, if the LENGTH field is zero), or if all dash and move lengths are specified as zero, a solid line is drawn.

If a dash length is not zero, but on a given device is less than one presentation device pel, the length of the dash is set to one pel. Similarly, if a move length is not zero, but is less than one presentation device pel, the

move length is set to one pel. In other words, a nonzero length, no matter how small, must not become a zero length.

The standard default for the line type attribute in AFP environments is the standard line type value X'07'—Solid line.

See [“Line Type” on page 29](#) for more information on the line type attribute, including a discussion of how the dash/dot and move lengths are used to generate lines, and a discussion of standard and custom line type values.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Set Extended Color (GSECOL) Order

This order provides a shorthand way of setting the following foreground color attributes to the same value:

- Character color
- Image color
- Line color
- Marker color
- Pattern color

Architecture Note: To fill an area with the color specified by this drawing order, select the drawing default with the Set Pattern Set order, and either the drawing default or solid fill with the Set Pattern Symbol order.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'26'	GSECOL order code
1	UBIN	LENGTH	2	Length of following data
2–3	CODE	COLOR	See Table 5 on page 15	Value for color attribute

Semantics

The Set Extended Color order sets the current value of all five color attributes to the value specified in the order. Color attributes control the color of the foreground bits of the output primitives as they are drawn.

The color value specified by this order is used as a two-byte color index value into the standard color table; see [“Color” on page 14](#) for the meaning of the two-byte values. The color values supported by this order are the same as the values defined in the standard color table, and they are also the same as the values defined in the Standard OCA Color Value Table defined in the MO:DCA controlling environment; see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

The standard default in AFP environments is the presentation device default color.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-0004 The attribute value specified in the order is not valid.
Standard action: The action is implementation dependent.

EC-000E The attribute value specified in the order is not supported.
Standard action: The action is implementation dependent.

Architecture Note: If colors are simulated in AFP environments, color exceptions need not be generated.

Set Fractional Line Width (GSFLW) Order

This order sets the value of the current line width attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'11'	GSFLW order code
1	UBIN	LENGTH	2	Length of following data
2	UBIN	MH	X'00'–X'FF'	Integral multiplier of normal line width
3	UBIN	MFR	X'00'–X'FF'	Fractional multiplier of normal line width

Semantics

The Set Fractional Line Width order sets the value of the current line width attribute to the value specified in the order. The current line width attribute controls the width of line used to draw line primitives.

MH specifies the integer portion of the normal line width multiplier; MFR specifies the fractional portion of the normal line width multiplier. A combined value of X'0000' specifies the drawing default. A combined value of X'0100' represents a unity multiplier, that is, normal line width. A decimal point is implied between MH and MFR. The fractional portion of the multiplier is calculated by dividing MFR by 256. For example, if MFR=X'40', its decimal value is 64, which, divided by 256 results in a fractional component for the multiplier of 1/4.

See [“Line Width” on page 32](#) for more information on the line width attribute.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception condition causes a standard action to be taken:

EC-000E The attribute value specified in the order is not supported.

Standard action: The standard default value of the attribute is used. In AFP environments, this is a multiplier of X'0100', that is, normal line width.

Set Line End (GSLE) Order

This order sets the value of the current line end attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'1A'	GSLE order code
1	CODE	LINEEND	X'00'–X'03'	Value for line end attribute: X'00' Drawing default X'01' Flat X'02' Square X'03' Round All other values Reserved

Semantics

The Set Line End order sets the value of the current line end attribute to the value specified in the order.

The current line end attribute applies to those output primitives that are drawn as straight or curved lines and have ends; that is, not complete figures, such as Box and Full Arc. It defines the shape of the start and end of groups of contiguous straight and curved lines. If the line type is not solid, the line end attribute also defines the shape of the internal ends of the dots and dashes, even for complete figures.

The standard default in AFP environments is X'03'—Round.

See [“Line End and Line Join” on page 32](#) for details of the line-end shapes and their application.

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'03'—Round.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'03'—Round.

Set Line Join (GSLJ) Order

This order sets the value of the current line join attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'1B'	GSLJ order code
1	CODE	LINEJOIN	X'00'–X'03'	Value for line join attribute: X'00' Drawing default X'01' Bevel X'02' Round X'03' Miter All other values Reserved

Semantics

The Set Line Join order sets the value of the current line join attribute to the value specified in the order.

The current line join attribute applies to those output primitives that are drawn as straight or curved lines and have joins; that is, not complete figures, such as Box and Full Arc. The line join attribute defines the shape of the joins between contiguous straight and curved lines.

The standard default in AFP environments is X'02'—Round.

See [“Line End and Line Join” on page 32](#) for details of the line-join shapes and their application.

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Round.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Round.

Set Line Type (GSLT) Order

This order sets the value of the current line type attribute to a standard value.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'18'	GSLT order code
1	CODE	LINETYPE	X'00'–X'08'	Value for line type attribute: X'00' Drawing default; solid if none specified X'01' Dotted line X'02' Short-dashed line X'03' Dash-dot line X'04' Double-dotted line X'05' Long-dashed line X'06' Dash-double-dot line X'07' Solid line X'08' Invisible line All other values Reserved

Semantics

The Set Line Type order sets the value of the current line type attribute to the standard value specified in the order. The current line type attribute controls the type of line used to draw line primitives.

When setting the line type attribute, this drawing order will set it to a standard value. The Set Custom Line Type drawing order will set the attribute to a custom value.

The standard default for the line type attribute in AFP environments is X'07'—Solid line.

See [“Line Type” on page 29](#) for more information on the line type attribute, for guidelines on how the sequence of dashes, dots, and spaces should be generated, and for a discussion of standard and custom line type values.

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'07'—Solid line.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'07'—Solid line.

Set Line Width (GSLW) Order

This order sets the value of the current line width attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'19'	GSLW order code
1	UBIN	MH	X'00'–X'FF'	Value for line width attribute: X'00' Drawing default X'01'–X'FF' Integral multiplier of normal line width

Semantics

The Set Line Width order sets the value of the current line width attribute to the value specified in the order. This order also resets the fractional part of the line width attribute to zero. The current line width attribute controls the width of line used to draw line primitives.

MH specifies an integer multiplier of the normal line width. A value of X'01' represents a unity multiplier, that is, normal line width.

The standard default in AFP environments is a multiplier of X'01'—normal line width.

See [“Line Width” on page 32](#) for more information on the line width attribute.

Exception Conditions

The following exception condition causes a standard action to be taken:

EC-000E The attribute value specified in the order is not supported.

Standard action: The standard default value of the attribute is used. In AFP environments, this is a multiplier of X'01', that is, normal line width.

Set Marker Cell (GSMC) Order

This order sets the value of the current marker cell-size attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'37'	GSMC order code
1	UBIN	LENGTH	4, 6	Length of following data
2–3	SBIN	CELLWI	X'8000'–X'7FFF'	Width of marker cell
4–5	SBIN	CELLHI	X'8000'–X'7FFF'	Height of marker cell
The following parameters are optional:				
6–7	BITS	FLAGS		Internal flags
Bit 0		NOTDEFLT	B'0', B'1'	How to interpret a zero cell-size: B'0' Zero means drawing default B'1' Zero means a size of zero
Bits 1–15		RES	B'0000000000000000'	Reserved; only valid value

Semantics

The Set Marker Cell order sets the value of the current marker cell-size attribute to the value specified in the order.

Implementation Note: In earlier versions of AFP GOCA, the Set Marker Cell order was processed as a No-Op with a LENGTH field with value 4. Thus, some implementations will ignore this drawing order, and some will raise an exception if a LENGTH field with value 6 is encountered.

The CELLWI and CELLHI values are in GPS units.

If the value of CELLWI is a negative value, this indicates to present the marker as a mirror image in the x-direction—that is, about the Y-axis—of the normal marker symbol. Similarly, a negative CELLHI value indicates to mirror the marker about the X-axis. Note, however, that all symbols in the default marker set are symmetric in both the x and y directions, so mirror imaging will have no effect on them.

The NOTDEFLT bit indicates how to interpret a CELLWI or CELLHI value of zero.

- If NOTDEFLT = B'0' (or the FLAGS field is omitted), if either or both of CELLWI or CELLHI are X'0000', the marker cell-size is set to the drawing default value.
- If NOTDEFLT = B'1', if either or both of CELLWI or CELLHI are X'0000', the marker cell-size is set to zero. While the marker cell-size attribute is set to zero, markers will be drawn with zero size: that is, the current position will be updated, but no actual markers will be drawn.

The standard default marker cell-size in AFP environments is device dependent. However, it is recommended that the standard default marker cell-size be 7/120 of an inch for both width and height (although due to possible scaling, default-sized markers will not necessarily appear at 7/120 of an inch in the usable area).

Markers are scaled along with the rest of the GPS if scaling is necessary in the mapping from the GPS window into the usable area (object area).

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Set Marker Set (GSMS) Order

This order sets the value of the current marker set attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'3C'	GSMS order code
1	CODE	LCID	X'00'–X'FF'	Local identifier (LCID) for the marker set: X'00' Default marker set X'01'–X'FE' Local identifier for marker set (not supported in AFP GOCA) X'FF' Default marker set (not supported in AFP GOCA)

Semantics

The Set Marker Set order sets the value of the current marker set attribute to the value specified in the order.

When the value of the marker set attribute is X'00', the marker is drawn from the default marker set. See [“Markers” on page 57](#) for diagrams of the marker symbols in the default marker set.

Values X'01' to X'FF' are not supported in AFP GOCA.

Exception Conditions

The following exception condition causes a standard action to be taken:

- EC-C200** The marker set identified by the value in the current marker set attribute is not available.
Standard action: The standard default marker set is used. In AFP environments, this is the default marker set.

Set Marker Symbol (GSMT) Order

This order sets the value of the current marker symbol attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'29'	GSMT order code
1	CODE	MCPT	X'00', X'01'–X'0A', X'40'	Value of marker symbol code point: X'00' Drawing default; cross if not specified When the default marker set is selected (Marker Set = X'00'): X'01' Cross X'02' Plus X'03' Diamond X'04' Square X'05' 6-point star X'06' 8-point star X'07' Filled diamond X'08' Filled square X'09' Dot X'0A' Small circle X'40' Blank All other values Reserved

Semantics

The Set Marker Symbol order sets the value of the current marker symbol attribute to the value in the order.

See [“Markers” on page 57](#) for diagrams of the marker symbols corresponding to attribute values X'01'–X'0A' in the default marker set.

The standard default in AFP environments is X'01'—Cross.

Exception Conditions

The following exception condition causes a standard action to be taken:

- EC-C201** The code point identified by the value in the current marker symbol attribute is not defined in the current marker set.
- Standard action:** The standard default marker symbol is used. In AFP environments, this is X'01'—Cross.

Set Mix (GSMX) Order

This order provides a shorthand way of setting the following foreground mix attributes to the same value:

- Character foreground mix
- Image foreground mix
- Line foreground mix
- Marker foreground mix
- Pattern foreground mix

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'0C'	GSMX order code
1	CODE	MODE	X'00'–X'05'	Mix-mode value: X'00' Drawing default X'01' Not supported in AFP GOCA X'02' Overpaint X'03'–X'05' Not supported in AFP GOCA All other values Reserved

Semantics

The Set Mix order sets the current value of all five mix attributes to the value specified in the order. Mix attributes control the way in which the color of the foreground of a primitive is combined with the color of the presentation space.

With MODE set to X'02', the foreground pels are opaque and their color replaces the color of underlying pels in the GPS. Since this is the only foreground mix mode supported in AFP GOCA, selecting the drawing default (MODE X'00') will also default to MODE X'02'.

For a description of the meaning of the various mix modes, see ["Mix" on page 16](#).

Exception Conditions

The following exception conditions cause a standard action to be taken:

- EC-0004** The attribute value specified in the order is not valid.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Overpaint.
- EC-000E** The attribute value specified in the order is not supported.
Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Overpaint.

Set Pattern Reference Point (GSPRP) Order

This order sets the value of the current pattern reference point attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'A0'	GSPRP order code
1	UBIN	LENGTH	6	Length of following data
2	BITS	FLAGS		Internal flags
Bit 0		DEFAULT	B'0' B'1'	Set to the specified value Set to the drawing default
Bits 1–7		RES1	B'0000000'	Reserved; only valid value
3		RES2	X'00'	Reserved; only valid value
4–5	SBIN	XPOS	X'8000' – X'7FFF'	X _g coordinate of the pattern reference point
6–7	SBIN	YPOS	X'8000' – X'7FFF'	Y _g coordinate of the pattern reference point

Semantics

The Set Pattern Reference Point order sets the value of the current pattern reference point attribute to the value specified in the order.

The value of the pattern reference point attribute is used as the origin for the placement of custom patterns when filling an area. The pattern reference point is not used when filling an area either with patterns from the default pattern set or with gradients.

Note that the pattern reference point does not have to be inside an area being filled. Conceptually, the custom pattern is tiled in all directions from the pattern reference point, all the way to the edges of the GPS. Therefore, the pattern reference point precisely determines the appearance of an area filled with a custom pattern, whether or not the pattern reference point is located on the inside of that area.

If DEFAULT is B'1', the pattern reference point is set to the drawing default and the XPOS and YPOS parameters are ignored.

The standard default pattern reference point in AFP GOCA is (0,0).

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

Set Pattern Set (GSPS) Order

This order sets the value of the current pattern set attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'08'	GSPS order code
1	CODE	LCID	X'00'–X'FF'	Local identifier (LCID) for the pattern set: X'00' Default pattern set X'01'–X'FD' Pattern set containing custom patterns and/or gradients X'FE' Local identifier for the pattern set (not supported in AFP GOCA) X'FF' Default pattern set (not supported in AFP GOCA)

Semantics

The Set Pattern Set order sets the value of the current pattern set attribute to the value specified in the order.

When the value of the pattern set attribute is X'00', the pattern is drawn from the default pattern set. See [Figure 19 on page 40](#) for diagrams of the patterns in the default pattern set.

When the value of the pattern set attribute is in the range X'01' – X'FD', the pattern is either a custom pattern that has been defined in the current segment using the Begin Custom Pattern drawing order or a gradient that has been defined in the current segment using the Linear Gradient or Radial Gradient drawing orders.

The standard default in AFP environments is the default pattern set, X'00'.

Values X'FE' and X'FF' are not supported in AFP GOCA.

Exception Conditions

No exceptions are generated until the pattern set is used for area fill. See [“Begin Area \(GBAR\) Order” on page 82](#).

Set Pattern Symbol (GSPT) Order

This order sets the value of the current pattern symbol attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'28'	GSPT order code
1	CODE	PATT	<p>For pattern set X'00': X'00'–X'10', X'40'</p> <p>For pattern sets X'01' – X'FD': X'00'–X'FF'</p>	<p>Value of pattern-symbol code point: X'00' Drawing default</p> <p>When the default pattern set is selected (Pattern Set = X'00'): X'01'–X'08' Dotted patterns of decreasing density X'09' Vertical lines X'0A' Horizontal lines X'0B' Diagonal lines 1 (bottom-left to top-right) X'0C' Diagonal lines 2 (bottom-left to top-right) X'0D' Diagonal lines 1 (top-left to bottom-right) X'0E' Diagonal lines 2 (top-left to bottom-right) X'0F' No fill X'10' Solid fill X'40' Blank (processed the same as X'0F', No fill)</p> <p>All other values Reserved</p> <p>When a non-default pattern set is selected (Pattern Set = X'01' – X'FD'): X'01'–X'FF' Pattern symbol value of a custom pattern or gradient</p>

Semantics

The Set Pattern Symbol order sets the value of the current pattern symbol attribute to the value specified in the order. The value of the pattern symbol attribute determines which particular pattern from the current pattern set is used to fill the interior of subsequent areas.

See [Figure 19 on page 40](#) for diagrams of the patterns corresponding to the attribute values X'01'–X'10' in the default pattern set.

The standard default in AFP environments is X'10'. If the default pattern set is selected, this corresponds to the Solid-fill pattern.

The pattern symbol value X'00' specifies to use the drawing default pattern symbol, no matter the value of the current pattern set attribute (as long as it is a supported value). For example, if the drawing default pattern, as set by the Set Current Defaults instruction, is pattern set X'03', pattern symbol X'14', that pattern will be used if the current pattern symbol attribute is set to X'00', whether the current pattern set attribute is X'03', X'00', X'11', or any other supported value.

Set Pattern Symbol

Exception Conditions

No exceptions are generated until the pattern symbol is used for area fill. See [“Begin Area \(GBAR\) Order” on page 82](#).

Set Process Color (GSPCOL) Order

This order specifies a process color, highlight color, or named color that sets the following color attributes to the same value:

- Character color
- Image color
- Line color
- Marker color
- Pattern color

Architecture Note: To fill an area with the color specified by this drawing order, select the drawing default with the Set Pattern Set order, and either the drawing default or solid fill with the Set Pattern Symbol order.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'B2'	GSPCOL order code
1	UBIN	LENGTH	12–14	Length of following data
2		RES1	X'00'	Reserved; only valid value
3	CODE	COLSPCE	X'01', X'04', X'06', X'08', X'40'	Color space: X'01' RGB X'04' CMYK X'06' Highlight color space X'08' CIELAB X'40' Standard OCA color space
4–7		RES2	X'00000000'	Reserved; only valid value
8	UBIN	COLSIZE1	X'01'–X'08', X'10'	Number of bits in component 1; see color space definitions
9	UBIN	COLSIZE2	X'00'–X'08'	Number of bits in component 2; see color space definitions
10	UBIN	COLSIZE3	X'00'–X'08'	Number of bits in component 3; see color space definitions
11	UBIN	COLSIZE4	X'00'–X'08'	Number of bits in component 4; see color space definitions
12–n		COLVALUE	See Semantics	Color specification

Semantics

COLSPCE is a code that defines the color space and the encoding for the color specification. If the color space is invalid, exception condition EC-0004 exists. The standard action is to use the device default color. If the color space is unsupported, exception condition EC-000E exists. The standard action is to use the device default color. A more specific and preferred exception for an invalid or unsupported color space is EC-0E02. The standard action is to use the device default color.

Value

Description

X'01'

RGB color space. The color value is specified with three components. Components 1, 2, and 3 are unsigned binary numbers that specify the red, green, and blue intensity values, in that order. COLSIZE1, COLSIZE2, and COLSIZE3 are nonzero and define the number of bits used to specify each component. COLSIZE4 is reserved and should be set to zero. The intensity

Set Process Color

range for the R,G,B components is 0 to 1, which is mapped to the binary value range 0 to $(2^{\text{COLSIZE}N} - 1)$, where $N=1,2,3$.

Architecture Note: The reference white point and the chromaticity coordinates for RGB are defined in SMPTE RP 145-1987, entitled *Color Monitor Colorimetry*, and in RP 37-1969, entitled *Color Temperature for Color Television Studio Monitors*, respectively. The reference white point is commonly known as *Illuminant D₆₅₀₀* or simply *D65*. The R,G,B components are assumed to be gamma-corrected (nonlinear) with a gamma of 2.2.

X'04'

CMYK color space. The color value is specified with four components. Components 1, 2, 3, and 4 are unsigned binary numbers that specify the cyan, magenta, yellow, and black intensity values, in that order. COLSIZE1, COLSIZE2, COLSIZE3, and COLSIZE4 are nonzero and define the number of bits used to specify each component. The intensity range for the C,M,Y,K components is 0 to 1, which is mapped to the binary value range 0 to $(2^{\text{COLSIZE}N} - 1)$, 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 presentation device to generate a highlight color. The color value is specified with one to three components.

Component 1 is a two-byte unsigned binary number that specifies the highlight color number. The first highlight color is assigned X'0001', the second highlight color is assigned X'0002', and so on. The value X'0000' specifies the presentation device default color. COLSIZE1 = X'10' and defines the number of bits used to specify component 1.

Component 2 is an optional one-byte unsigned binary number that specifies a percent coverage for the specified color. Percent coverage can be any value from 0% to 100% (X'00'–X'64'). The number of distinct values supported is presentation-device dependent. If the coverage is less than 100%, the remaining coverage is achieved with color of medium. COLSIZE2 = X'00' or X'08' and defines the number of bits used to specify component 2. A value of X'00' indicates that component 2 is not specified in the color value, in which case the architected default for percent coverage is 100%. A value of X'08' indicates that component 2 is specified in the color value.

Component 3 is an optional one-byte unsigned binary number that specifies a percent shading, which is a percentage of black that is to be added to the specified color. Percent shading can be any value from 0% to 100% (X'00'–X'64'). The number of distinct values supported is presentation-device dependent. If percent coverage and percent shading are specified, the effective range for percent shading is 0% to $(100 - \text{coverage})\%$. If the sum of percent coverage plus percent shading is less than 100%, the remaining coverage is achieved with color of medium. COLSIZE3 = X'00' or X'08' and defines the number of bits used to specify component 3. A value of X'00' indicates that component 3 is not specified in the color value, in which case the architected default for percent shading is 0%. A value of X'08' indicates that component 3 is specified in the color value.

Implementation Note: The percent shading parameter is currently not supported in AFP environments.

If the percent value for component 2 or component 3 is invalid, exception condition EC-0E04 exists. The standard action is to use the maximum valid percent value.

COLSIZE4 is reserved and should be set to zero.

This is a device-dependent color space.

Architecture Notes:

1. The color that is rendered when a highlight color is specified is device dependent. For presentation devices that support colors other than black, highlight color values in the range X'0001' to X'FFFF' may be mapped to any color. For bilevel devices, the color may be simulated with a graphic pattern.

2. If the specified highlight color is “presentation device default”, devices whose default color is black use the percent coverage parameter, which is specified in component 2, to render a percent shading.
3. On printing devices, the color of medium is normally white, in which case a coverage of n % results in adding $(100-n)$ % white to the specified color, or *tinting* the color with $(100-n)$ % white. Display devices may assume the color of medium to always be white and use this algorithm to render the specified coverage.
4. The highlight color space can also specify indexed colors when used in conjunction with a Color Mapping Table (CMT) or an Indexed (IX) Color Management Resource (CMR). In that case, component 1 specifies a two-byte value that is the index into the CMT or the IX CMR, and components 2 and 3 are ignored. Note that when both a CMT and Indexed CMRs are used, the CMT is always accessed first. To preserve compatibility with existing highlight color devices, indexed color values X'0000' – X'00FF' are reserved for existing highlight color applications and devices. That is, indexed color values in the range X'0000' – X'00FF', assuming they are not mapped to a different color space in a CMT, are mapped directly to highlight colors. Indexed color values in the range X'0100' – X'FFFF', assuming they are not mapped to a different color space in a CMT, are used to access Indexed CMRs. For a description of the Color Mapping Table in MO:DCA environments, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

X'08'

CIELAB color space. The color value is specified with three components. Components 1, 2, and 3 are binary numbers that specify the L, a, b values, in that order, where L is the luminance and a and b are the chrominance differences. Component 1 specifies the L value as an unsigned binary number; components 2 and 3 specify the a and b values as signed binary numbers. COLSIZE1, COLSIZE2, and COLSIZE3 are nonzero and define the number of bits used to specify each component. COLSIZE4 is reserved and should be set to zero. The range for the L component is 0 to 100, which is mapped to the binary value range 0 to $(2^{\text{COLSIZE1}} - 1)$. The range for the a and b components is -127 to +127, which is mapped to the binary range $-(2^{\text{COLSIZE}N} - 1 - 1)$ to $+(2^{\text{COLSIZE}N} - 1 - 1)$, 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 because of the quantization error.

Architecture 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 value is specified with one component. Component 1 is an unsigned binary number that specifies a named color using a two-byte value from the Standard OCA Color Value Table. For a complete description of the Standard OCA Color Value Table, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*. COLSIZE1 = X'10' and defines the number of bits used to specify component 1. COLSIZE2, COLSIZE3, and COLSIZE4 are reserved and should be set to zero. This is a device-dependent color space.

See [Table 5 on page 15](#) for the meaning of the two-byte values.

All others

Reserved

COLSIZE1 defines the number of bits used to specify the first color component. The color component is right-aligned and padded with zeros on the left to the nearest byte boundary. For example, if COLSIZE1 = X'06', the first color component has two padding bits.

COLSIZE2 defines the number of bits used to specify the second color component. The color component is right-aligned and padded with zeros on the left to the nearest byte boundary.

Set Process Color

COLSIZE3 defines the number of bits used to specify the third color component. The color component is right-aligned and padded with zeros on the left to the nearest byte boundary.

COLSIZE4 defines the number of bits used to specify the fourth color component. The color component is right-aligned and padded with zeros on the left to the nearest byte boundary.

For COLSIZE1–COLSIZE4, if the specified value is invalid, exception condition EC-0004 exists. The standard action is to use the device default color. If the specified value is unsupported, exception condition EC-000E exists. The standard action is to use the device default color. A more specific and preferred exception for an invalid or unsupported number of bits in a color component is EC-0E05. The standard action is to use the device default color.

COLVALUE specifies the color value in the defined format and encoding. If the color value is invalid, exception condition EC-0004 exists. The standard action is to use the device default color. If the color value is unsupported, exception condition EC-000E exists. The standard action is to use the device default color. A more specific and preferred exception for an invalid or unsupported color value is EC-0E03. The standard action is to use the device default color. Note that the number of bytes specified for this parameter depends on the color space. For example, when there are 8 bits per component, an RGB color value is specified with 3 bytes, while a CMYK color value is specified with 4 bytes. If extra bytes are specified, they are ignored as long as the drawing order length is valid.

Exception Conditions

The following exception condition raises a drawing process check:

EC-0003 The order has an incorrect length.

The following exception conditions cause a standard action to be taken:

EC-0004 The attribute value specified in the order is not valid.

Standard action: The device default color is used.

EC-000E The attribute value specified in the order is not supported.

Standard action: The device default color is used

EC-0E02 The color space specified in the order is invalid or unsupported.

Standard action: The device default color is used.

EC-0E03 The color value specified in the order is invalid or unsupported.

Standard action: The device default color is used.

EC-0E04 The highlight color percent value specified in the order is invalid.

Standard action: The maximum valid percent value is used.

EC-0E05 The number of bits for a color component specified in the order is invalid or unsupported.

Standard action: The device default color is used.

Architecture Notes:

1. AFP printers should generate the specific and preferred exceptions defined for this drawing order. For example, if the color value is invalid or unsupported, AFP printers should generate EC-0E03.
2. If colors are simulated in AFP environments, color exceptions need not be generated.
3. When a color space other than the standard OCA color space is selected with this drawing order, the concept of mixing color index values in the GPS does not apply. The use of mixing rules other than “Overpaint” or “Leave Alone” is not possible.

4. For a description of color spaces and their relationships, see R. Hunt, *The Reproduction of Colour in Photography, Printing, and Television*, Fifth Edition, Fountain Press, 1995.

Chapter 8. Exception Conditions

Exception conditions are detected in the graphics processor by the environment interface when interpreting instructions and commands, and by the drawing processor when interpreting commands and orders. The detection of exception conditions is mandatory unless noted otherwise.

Set Current Defaults Instruction Exceptions

The following are the exceptions, called instruction process checks, detected when interpreting Set Current Defaults control instructions:

- IPC-0002** This instruction process check is detected:
- If the SET parameter (byte 2) is invalid or unsupported
 - If the FLAGS parameter (byte 5) bits 1–3 are not B'000', or bits 4–7 are not B'1111'
 - If an unallocated item is referenced in the MASK parameter (bytes 3-4)
- IPC-0003** This instruction process check is detected:
- If the FLAGS parameter (byte 5) bit 0 is B'0' and LENGTH is not X'04'
 - If the FLAGS parameter (byte 5) bit 0 is B'1' and the length of the immediate data (byte 6 onward) does not exactly match the length implied by the MASK parameter
- IPC-0021** This instruction process check is detected if any values in the data are invalid or unsupported.

Begin Segment Command Exceptions

The following are the exceptions, called command process checks, detected when interpreting Begin Segment commands:

- CPC-0001** Invalid command code specified.
- CPC-7001** Begin Segment APP parameter has the value B'10'.
- CPC-7082** Begin Segment APP parameter has the value B'01'.
- CPC-70C1** Invalid parameter length specified.
- CPC-70C5** Insufficient data. The segment data is less than the length specified by SEGL parameter.

Drawing Order Exceptions

A drawing process *exception condition (EC)* exists whenever the drawing processor detects an invalid or unsupported order or an invalid or unsupported parameter value on an order. Each exception condition identified by the architecture has been assigned a unique code of the form EC-xxxx.

The architecture provides control over the way in which an exception condition is to be handled, as follows:

- For each exception condition, the AFP GOCA architecture defines the action that is to be taken when the condition arises. This action is one of the following:
 - Report a drawing process check (DPC). The identifier of the DPC is the same as that of the exception condition; that is, exception condition EC-xxxx raises DPC-xxxx.
 - Perform some architecture- or implementation-defined *Standard action*. For example, for EC-C301 on the Character String order, which is the condition where a code point in the order does not refer to a valid graphic character, the architected standard action is to draw the standard default character symbol.
- The environment—for example, the IPDS environment—optionally can provide an exception handling control that causes the drawing processor to raise a drawing process check for each and every exception condition, rather than execute the standard action, if any, defined for the exception condition. This exception handling control, if provided, can specify what is to happen after the drawing process check has been raised; for example, terminate the draw function or skip to the next drawing order.

The exception conditions associated with each drawing order are listed with each order.

There are two types of exception condition detected when interpreting drawing orders:

- Those for which no architected standard action is defined
- Those that have a standard action defined

Exception Conditions without Standard Actions

This section lists those exception conditions that raise a drawing process check and that do not have a standard action defined:

- EC-0002** Retired. See [“Retired Exception Conditions” on page 197](#) for more information.
- EC-0003** Incorrect length specification. The length in the order is not a valid value for the order.
- EC-0008** Truncated order. The order about to be executed is not a complete order.
- This error can occur when the last order in a segment is being executed. This order meets one of the following conditions:
- The order is a fixed, two-byte format order, and the second byte is not in the segment.
 - The order is a long format order, and the length byte is not in the segment.
 - The order is a long or extended format order, and the number of bytes from the end of the length field to the end of the segment is less than the value of the length count.
 - The order is an extended format order, and the qualifier byte is not in the segment.
 - The order is an extended format order, and one or both of the length bytes are not in the segment.
- Note that it is valid for drawing orders to be split across a segment boundary, as long as the second segment is an appended segment.
- EC-000A** Invalid descriptor. This condition occurs when the Graphics Data Descriptor (GDD), passed in the invocation sent to the drawing processor, contains invalid bits. A drawing process check is raised. This exception may optionally be generated in MO:DCA environments.
- EC-000C** One of the following conditions has occurred within the prolog section of a segment:
- An order that is not valid in the prolog has been detected.
 - The end of the segment has been reached without an End Prolog order.
- EC-0400** The Segment Characteristics order is detected outside the prolog section of a segment. This is an optional exception.
- EC-3E00** An End Prolog order has occurred outside the prolog section of a segment.
- EC-6000** An End Area order has been executed without an unmatched Begin Area order having previously been executed.
- EC-6800** A Begin Area order has been executed after another Begin Area order, without an intervening End Area order being executed.
- EC-6801** A Begin Area order has been executed in a segment, and the end of the segment is reached without an End Area order having been executed.
- EC-6802** A supported order that is invalid within an area is detected.
- EC-9200** A Begin Image order was not executed before the Image Data order in this segment.
- EC-9201** There are insufficient, or too many, bytes of data in the Image Data order.
- EC-9300** An End Image order is executed without an unmatched Begin Image order having been executed previously.
- EC-9301** The number of Image Data orders between the Begin Image and End Image orders is not equal to the number of rows in the image, as specified by the HEIGHT parameter in the Begin Image order.
- EC-D100** A Begin Image order has been executed in a segment, and the end of the segment is reached without an End Image order having been executed.

Drawing Order Exceptions

- EC-D101** A Begin Image order has been executed in a segment, and a supported order other than a Comment, No-Operation, Image Data, or End Image order is executed.
- EC-D102** The value specified for the FORMAT parameter in a Begin Image order is not supported.
- EC-E100** A relative line starts inside GPS, but then goes outside GPS. For devices that can maintain a position outside the GPS, this is an optional exception.
- EC-E300** A partial arc started inside GPS, but then finished outside. Therefore, the calculated new current position is outside GPS. For devices that can maintain a position outside the GPS, this is an optional exception.
- EC-E302** A negative value is specified for the SWEEP angle in a Partial Arc order.
- EC-E303** A negative value is specified for the START angle in a Partial Arc order.

Exception Conditions with Standard Actions

This section lists those exception conditions that raise a drawing process check and that do have standard actions defined.

- EC-0001** Unallocated order codes. All unallocated order codes are reserved for future use. If an attempt is made to execute one of these unallocated order codes, this exception condition occurs. This exception condition is also raised when a device tries to execute an order that it does not support.
- EC-0001 takes priority over all other exception conditions when multiple exception conditions occur; for example, an unsupported order that is invalid in the current context.
- Standard action:** Skip over the order.
- EC-0004** The attribute value specified in the order is not valid.
- Standard action:** The standard default value of the attribute is used, except for the Set Color and Set Extended Color drawing orders, where the action is implementation dependent.
- EC-000D** Graphic presentation space overflow. This condition occurs when an order is executed that tries to draw something outside the Graphic Presentation space. For devices that can maintain a position outside the GPS, this is an optional exception.
- Standard action:** The action is implementation dependent.
- EC-000E** The attribute value specified in the order is not supported.
- Standard action:** The standard default value of the attribute is used, except for the Set Color and Set Extended Color drawing orders, where the action is implementation dependent.
- EC-0E02** Invalid or unsupported color space in Set Process Color, **Linear Gradient, or Radial Gradient** order.
- Standard action:** For the Set Process Color order, the device default color is used. For the Linear Gradient or Radial Gradient orders, ignore the order.
- EC-0E03** Invalid or unsupported color value in Set Process Color, **Linear Gradient, or Radial Gradient** order.
- Standard action:** For the Set Process Color order, the device default color is used. For the Linear Gradient or Radial Gradient orders, ignore the order.
- EC-0E04** Invalid **highlight color** percent value in Set Process Color, **Linear Gradient, or Radial Gradient** order.
- Standard action:** The maximum valid percent value is used.
- EC-0E05** Invalid or unsupported number of bits for a color component in Set Process Color, **Linear Gradient, or Radial Gradient** order.
- Standard action:** For the Set Process Color order, the device default color is used. For the Linear Gradient or Radial Gradient orders, ignore the order.
- EC-3400** The specific character angle requested is not supported.
- Standard action:** The closest character angle supported is used.
- EC-5E00** An End Custom Pattern order has been executed without an unmatched Begin Custom Pattern order having previously been executed.
- Standard action:** Ignore the End Custom Pattern order.
- EC-6803** The pattern set identified by the current pattern set is not **supported**. This exception condition exists even if the current pattern symbol is set to X'00'.
- Standard action:** The standard default pattern set is used. In AFP environments, this is the default pattern set.

Drawing Order Exceptions

- EC-6804** The current pattern symbol identifies an undefined symbol in the current pattern set.
Standard action: The standard default pattern symbol is used. In AFP environments, this is X'10'—Solid fill.
- EC-6805** Temporary storage overflow while drawing an area in an immediate segment.
Standard action: Drawing of the area is completed in an implementation-defined manner.
- EC-6806** While inside the definition of a custom pattern, a Begin Area order has been executed but the current pattern symbol identifies a custom pattern or gradient.
Standard action: The standard default pattern symbol is used. In AFP environments this is X'10'—Solid fill.
- EC-C000** The HAXIS or VAXIS parameter in a Box order is too large to fit the indicated corner into the size of the box.
Standard action: Corners with the largest axis that fit the box are drawn.
- EC-C001** The HAXIS or VAXIS parameter in a Box order is outside the range.
Standard action: A box with square corners is drawn.
- EC-C200** The marker set identified by the value in the current marker set attribute is not available.
Standard action: The standard default marker set is used. In AFP environments, this is the default marker set.
- EC-C201** The code point in the current marker symbol attribute is not defined in the current marker set.
Standard action: The standard default marker symbol is used. In AFP environments, this is X'01'—Cross.
- EC-C300** The font identified by the value in the current character set attribute is not available.
Standard action: The standard default character set is used. In AFP environments, this is the presentation device default font.
- EC-C301** A code point in the order does not refer to a valid graphic character.
Standard action: The standard default character symbol is used.
- EC-C302** The current character set attribute value identifies a character set definition that cannot support the functions implied by the current character precision attribute.
Standard action: The character set identified by the current character set attribute value is used, with the highest value of precision that the character set can support.
- EC-C303** The character encoding is Unicode, but the code points are not valid Unicode data. For more details on the possible causes of this exception, see page [93](#).
Standard action: The remainder of the code points are skipped.
- EC-C601** The drawing processor has detected an exceptional condition that might prevent the drawing of the arc within the normal limits of pel accuracy.
Standard action: The arc is drawn as accurately as the implementation can define. This action might produce straight lines.
- EC-D103** The value specified for the WIDTH parameter in a Begin Image order is too large to allow the environment to completely present the image.
Standard action: The width of the image is truncated to allow presentation of the smaller image.
- EC-D104** The value specified for the HEIGHT parameter in a Begin Image order is too large to allow the environment to completely present the image.

Standard action: The height of the image is truncated to allow presentation of the smaller image.

- | | |
|----------------|---|
| EC-DC00 | The value specified for the pattern set parameter in a Linear Gradient order is invalid.
Standard action: Ignore the Linear Gradient order. |
| EC-DC01 | The value specified for the pattern symbol parameter in a Linear Gradient order is invalid; a gradient cannot be assigned to pattern symbol X'00'.
Standard action: Ignore the Linear Gradient order. |
| EC-DC02 | The pattern set and pattern symbol values in a Linear Gradient order are within the valid range, but a pattern already resides at that location.
Standard action: Ignore the Linear Gradient order. |
| EC-DC03 | Insufficient storage available to process and store a linear gradient.
Standard action: Ignore the Linear Gradient order. |
| EC-DC04 | A value for one or both of the outside parameters in a Linear Gradient order is invalid.
Standard action: Use the value X'00' – None. |
| EC-DC05 | The value specified for a color stop offset in a Linear Gradient order is invalid or is less than a previous color stop offset.
Standard action: The color stop is ignored. |
| EC-DC06 | Invalid length value in the color specification in a Linear Gradient order.
Standard action: Ignore the Linear Gradient order. |
| EC-DC07 | Color space or color values in the Linear Gradient order are valid, but do not follow the rules for colors in a gradient.
Standard action: Ignore the Linear Gradient order. |
| EC-DD00 | The value specified for the pattern set parameter in a Radial Gradient order is invalid.
Standard action: Ignore the Radial Gradient order. |
| EC-DD01 | The value specified for the pattern symbol parameter in a Radial Gradient order is invalid; a gradient cannot be assigned to pattern symbol X'00'.
Standard action: Ignore the Radial Gradient order. |
| EC-DD02 | The pattern set and pattern symbol values in a Radial Gradient order are within the valid range, but a pattern already resides at that location.
Standard action: Ignore the Radial Gradient order. |
| EC-DD03 | Insufficient storage available to process and store a radial gradient.
Standard action: Ignore the Radial Gradient order. |
| EC-DD04 | A value for one or both of the outside parameters in a Radial Gradient order is invalid.
Standard action: Use the value X'00' – None. |
| EC-DD05 | The value specified for a color stop offset in a Radial Gradient order is invalid or is less than a previous color stop offset.
Standard action: The color stop is ignored. |
| EC-DD06 | Invalid length value in the color specification in a Radial Gradient order.
Standard action: Ignore the Radial Gradient order. |

Drawing Order Exceptions

- EC-DD07** Color space or color values in the Radial Gradient order are valid, but do not follow the rules for colors in a gradient.
Standard action: Ignore the Radial Gradient order.
- EC-DE00** A Begin Custom Pattern order has been executed after another Begin Custom Pattern order, without an intervening End Custom Pattern order being executed.
Standard action: Act as if an End Custom Pattern order had been received just prior to the current Begin Custom Pattern order.
- EC-DE01** A Begin Custom Pattern order has been executed in a segment, and the end of the segment is reached without an End Custom Pattern order having been executed.
Standard action: Act as if an End Custom Pattern order had been received just prior to the end of the segment.
- EC-DE02** The value specified for the pattern set parameter in a Begin Custom Pattern order is invalid.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE03** The value specified for the pattern symbol parameter in a Begin Custom Pattern order is invalid; a custom pattern cannot be assigned to pattern symbol X'00'.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE04** In a Begin Custom Pattern order, the pattern window values are ill defined: XLWIND is equal to or greater than XRWIND, or YBWIND is equal to or greater than YTWIND.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE05** The pattern set and pattern symbol parameters in a Begin Custom Pattern are within the valid range, but a pattern already resides at that location.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE06** Insufficient storage available to process and store a custom pattern.
Standard action: Skip to the End Custom Pattern drawing order, ignoring all orders making up the custom pattern definition, including this Begin Custom Pattern order.
- EC-DE07** A supported order that is invalid within a custom pattern definition is detected.
Note: The exception condition is raised when the order that is invalid is detected.
Standard action: Ignore the invalid order.
- EC-DF00** The pattern set and pattern symbol parameters in a Delete Pattern order are within the valid range, but no pattern exists with the pattern set and pattern symbol specified.
Standard action: Ignore the Delete Pattern order.
- EC-DF01** The value specified for the pattern set parameter in a Delete Pattern order is invalid.
Standard action: Ignore the Delete Pattern order.
- EC-DF02** The value specified for the pattern symbol parameter in a Delete Pattern order is invalid; pattern symbol X'00' cannot be deleted.
Standard action: Ignore the Delete Pattern order.

Chapter 9. Compliance

This chapter describes the AFP GOCA subsets that are supported in the AFP GOCA architecture.

AFP GOCA Subsets

AFP GOCA subsets are used to identify a specific level of AFP GOCA functionality. Each new (higher level) subset must incorporate the complete functionality of the previous (lower level) subset. The naming of AFP GOCA subsets and subsets of GOCA architecture is rooted in the original IBM GOCA architecture that was defined in the late 1980s. The first AFP GOCA subset, DR/2V0, follows the original GOCA naming conventions and represents a GOCA subset that contains drawing orders at level 2 and base functionality at version 0. However, using this naming scheme to identify additional AFP GOCA subsets is impractical, and therefore a new naming scheme will be used to identify AFP GOCA subsets beyond DR/2V0. This new naming scheme is defined as follows.

Each new AFP GOCA subset beyond DR/2V0 will be identified by the term GRS x , where:

- GRS = Graphics Subset
- x = level. GRS levels start with “2” = level 2. The next level can be called “3” = level 3. The GRS levels are not tied to the GOCA “DR” levels, although the starting level (“2”) is chosen to indicate a relationship to GOCA drawing order level 2 (DR/2), which is supported in the first AFP GOCA subset DR/2V0.

Based on this new naming scheme, the current DR/2V0 subset is assigned the synonym “GRS2”, and the next new subset is called GRS3 = Graphics Subset Level 3.

Base (Mandatory) Level (Version 0)

This level represents the base set of functions defined within Version 0 of GOCA. It is the minimum set of functions required to be supported in any environment. It consists of the following minimum general communication capabilities:

- Recognition of commands and modes
- Interpretation and validation of the commands within the mode
- Rejection of those commands and modes that are not supported, and return of error data, within the supported subset levels
- Reporting, on request of the environment, the supported features of the drawing process
- Reporting error conditions to the environment

The following commands are supported:

- Begin Segment (chained) in immediate mode

Drawing Order Level 2, Version 0 (DR/2V0)

This is a GOCA subset supported by printers and viewers in AFP environments. The DR/2V0 subset is also referred to as “GRS2”.

Immediate segments are a prerequisite to DR/2V0.

The following segment properties *must* be supported:

- Length
- Name (ignored in AFP GOCA)
- Chain
- Prolog
- New/Append

The coordinate type value is X'00'—2-D coordinates.

The geometric parameter format is X'00'—16-bit signed integer, high-order byte first.

The functions include straight and curved lines, areas, images, character strings, patterns, and markers. The following drawing orders must be supported:

- Begin Area (GBAR) order. The required support for INSIDE flag is Alternate Mode.
- Begin Image (GBIMG, GCBIMG) orders (format X'00' only)
- Character String (GCHST, GCCHST) orders
- Comment (GCOMT) order
- End Area (GEAR) order
- End Image (GEIMG) order
- End Prolog (GEPROL) order
- Fillet (GFLT, GCFLT) orders
- Full Arc (GFARC, GCFARC) orders
- Image Data (GIMD) order
- Line (GLINE, GCLINE) orders
- Marker (GMRK, GCMRK) orders
- No-Operation (GNOP1) order
- Relative Line (GRLINE, GCRLINE) orders
- Segment Characteristics (GSGCH) order. A check that this order is in the prolog state is optionally performed.
- Set Arc Parameters (GSAP) order
- Set Background Mix (GSBMX) order. The required support is X'00' and X'05'—Leave Alone.
- Set Character Angle (GSCA) order. The required support is 90-degree angles when applied to precision 2 symbols.
- Set Character Cell (GSCC) order
- Set Character Direction (GSCD) order
- Set Character Precision (GSCR) order. The required support is drawing default and precisions 1 and 2.
- Set Character Set (GSCS) order

- Set Character Shear (GSCH) order. The required support is drawing default and “no shear”. Other values can be treated as “no shear”, but generators should not produce these values.
- Set Color (GSCOL) order
- Set Current Position (GSCP) order
- Set Extended Color (GSECOL) order
- Set Line Type (GSLT) order
- Set Line Width (GSLW) order. The required support is normal line width, plus a further line width selectable by a multiplier of two.
- Set Marker Cell (GSMC) order. The required support is drawing default.

Note: At the time the DR/2V0 subset was defined in AFP GOCA, it was specified in this Reference that the Set Marker Cell order was to be processed as a No-Op, and the only valid value for the LENGTH field was 4, so this is the only functionality required in DR/2V0.

- Set Marker Precision (GSMP) order. The required support is drawing default and precisions 1 and 2.
- Set Marker Set (GSMS) order. The required support is drawing default (default marker set).

Note: Because the required support for marker set is drawing default, the *only* marker set available in the AFP environment is the default marker set.

- Set Marker Symbol (GSMT) order
- Set Mix (GSMX) order. The required support is X'00' and X'02'—Overpaint.
- Set Pattern Set (GSPS) order. The required support is drawing default (default pattern set).

Note: Because the required support for pattern set is drawing default, the *only* pattern set available in the AFP environment is the default pattern set.

- Set Pattern Symbol (GSPT) order

Architecture Notes:

1. Some AFP printers accept the Set Fractional Line Width (GSFLW) order.
2. Some AFP printers accept the following drawing orders and process them as No-Ops:
 - Set Pick Identifier (GSPIK, X'43'). This drawing order is in long format.
 - End Segment drawing order (X'71'). This drawing order is in fixed 2-byte format, where the second byte is reserved and should be set to X'00'.

Graphics Subset Level 3 (GRS3)

This subset contains all of subset DR/2V0 (GRS2), as well as the following additional drawing orders and functionality:

- Set Fractional Line Width (GSFLW) order
- Set Process Color (GSPCOL) order
- Box (GBOX, GCBOX) orders; **required support does not include the ability to draw boxes in a clockwise direction**
- Partial Arc (GPARC, GCPARC) orders
- Image resolution information for GOCA image in MO:DCA and IPDS Graphics Data Descriptor (GDD)
- New exception EC-C303 for TrueType/OpenType font support
- Extensions to Set Current Defaults instruction:
 - Set Process Color (SET = X'10')
 - Set Normal Line Width (SET = X'11')
- Support for clockwise arcs in addition to the support for counterclockwise arcs (which was part of DR/2V0), as specified by the determinant of the arc parameters
- Support for the full range of possible line widths in the Set Line Width (GSLW) order

Architecture note: As with DR/2V0, the Set Pick Identifier (X'43') and End Segment (X'71') drawing orders are tolerated, but not defined, in GRS3. As a recommendation, GRS3-compliant receivers should accept these two drawing orders and treat them as No-Ops, GRS3-compliant generators should not generate them, and GRS3 validators must allow them (but may generate a warning to discourage future use).

Appendix A. Mixed Object Document Content Architecture (MO:DCA) Environment

This appendix describes how graphics objects may be included within a Mixed Object Document Content Architecture (MO:DCA) document for the purpose of interchanging the graphics objects between a generating node and one or more receiving nodes. See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for a full description of the MO:DCA architecture. The Graphics Data Descriptor and Graphics Data structured fields used to carry graphics objects in MO:DCA documents are defined in the following sections.

To guarantee interchange, a MO:DCA document carrying a graphics object must include all information related to the object. The MO:DCA document must therefore contain not only the definition of the graphics object, but it must also provide linkage to the resources that the object references.

The discussion of MO:DCA structured fields is included in this appendix solely for setting the context of their use by graphics.

Compliance with MO:DCA Interchange Sets

When graphics objects are interchanged with the purpose of outputting the objects on a display, printer, or other output device, it is very important that visual fidelity be maintained as far as is possible. In an attempt to satisfy this objective, the GOCA architecture defines the following for the MO:DCA environments:

- A set of rules that must be followed by all generators when constructing graphics objects
- A set of graphics processing capabilities that are guaranteed to be supported by all receivers

In order to comply with a particular MO:DCA Interchange Set, products that generate graphics objects must only generate objects that contain graphics items and values defined in that interchange set. Including items or values not in the interchange set can result in processing exceptions being raised by the receiving processor, and exception actions being taken. However, a generator must not rely on a receiver taking these actions.

In order to conform to a particular MO:DCA Interchange Set, products that receive graphics objects and convert them using a processor for output to some device, are required to support all the graphics facilities defined in that interchange set.

Graphics Structured Fields in the MO:DCA Environment

This section describes the syntax of the Graphics Data Descriptor (GDD) and Graphics Data (GAD) structured fields in a MO:DCA document.

Graphics Data Descriptor (GDD) in the MO:DCA Environment

The GDD is a mandatory structured field in the Object Environment Group of a MO:DCA graphics object. The GDD contains GOCA control instructions that define the following:

- The drawing order subset that needs to be supported by the receiver for proper interpretation of the graphics data
- The GPS measurement units; note that these are also the DOCS measurement units
- The size and position of the GPS window that will be mapped to the MO:DCA object area
- The resolution of raster images in the object
- The graphics drawing defaults, specified by the Set Current Defaults instruction, that must be set up by the receiver

In this environment, only the following attributes can have their default values set using the Set Current Defaults instruction:

- Drawing Attributes
- Line Attributes
- Character Attributes
- Marker Attributes
- Pattern Attributes
- Arc Parameters
- Process Color Attributes
- Normal Line Width Attribute

Note: This is the same set of defaults as are supported by the Intelligent Printer Data Stream (IPDS) architecture.

Structured Field Introducer				
SF Length	X'D3A6BB'	Flags	Reserved	Self-Identifying Parameters

GDD Self-Identifying Parameters

Drawing Order Subset

This parameter has been retired for the DR/2V0 (GRS2) subset. New GOCA GRS2 generators should not specify this parameter and new receivers should ignore it. GOCA generators that generate functions that are only in a higher-level subset, such as GRS3, must not generate this parameter since there is no method to specify a subset other than DR/2V0 (GRS2). If this parameter is not specified, the functional level of the GOCA object is DR/2V0 (GRS2) or higher. If invalid bits are specified in this self-identifying parameter, EC-000A may optionally be detected.

See [Appendix C, “AFP GOCA Migration Functions”, on page 195](#) for information about this retired parameter.

Architecture Note: The obsolete [IBM AFP Data Stream Reference, S544-3202](#), allowed the Drawing Order Subset Parameter to be optional. If this parameter was not provided, the default was defined to be Drawing Order Subset level 2, version 0 (DR/2V0).

Window Specification (Mandatory)

Offset	Type	Name	Range	Meaning
0	CODE		X'F6'	Window Specification
1	UBIN	LENGTH	18	Length of following data
2	BITS	FLAGS		
Bit 0		PPS	B'0'	Picture Presentation Space: B'0' 2-D
Bit 1		ABS	B'1'	Picture Dimensions: B'1' Absolute; picture is designed for presentation in L-units (see bytes 5–9)
Bit 2		RES1	B'0'	Reserved; only valid value
Bit 3		IMGRES	B'0', B'1'	Image Resolution: B'0' Resolution not defined or non-symmetric image B'1' X and Y resolutions are equal and are defined by IMXYRES (see bytes 10–11)
Bit 4		IMGNS	B'0', B'1'	Non-symmetric image; ignored if bit 3 = B'1' B'0' Resolution not defined or symmetric image B'1' Image resolution is 120 × 144 points per inch
Bits 5–7		RES2	B'000'	Reserved; only valid value
3		RES3	X'00'	Reserved; only valid value
4	CODE	CFORMAT	X'00'	Geometric parameter format: X'00' 16-bit high-byte-first signed integer
5	CODE	UBASE	X'00' - X'01'	Unit Base for GPS: X'00' Ten inches X'01' Ten centimeters
6–7	UBIN	XRESOL	X'0001'–X'7FFF'	Number of X _g units/UBASE; must be the same as YRESOL
8–9	UBIN	YRESOL	X'0001'–X'7FFF'	Number of Y _g units/UBASE; must be the same as XRESOL
10–11	UBIN	IMXYRES	X'0000'–X'7FFF'	X'0000' Not specified X'0001'–X'7FFF' Number of image points per UBASE in X and Y directions
12–13	SBIN	XLWIND	X'8000'–X'7FFF', see note	X _g coordinate for left edge of GPS window
14–15	SBIN	XRWIND	X'8000'–X'7FFF', see note	X _g coordinate for right edge of GPS window
16–17	SBIN	YBWIND	X'8000'–X'7FFF', see note	Y _g coordinate for bottom edge of GPS window
18–19	SBIN	YTWIND	X'8000'–X'7FFF', see note	Y _g coordinate for top edge of GPS window
Note: The complete range is valid, and assumes a measurement unit of 1/1440 inch. That is, the measurement base is ten inches, and the X _g , Y _g units per unit base are 14,400.				

If invalid bits are specified in this self-identifying parameter, EC-000A may optionally be detected.

If a measurement unit other than 1/1440 inch is used, then the range values for XLWIND, XRWIND, YBWIND, and YTWIND can be determined by using the following steps.

1. Calculate the number of actual supported units per inch X as follows:
 - If the measurement base is ten inches, divide the number of supported units per ten inches by 10.
 - If the measurement base is ten centimeters, multiply the number of supported units per ten centimeters by 0.254.
2. Calculate the ratio of actual supported units per inch X to the assumed 1440 units per inch. To do this, divide X by 1440, yielding the ratio Y .
3. Calculate the new range value in the supported measurement units as follows:
 - a. Convert the old range value to base ten, then multiply it by the ratio Y .
 - b. Round to the nearest integer.

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

1. Supported units per inch:

$$2400 \div 10 = 240$$
2. Ratio of supported units per inch to 1440 units per inch:

$$240 \div 1440 = 1/6$$
3. Range at 2400 units per 10 inches:

$$\begin{aligned} \text{X}'8000' &= -32,768 \text{ (converted to base 10)} \\ -32,768 \cdot 1/6 &= -5461.3333 \\ \\ \text{X}'7FFF' &= 32,767 \text{ (converted to base 10)} \\ 32,767 \cdot 1/6 &= 5461.1667 \end{aligned}$$

Therefore, the equivalent range at 2400 units per 10 inches is -5461 to 5461, which in hexadecimal is X'EAAB' to X'1555'.

Architecture Notes:

1. The obsolete **IBM AFP Data Stream Reference**, S544-3202, allowed 4 additional reserved bytes following the YTWIND parameter. These bytes are supported by AFP GOCA receivers for migration, but new AFP GOCA generators should not generate these bytes.
2. The image resolution value specified by the IMGRES, IMGNS, and IMXYRES parameters allows a presentation device to maintain the size of GOCA images when scaling or resolution-correcting the GOCA object. In the absence of this information and any other externally-provided information on the resolution of a GOCA image, the image is mapped point-to-pel in the presentation device. In that case, the resulting image size varies with the resolution of the device.
3. The IPDS environment defines additional exceptions for invalid parameters in the IPDS version of the GDD. For example, an exception is defined for the case where the GPS Window coordinates are inconsistent.

Set Current Defaults (Optional)

Defaults can be set by the appropriate Set Current Defaults instructions. For a complete description of this instruction, see [“Set Current Defaults \(SCD\) Instruction” on page 66](#). Each occurrence of the Set Current Defaults instruction specifies a particular attribute set. The following tables show the *maximum* set of attributes allowed. Subsets of these attribute sets are also allowed, using the MASK bits as selectors for attributes in the particular attribute set. The format of the attribute sets is described in [“Set Current Defaults \(SCD\) Instruction” on page 66](#).

In the tables below, two possibilities exist. If the FLAG byte equals X'8F', the LENGTH byte would be specified as the second value shown, and the values shown in bytes 6-*n* would be specified as shown. If the FLAG byte instead equals X'0F', the LENGTH byte would be specified as the first value shown (that is, 4), and bytes 6-*n* would not be specified.

Set Current Defaults—Drawing Attributes

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 8	Length of following data
2	CODE	SET	X'00'	Drawing Attributes
3–4	BITS	MASK	X'B000'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6- <i>n</i>
6–7	CODE	COLOR	See Table 5 on page 15	Color
8	CODE	FORMIX	X'00', X'02'	Foreground mix
9	CODE	BACKMIX	X'00', X'05'	Background mix

Set Current Defaults—Line Attributes

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 8	Length of following data
2	CODE	SET	X'01'	Line Attributes
3–4	BITS	MASK	X'F000'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6- <i>n</i>
6	CODE	LINETYPE	X'00'–X'08'	Line type
7	UBIN	LINEWID	X'00'–X'FF'	Line width
8	CODE	LINEEND	X'00'–X'03'	Line end
9	CODE	LINEJOIN	X'00'–X'03'	Line join

Set Current Defaults—Character Attributes

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 19	Length of following data
2	CODE	SET	X'02'	Character Attributes
3–4	BITS	MASK	X'FC00'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6– <i>n</i>
6–9	SBIN	ANGLE	X'8000'–X'7FFF'	Character Angle X,Y
10–13	SBIN	CELLSIZE	X'8000'–X'7FFF'	Character cell-size CW,CH
14	CODE	DIRN	X'00'–X'04'	Character direction
15	CODE	PREC	X'00'–X'02'	Character precision
16	CODE	SET	X'00'–X'FF'	Character set
17–20	SBIN	SHEAR	X'8000'–X'7FFF'	Character shear X,Y

Set Current Defaults—Marker Attributes

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 11	Length of following data
2	CODE	SET	X'03'	Marker Attributes
3–4	BITS	MASK	X'5900'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6– <i>n</i>
6–9	SBIN	CELLSIZE	X'8000'–X'7FFF'	Marker cell-size width, height
10	CODE	PREC	X'00'–X'02'	Marker precision (obsolete, see Appendix C, “AFP GOCA Migration Functions”, on page 195)
11	CODE	SET	X'00'	Marker set
12	CODE	SYMBOL	X'00'–X'0A', X'40'	Marker symbol

Set Current Defaults—Pattern Attributes

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 10	Length of following data
2	CODE	SET	X'04'	Pattern Attributes
3–4	BITS	MASK	X'0910'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6– <i>n</i>
6	CODE	SET	X'00'–X'FD'	Pattern set
7	CODE	SYMBOL	X'00'–X'FF'	Pattern symbol
8–9	SBIN	XPOS	X'8000'–X'7FFF'	X _g coordinate of the pattern reference point
10–11	SBIN	YPOS	X'8000'–X'7FFF'	Y _g coordinate of the pattern reference point

Set Current Defaults—Arc Parameters

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 12	Length of following data
2	CODE	SET	X'0B'	Arc Parameters
3–4	BITS	MASK	X'F000'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6– <i>n</i>
6–7	SBIN	P	X'8000'–X'7FFF'	P parameter of arc transform
8–9	SBIN	Q	X'8000'–X'7FFF'	Q parameter of arc transform
10–11	SBIN	R	X'8000'–X'7FFF'	R parameter of arc transform
12–13	SBIN	S	X'8000'–X'7FFF'	S parameter of arc transform

Set Current Defaults—Process Color Attributes

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 18-20	Length of following data
2	CODE	SET	X'10'	Process Color Attributes
3-4	BITS	MASK	X'E000'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6-n
6	CODE	FORMIX	X'00', X'02'	Foreground mix
7	CODE	BACKMIX	X'00', X'05'	Background mix
8-19, 8-20, 8-21	CODE	PROCOLOR		Process Color value; syntax defined by Set Process Color drawing order starting with byte 2 (reserved)

Set Current Defaults—Normal Line Width Attribute

Offset	Type	Name	Range	Meaning
0	CODE		X'21'	Set Current Defaults instruction
1	UBIN	LENGTH	4, 6	Length of following data
2	CODE	SET	X'11'	Normal Line Width Attribute
3-4	BITS	MASK	X'8000'	Set Mask
5	CODE	FLAG	X'0F', X'8F'	X'0F' Use standard default X'8F' Use values in bytes 6-n
6-7	UBIN	NORMLW	X'0000'–X'FFFF'	Normal line width in 1440ths of an inch

Graphics Data (GAD) in the MO:DCA Environment

The graphics segments for a graphics object are contained within one or more GAD structured fields. Receipt of the first segment starts the drawing process. No restrictions exist on how much or how little graphics data is specified in a single GAD, except for the length limit of the structured field. A GAD, for example, can carry 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 GAD completes the last segment.

Because this environment does not support the calling of segments, all segments should be chained segments. Any unchained segments in the data are ignored.

The GAD structured field is optional in a MO:DCA graphics object and may be repeated multiple times.

Structured Field Introducer				
SF Length	X'D3EEBB'	Flags	Reserved	Begin Segment commands followed by segment data in the form of drawing orders

Syntax and semantics for the Begin Segment command are described in [“Begin Segment Command” on page 75](#).

GOCA Subsets within the MO:DCA Environment

GOCA objects in MO:DCA documents must comply with the architecture described in this Reference. MO:DCA interchange sets may restrict the GOCA content in GOCA objects.

In the MO:DCA architecture, for any of the color attribute-setting orders, there are no color values that are required to be supported. These color orders can specify any of the values allowed by the architecture. If a receiver does not support the requested value, an exception condition is optionally raised and the standard action is performed; that is, a device-dependent color is used.

All receiving products that claim to support the MO:DCA environment must support at least all the orders in GOCA subset DR/2V0.

Appendix B. Intelligent Printer Data Stream (IPDS) Environment

The Intelligent Printer Data Stream (IPDS) architecture is the strategic AFP data stream for controlling advanced function printer devices. It supports the *all-points-addressable* printing function that allows text and individual image, graphics, and barcode objects to be positioned and presented at any point on the printed page.

All IPDS printer commands are defined in self-defining field formats that are described in the *Intelligent Printer Data Stream Reference*. The reader is referred to this document for a definitive description of the architecture.

Graphics in the IPDS Environment

The Write Graphics Control command is sent to the printer to establish the control parameters and initial drawing conditions to be used in presenting the picture data. The picture segments themselves are sent to the printer as data in one or more Write Graphics commands.

The IPDS architecture supports the GOCA subsets DR/2V0 and GRS3, as described in [Chapter 9, "Compliance", on page 175](#).

IPDS Graphics Command Set

The IPDS Graphics Command Set consists of the following commands:

- Write Graphics Control (X'D684')
- Write Graphics (X'D685')

Write Graphics Control Command

The Write Graphics Control command is sent to the printer to indicate that the command sequence that follows is directed to a graphics object on the current page, overlay, or page segment that is being constructed by the device. The parameters of this command define the size, placement, and orientation of the graphics object area and establish the initial conditions for interpreting the graphics data.

Upon receiving this command the printer enters the appropriate graphics state and initializes control for processing graphics picture segments that are sent in subsequent Write Graphics commands. The End command received in graphics state terminates the processing of graphics data.

The drawing processor can be invoked in any one of three IPDS printer states, as follows:

- Page state
- Overlay state
- Page Segment state

When the drawing processor is invoked in Overlay or Page Segment state, the picture data sent to the printer is saved as part of the Overlay or Page Segment definition for later inclusion on pages via the Load Copy Control, Include Overlay, or Include Page Segment commands.

Positive acknowledgement of graphics commands in Overlay state or Page Segment state means that general syntax and validity checks have been made and that the command, or command sequence, has been accepted for processing. Additional exceptions that are detected when the data is included on the page are reported at that time, assuming that exception reporting is enabled.

Output Control Definitions

Graphics Object Areas

Pictures are presented in rectangular output areas called object areas. Object areas can be positioned at any addressable point on a page or in any Overlay or Page Segment definition and can be defined in any one of four orientations (0°, 90°, 180°, and 270°) relative to the axis of the reference system. Object areas correspond to the Usable Area (UA) defined in [“Usable Area \(UA\)” on page 14](#).

The size, position, and orientation of object areas are defined to the printer by parameters that are specified in the Write Graphics Control command.

GPS Window

The GPS window is a rectangular area within the GPS specified in GPS coordinates. This is the part of the picture that is mapped to the object area. The graphics data within this window is always trimmed by the printer, before the data is mapped to the object area.

The GPS window parameters are specified to the printer in the Write Graphics Control command.

Mapping Control Options

The data within the GPS Window can be mapped to the object area as specified by the Mapping Control Option parameter of the Write Graphics Control command. These options are:

Center and Trim	Map the center of the GPS Window to the center of the object area and present to scale. Excess picture data, if any, is trimmed at object area boundaries.
Scale to Fit	Map the center of the window to the center of the object area and scale to fit. The scaling is symmetric and the aspect ratio is preserved. All picture data within the window is always presented when this option is specified.
Scale to Fill	Map the center of the window to the center of the object area and scale to fill. The scaling may be asymmetric and the aspect ratio may not be preserved. All picture data within the window is always presented when this option is specified.
Position and Trim	Map the upper left-hand corner of the window to an offset point within the object area and present to scale. Excess picture data, if any, is trimmed at object area boundaries.

Mapping Defaults

If this parameter is omitted, Position and Trim is used. Excess picture data, if any, is trimmed at page boundaries and the offset position is defined to be the origin of the object area.

Write Graphics Control Data

The Write Graphics Control data is made up of three consecutive self-defining fields, as follows:

- Graphics Area Position (GAP)
- Graphics Output Control (GOC)
- Graphics Data Descriptor (GDD)

Graphics Area Position

This self-defining field defines the position and orientation of the Graphics object area relative to a reference coordinate system. It is a mandatory field in the Write Graphics Control command.

Graphics Output Control

This self-defining field specifies the size of the graphics object area and the mapping option for the graphics object. It is an optional data field in the Write Graphics Control command. If this field is omitted, the size of the graphics object area is made equal to the size of the GPS window, as specified in the Graphics Data Descriptor, and the Position and Trim option applies, where the offset origin position is defined to be the same as the object area origin.

It is an exception if there is an attempt to present data outside the boundary of the logical page.

Graphics Data Descriptor

This is a mandatory self-defining field in the Write Graphics Control command. It specifies the parameters that define the GPS Window in GPS and sets the drawing default conditions.

Write Graphics Command

The Write Graphics command transmits graphics data to the printer. The data that is carried in this command consists of picture segments that in turn contain the drawing orders that define the picture in GPS.

The segments that are sent to the printer are of two types:

- Chained
- Unchained

The type is indicated by the flag specified in the Begin Segment header.

The chained segments are the picture. The unchained segments are ignored, since calling of segments is not supported in AFP GOCA.

All segments sent to the printer are executed in *immediate mode*. That is, the drawing orders, except for unchained segments, are executed as they are received and are *not retained or stored* as named segments. The receipt of the first “chained segment” is an implicit command to the printer to start the drawing process.

There are no restrictions on how much, or how little, data is sent to the printer in a single IPDS Write Graphics command, except for the 32K length limit of the command. A Write Graphics 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.

The Begin Segment command supported by IPDS printers is shown in [“Begin Segment Command” on page 75](#).

Additional Related Commands

The following commands are used for query and resource management functions. Only an overview of these commands is presented in this document. They are described in detail in the *Intelligent Printer Data Stream Reference*.

Sense Type and Model (STM)

Requests information from the printer that identifies the type and model of the device and the command sets supported. The information requested is returned in the Special Data Area of the Acknowledge Reply to the STM command. The command sets and data levels supported are also returned as part of the acknowledgement data.

Execute Order Homestate—Obtain Printer Characteristics (XOH OPC)

Requests information from the printer that identifies various characteristics of the device. The characteristics include information about the printable area currently available, symbol-set support, image and font resolution, and other device characteristics.

Execute Order Anystate—Request Resource List (XOA RRL)

Requests the printer to return a specified list of available resources—that is, fonts, overlays, and page segments—in the Acknowledge Reply to this command. This information can be used by host programs to perform a variety of resource management functions.

Load Font Equivalence (LFE)

This command is sent to the printer to map Local Identifiers referenced in graphics to a specified font in the printer.

IPDS Environment

The correlation function provided by this command is independent of any specific font technology implemented by the printing device. That is, the device can resolve this mapping to stored font patterns downloaded from the host, or from permanently resident patterns.

The same font resource can be used for text, graphics, and bar code data.

Font Commands

The host can use commands defined in the IPDS Loaded Font command set and Device Control command set to download and manage fonts in the printer. The following commands are provided:

- Activate Resource
- Deactivate Font
- Load Code Page
- Load Code Page Control
- Load Font
- Load Font Character Set Control
- Load Font Control
- Load Font Index
- Load Symbol Set

IPDS Exceptions

In the IPDS environment, GOCA exception conditions are mapped to IPDS exceptions and reported. The mapping is shown below in [Table 15](#).

To map a GOCA exception condition to an IPDS exception, the general rule is simply to add a X'03' on the front of the four digits of the GOCA exception condition: GOCA exception condition EC-9301 becomes IPDS exception X'0393..01'. However, there are exception conditions that do not follow the general rule—these are noted in [Table 15](#).

Table 15. Mapping from GOCA Exception Condition to IPDS Exception

GOCA exception condition	IPDS exception
IPC-0002	X'0300..02'
IPC-0003	X'0300..03'
IPC-0021	X'0300..21'
CPC-0001	X'0300..01'
CPC-7001	X'0370..01'
CPC-7082	X'0370..82'
CPC-70C1	X'0370..C1'
CPC-70C5	X'0370..C5'
EC-0001	X'0300..01'
EC-0002 (retired)	X'0300..02'
EC-0003	X'0300..03'
EC-0004	X'0300..04'
EC-0008	X'0300..08'
EC-000A	None; exception condition not reported in IPDS environment (see “Note” on page 194)
EC-000C	X'0300..0C'
EC-000D	X'0300..0D'

Table 15 Mapping from GOCA Exception Condition to IPDS Exception (cont'd.)

GOCA exception condition	IPDS exception
EC-000E	X'0300..0E'
EC-0400	X'0304..00'
EC-0E02	X'020E..02' (see "Note" on page 194)
EC-0E03	X'020E..03' (see "Note" on page 194)
EC-0E04	X'020E..04' (see "Note" on page 194)
EC-0E05	X'020E..05' (see "Note" on page 194)
EC-3400	X'0334..00'
EC-3E00	X'033E..00'
EC-5E00	X'035E..00'
EC-6000	X'0360..00'
EC-6800	X'0368..00'
EC-6801	X'0368..01'
EC-6802	X'0368..02'
EC-6803	X'0368..03'
EC-6804	X'0368..04'
EC-6805	X'0368..05'
EC-6806	X'0368..06'
EC-9200	X'0392..00'
EC-9201	X'0392..01'
EC-9300	X'0393..00'
EC-9301	X'0393..01'
EC-C000	X'03C0..00'
EC-C001	X'03C0..01'
EC-C200	X'03C2..00'
EC-C201	X'03C2..01'
EC-C202 (obsolete)	X'03C2..02'
EC-C300	X'03C3..00'
EC-C301	X'03C3..01'
EC-C302	X'03C3..02'
EC-C303	X'03C3..03'
EC-C601	X'03C6..01'
EC-D100	X'03D1..00'
EC-D101	X'03D1..01'
EC-D102	X'03D1..02'
EC-D103	X'03D1..03'
EC-D104	X'03D1..04'
EC-DC00	X'03DC..00'

IPDS Environment

Table 15 Mapping from GOCA Exception Condition to IPDS Exception (cont'd.)

GOCA exception condition	IPDS exception
EC-DC01	X'03DC..01'
EC-DC02	X'03DC..02'
EC-DC03	X'03DC..03'
EC-DC04	X'03DC..04'
EC-DC05	X'03DC..05'
EC-DC06	X'03DC..06'
EC-DC07	X'03DC..07'
EC-DD00	X'03DD..00'
EC-DD01	X'03DD..01'
EC-DD02	X'03DD..02'
EC-DD03	X'03DD..03'
EC-DD04	X'03DD..04'
EC-DD05	X'03DD..05'
EC-DD06	X'03DD..06'
EC-DD07	X'03DD..07'
EC-DE00	X'03DE..00'
EC-DE01	X'03DE..01'
EC-DE02	X'03DE..02'
EC-DE03	X'03DE..03'
EC-DE04	X'03DE..04'
EC-DE05	X'03DE..05'
EC-DE06	X'03DE..06'
EC-DE07	X'03DE..07'
EC-DF00	X'03DF..00'
EC-DF01	X'03DF..01'
EC-DF02	X'03DF..02'
EC-E100	X'03E1..00'
EC-E300	X'03E3..00'
EC-E302	X'03E3..02'
EC-E303	X'03E3..03'
Note: This exception condition does not follow the general rule for mapping described above.	

If a GOCA exception condition has a Standard Action shown in this Reference, that action is ignored in the IPDS environment; instead, the IPDS Alternate Exception Action or Page Continuation Action is performed, when appropriate.

Appendix C. AFP GOCA Migration Functions

Introduction

This appendix:

- Describes migration functions that may occur in an AFP GOCA object.

General

The objective in defining migration functions is twofold:

1. To allow existing applications to run unchanged.
2. To provide a clear growth direction for future applications.

Migration Functions

The migration functions are divided into the following categories:

1. **Obsolete functions.** These are attributes, drawing orders, and parameters that will be accepted but ignored. New products must not generate these functions.
2. **Obsolete exception conditions.** These are exception conditions that will be accepted but ignored. New products must not generate these functions.
3. **Retired functions.** These are drawing orders and parameters whose use has been retired except for specific products. These specific products may use these functions. All other products should not use these functions; that is, generators should not generate these functions and receivers may ignore them.
4. **Retired exception conditions.** These are exception conditions whose use has been retired except for specific products. These specific products may report these exception conditions. All other products should not report these exception conditions.

Obsolete Functions

Obsolete Attributes

Marker Precision Attribute

The marker precision attribute was used in AFP GOCA as a method to allow implementations to draw markers in a device-dependent fashion, rather than necessarily using the marker attributes. Previously, the attribute was defined in the following way in [“Markers” on page 57](#):

The position and appearance of a marker are dependent on the value of the marker precision attribute, as follows:

Precision 1 *String Precision.* In AFP GOCA, the size of the marker symbols in the default marker set are device dependent. The marker is positioned at a specified point, or at the current position.

Precision 2 *Character Precision.* In AFP GOCA, this is the same as Precision 1—String Precision.

More information about the marker precision attribute and the meaning of its possible values can be found in the description of the drawing order used to set the marker precision attribute, the obsolete [Set Marker Precision \(GSMP\) Order](#), shown in the next section.

Obsolete Drawing Orders

Set Marker Precision (GSMP) Order

This order sets the value of the current marker precision attribute.

Syntax

Offset	Type	Name	Range	Meaning
0	CODE		X'3B'	GSMP order code
1	CODE	PREC	X'00'–X'03'	Value for marker precision attribute: X'00' Drawing default X'01' String precision X'02' Character precision X'03' Stroke precision (not supported in AFP GOCA) All other values Reserved

Semantics

The Set Marker Precision order sets the value of the current marker precision attribute to the value specified in the order.

Value	Description
X'00'	<i>Drawing Default.</i> This value sets the current marker precision attribute to the value of the drawing default. The standard default in AFP environments is X'02'—Character precision.
X'01'	<i>Precision 1—String Precision.</i> The markers are drawn using the quickest, simplest mode possible in the device. In this mode, the only attributes that must, as a minimum, be considered when the markers are drawn are the marker symbol and the marker set. The positioning of the marker can be approximate.
X'02'	<i>Precision 2—Character Precision.</i> In AFP GOCA, this is treated the same as precision 1.
X'03'	<i>Precision 3—Stroke Precision.</i> This value is not supported in AFP GOCA.

Exception Conditions

The following exception conditions cause a standard action to be taken:

EC-0004	The attribute value specified in the order is not valid. Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Character precision.
EC-000E	The attribute value specified in the order is not supported. Standard action: The standard default value of the attribute is used. In AFP environments, this is X'02'—Character precision.

Obsolete Exception Conditions

EC-C202 The current marker set attribute value identifies a marker set definition that cannot support the functions implied by the current marker precision attribute.

Standard action: The marker set identified by the current marker set attribute value is used, with the lowest value of precision that the marker set can support.

Retired Functions

Retired Parameters

Drawing Order Subset Parameter

The use of this parameter in the GDD is restricted to pre-2012 applications that generate or receive DR/2V0 (GRS2) AFP GOCA objects.

Offset	Type	Name	Range	Meaning
0	CODE		X'F7'	Drawing Order Subset
1	UBIN	LENGTH	7	Length of following data
2	CODE		X'B0'	Drawing order subset
3–4		RES	X'0000'	Reserved; only valid value
5	CODE	SUBLEV	X'02'	Drawing order subset level 2.0
6	CODE	VERSION	X'00'	Version 0
7	UBIN	LENGTH	X'01'	Length of following field
8	CODE	GEOM	X'00'	Coordinate formats in data: X'00' 16-bit high-byte-first signed integer

If invalid bits are specified in this self-identifying parameter, EC-000A may optionally be detected.

Retired Exception Conditions

EC-0002 A reserved byte or bit in the order is not set to zero. This is an optional exception.

The use of this exception condition is restricted to pre-May-2012 AFP GOCA receivers. New AFP GOCA receivers should not report this exception condition.

Note: Some pre-May-2012 receivers might report this exception condition when new functions are encountered that use previously reserved bits.

Appendix D. Cross-References

This appendix provides tables that list:

- AFP GOCA commands sorted by identifier
- AFP GOCA commands sorted by acronym
- AFP GOCA control instructions sorted by identifier
- AFP GOCA control instructions sorted by acronym
- AFP GOCA drawing orders sorted by identifier
- AFP GOCA drawing orders sorted by acronym

AFP GOCA Commands Sorted by Identifier

Table 16. Commands Sorted by ID

Identifier	Command Name	Acronym	Page
X'70'	Begin Segment	BSI	75

AFP GOCA Commands Sorted by Acronym

Table 17. Commands Sorted by Acronym

Acronym	Identifier	Command Name	Page
BSI	X'70'	Begin Segment	75

AFP GOCA Control Instructions Sorted by Identifier

Table 18. Control Instructions Sorted by ID

Identifier	Instruction Name	Acronym	Page
X'21'	Set Current Defaults	SCD	66

AFP GOCA Control Instructions Sorted by Acronym

Table 19. Control Instructions Sorted by Acronym

Acronym	Identifier	Instruction Name	Page
SCD	X'21'	Set Current Defaults	66

AFP GOCA Drawing Orders Sorted by Identifier

Table 20. Drawing Orders Sorted by ID

Identifier	Drawing Order Name	Acronym	Page
X'00'	No-Operation	GNOP1	118
X'01'	Comment	GCOMT	95
X'04'	Segment Characteristics	GSGCH	129
X'08'	Set Pattern Set	GSPS	158
X'0A'	Set Color	GSCOL	142
X'0C'	Set Mix	GSMX	156
X'0D'	Set Background Mix	GSBMX	132
X'11'	Set Fractional Line Width	GSFLW	147
X'18'	Set Line Type	GSLT	150
X'19'	Set Line Width	GSLW	151
X'1A'	Set Line End	GSLE	148
X'1B'	Set Line Join	GSLJ	149
X'20'	Set Custom Line Type	GSCLT	144
X'21'	Set Current Position	GSCP	143
X'22'	Set Arc Parameters	GSAP	130
X'26'	Set Extended Color	GSECOL	146
X'28'	Set Pattern Symbol	GSPT	159
X'29'	Set Marker Symbol	GSMT	155
X'33'	Set Character Cell	GSCC	134
X'34'	Set Character Angle	GSCA	133
X'35'	Set Character Shear	GSCH	141
X'37'	Set Marker Cell	GSMC	152
X'38'	Set Character Set	GSCS	140
X'39'	Set Character Precision	GSCR	138
X'3A'	Set Character Direction	GSCD	136
X'3B'	Set Marker Precision (<i>obsolete</i>)	GSMP	196
X'3C'	Set Marker Set	GSMS	154
X'3E'	End Prolog	GEPROL	104
X'43'	Set Pick Identifier	GSPIK	Note 1
X'5E'	End Custom Pattern	GECP	102
X'60'	End Area	GEAR	101
X'68'	Begin Area	GBAR	82
X'71'	End Segment		Note 2
X'80'	Box at Current Position	GCBOX	90

Table 20 Drawing Orders Sorted by ID (cont'd.)

Identifier	Drawing Order Name	Acronym	Page
X'81'	Line at Current Position	GCLINE	110
X'82'	Marker at Current Position	GCMRK	116
X'83'	Character String at Current Position	GCCHST	92
X'85'	Fillet at Current Position	GCFLT	105
X'87'	Full Arc at Current Position	GCFARC	107
X'91'	Begin Image at Current Position	GCBIMG	87
X'92'	Image Data	GIMD	109
X'93'	End Image	GEIMG	103
X'A0'	Set Pattern Reference Point	GSPRP	157
X'A1'	Relative Line at Current Position	GCRLINE	127
X'A3'	Partial Arc at Current Position	GCPARC	119
X'A5'	Cubic Bezier Curve at Current Position	GCCBEZ	97
X'B2'	Set Process Color	GSPCOL	161
X'C0'	Box at Given Position	GBOX	90
X'C1'	Line at Given Position	GLINE	110
X'C2'	Marker at Given Position	GMRK	116
X'C3'	Character String at Given Position	GCHST	92
X'C5'	Fillet at Given Position	GFLT	105
X'C7'	Full Arc at Given Position	GFARC	107
X'D1'	Begin Image at Given Position	GBIMG	87
X'DE'	Begin Custom Pattern	GBCP	84
X'DF'	Delete Pattern	GDPT	99
X'E1'	Relative Line at Given Position	GRLINE	127
X'E3'	Partial Arc at Given Position	GPARC	119
X'E5'	Cubic Bezier Curve at Given Position	GCBEZ	96
X'FEDC'	Linear Gradient	GLGD	112
X'FEDD'	Radial Gradient	GRGD	122
Notes:			
<ol style="list-style-type: none"> The Set Pick Identifier (X'43') long-format drawing order is not formally part of AFP GOCA, but is accepted by some AFP printers and treated as a No-Op. The End Segment (X'71') fixed two-byte drawing order is not formally part of AFP GOCA, but is accepted by some AFP printers and treated as a No-Op. 			

AFP GOCA Drawing Orders Sorted by Acronym

Table 21. Drawing Orders Sorted by Acronym

Acronym	Identifier	Drawing Order Name	Page
GBAR	X'68'	Begin Area	82
GBCP	X'DE'	Begin Custom Pattern	84
GBIMG	X'D1'	Begin Image at Given Position	87
GBOX	X'C0'	Box at Given Position	90
GCBEZ	X'E5'	Cubic Bezier Curve at Given Position	96
GCBIMG	X'91'	Begin Image at Current Position	87
GCBOX	X'80'	Box at Current Position	90
GCCBEZ	X'A5'	Cubic Bezier Curve at Current Position	97
GCCHST	X'83'	Character String at Current Position	92
GCFARC	X'87'	Full Arc at Current Position	107
GCFLT	X'85'	Fillet at Current Position	105
GCHST	X'C3'	Character String at Given Position	92
GCLINE	X'81'	Line at Current Position	110
GCMRK	X'82'	Marker at Current Position	116
GCOMT	X'01'	Comment	95
GCPARC	X'A3'	Partial Arc at Current Position	119
GCRLINE	X'A1'	Relative Line at Current Position	127
GDPT	X'DF'	Delete Pattern	99
GEAR	X'60'	End Area	101
GECP	X'5E'	End Custom Pattern	102
GEIMG	X'93'	End Image	103
GEPROL	X'3E'	End Prolog	104
GFARC	X'C7'	Full Arc at Given Position	107
GFLT	X'C5'	Fillet at Given Position	105
GIMD	X'92'	Image Data	109
GLGD	X'FEDC'	Linear Gradient	112
GLINE	X'C1'	Line at Given Position	110
GMRK	X'C2'	Marker at Given Position	116
GNOP1	X'00'	No-Operation	118
GPARC	X'E3'	Partial Arc at Given Position	119
GRGD	X'FEDD'	Radial Gradient	122
GRLINE	X'E1'	Relative Line at Given Position	127
GSAP	X'22'	Set Arc Parameters	130
GSBMX	X'0D'	Set Background Mix	132

Table 21 Drawing Orders Sorted by Acronym (cont'd.)

Acronym	Identifier	Drawing Order Name	Page
GSCA	X'34'	Set Character Angle	133
GSCC	X'33'	Set Character Cell	134
GSCD	X'3A'	Set Character Direction	136
GSCH	X'35'	Set Character Shear	141
GSCLT	X'20'	Set Custom Line Type	144
GSCOL	X'0A'	Set Color	142
GSCP	X'21'	Set Current Position	143
GSCR	X'39'	Set Character Precision	138
GSCS	X'38'	Set Character Set	140
GSECOL	X'26'	Set Extended Color	146
GSFLW	X'11'	Set Fractional Line Width	147
GSGCH	X'04'	Segment Characteristics	129
GSLE	X'1A'	Set Line End	148
GSLJ	X'1B'	Set Line Join	149
GSLT	X'18'	Set Line Type	150
GSLW	X'19'	Set Line Width	151
GSMC	X'37'	Set Marker Cell	152
GSMP	X'3B'	Set Marker Precision (<i>obsolete</i>)	196
GSMS	X'3C'	Set Marker Set	154
GSMT	X'29'	Set Marker Symbol	155
GSMX	X'0C'	Set Mix	156
GSPCOL	X'B2'	Set Process Color	161
GSPIK	X'43'	Set Pick Identifier	Note 1
GSPRP	X'A0'	Set Pattern Reference Point	157
GSPS	X'08'	Set Pattern Set	158
GSPT	X'28'	Set Pattern Symbol	159
	X'71'	End Segment	Note 2

Notes:

1. The Set Pick Identifier (X'43') long-format drawing order is not formally part of AFP GOCA, but is accepted by some AFP printers and treated as a No-Op.
2. The End Segment (X'71') fixed two-byte drawing order is not formally part of AFP GOCA, but is accepted by some AFP printers and treated as a No-Op.

Notices

The AFP Consortium or consortium member companies might have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents.

The following statement does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: AFP Consortium PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. The AFP Consortium might make improvements and/or changes in the architecture described in this publication at any time without notice.

Any references in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this architecture and use of those Web sites is at your own risk.

The AFP Consortium may use or distribute any information you supply in any way it believes appropriate without incurring any obligation to you.

This information contains examples of data and reports used in daily business operations. To illustrate them in a complete manner, some examples include the names of individuals, companies, brands, or products. These names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

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

AFPC and AFP Consortium are trademarks of the AFP Consortium.

These terms are registered trademarks of the International Business Machines Corporation in the United States, other countries, or both:

IBM

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, other countries, or both:

ACMA

Advanced Function Presentation

AFP

AFPCC

AFP Color Consortium

AFP Color Management Architecture

Bar Code Object Content Architecture

BCOCA

CMOCA

Color Management Object Content Architecture

InfoPrint

Intelligent Printer Data Stream

IPDS

Mixed Object Document Content Architecture

MO:DCA

Ricoh

Other company, product, or service names may be trademarks or service marks of others.

Glossary

This glossary contains terms that apply to the Advanced Function Presentation (AFP) Architecture and also terms that apply to other related presentation architectures.

Note: Only changes having to do with newly-added GOCA functionality in this edition are marked in color with a colored revision bar to the left. All other changes—terms or definitions that have been added, deleted, or reworded—are not marked.

If you do not find the term that you are looking for, please refer to the *IBM Dictionary of Computing*, document number ZC20-1699 or the *InfoPrint Dictionary of Printing*.

The following definitions are provided as supporting information only, and are not intended to be used as a substitute for the semantics described in the body of this reference.

A

absolute coordinate. One of the [coordinates](#) that identify the location of an addressable point with respect to the [origin](#) of a specified [coordinate system](#). Contrast with [relative coordinate](#).

absolute move. A method used to designate a new [presentation position](#) by specifying the distance from the designated axes to the new presentation position. The reference for locating the new presentation position is a fixed position as opposed to the current presentation position.

absolute positioning. The establishment of a position within a [coordinate system](#) as an offset from the coordinate system [origin](#). Contrast with [relative positioning](#).

addressable position. A position in a [presentation space](#) or on a [physical medium](#) that can be identified by a coordinate from the [coordinate system](#) of the presentation space or physical medium. See also [picture element](#). Synonymous with [position](#).

Advanced Function Presentation (AFP). An open architecture for the management of presentable information that is developed by the [AFP Consortium \(AFPC\)](#). AFP comprises a number of [data stream](#) and data object architectures:

- [Mixed Object Document Content Architecture \(MO:DCA\)](#); formerly referred to as AFPDS
- [Intelligent Printer Data Stream \(IPDS\)](#)
- [AFP Line Data Architecture](#)

- [Bar Code Object Content Architecture \(BCOCA\)](#)
- [Color Management Object Content Architecture \(CMOCA\)](#)
- [Font Object Content Architecture \(FOCA\)](#)
- [Graphics Object Content Architecture for AFP \(AFP GOCA\)](#)
- [Image Object Content Architecture \(IOCA\)](#)
- [Metadata Object Content Architecture \(MOCA\)](#)
- [Presentation Text Object Content Architecture \(PTOCA\)](#)

AEA. See [alternate exception action](#).

AFP. See [Advanced Function Presentation](#).

AFP Consortium (AFPC). A formal open standards body that develops and maintains [AFP](#) architecture. Information about the consortium can be found at www.apfcinc.org.

AFP data stream. A presentation data stream that is processed in [AFP](#) environments. The [MO:DCA](#) architecture defines the strategic AFP [interchange](#) data stream. The [IPDS](#) architecture defines the strategic AFP printer data stream.

AFPDS. A term formerly used to identify the composed-page [MO:DCA](#)-based data stream interchanged in [AFP](#) environments. See also [MO:DCA](#) and [AFP data stream](#).

AFP GOCA. See [Graphics Object Content Architecture for Advanced Function Presentation](#).

AFP Line Data Architecture. An AFP architecture that controls formatting of unformatted text data, often called line data, using a Page Definition (PageDef).

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 [character](#)s can appear. Only the cells are addressable. An example of all points addressability is the positioning of [text](#), graphics, and [image](#)s 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

anamorphic scaling • character

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](#).

ANSI. See [American National Standards Institute](#).

APA. See [all points addressable](#).

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. 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.

aspect ratio. The ratio of the horizontal size of a picture to the vertical size of the picture.

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](#).

B

background. (1) The part of a [presentation space](#) that is not occupied with [object data](#). Contrast with [foreground](#). (2) In [GOCA](#), that portion of a graphics primitive that is mixed into the presentation space under the control of the current values of the [background mix](#) and background [color attributes](#). (3) In [GOCA](#), that portion of a character cell that does not represent a [character](#).

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](#).

Bar Code Object Content Architecture (BCOCA). An architected collection of [constructs](#) used to [interchange](#) and present bar code data.

base-and-towers concept. A conceptual illustration of an architecture that shows the architecture as a base with optional towers. The base and the towers represent different degrees of function achieved by the architecture.

baseline direction (B). The direction in which successive lines of [text](#) appear on a [logical page](#). Synonymous with [B direction](#).

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](#).

BCOCA. See [Bar Code Object Content Architecture](#).

B direction. Synonymous with [baseline direction](#).

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.

bilevel custom pattern. A [custom pattern](#) that is uncolored at definition time, then has a single color assigned to it when it is used to fill an area. Contrast with [full-color custom pattern](#).

BITS. A data type for architecture [syntax](#), indicating one or more bytes to be interpreted as bit string information.

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](#).

C

CCSID. See [Coded Character Set Identifier](#).

CGCSGID. See [Coded Graphic Character Set Global Identifier](#).

CHAR. A data type for architecture [syntax](#), indicating one or more bytes to be interpreted as [character](#) information.

character. 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](#).

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 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 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 metrics. Measurement information that defines individual [character](#) values such as height, width, and space. Character metrics can be expressed in specific fixed units, such as [pels](#), or in relative units that are independent of both the [resolution](#) and the size of the [font](#). Often included as part of the more general term font metrics. See also [character set metrics](#) and [font metrics](#).

character origin. The point within the graphic pattern of a [character](#) that is to be aligned with the [presentation position](#). See also [character reference point](#).

character pattern. The scan [pattern](#) for a [graphic character](#) of a particular size, style, and weight.

character positioning. A method used to determine where a [character](#) is to appear in a [presentation space](#) or on a [physical medium](#).

character precision. The acceptable amount of variation in the appearance of a [character](#) on a [physical medium](#) from a specified ideal appearance, including no acceptable variation. Examples of appearance characteristics that can vary for a character are [character shape](#) and character position.

character reference point. The [origin](#) of a [character coordinate system](#). The X axis is the [character baseline](#). See also [character origin](#).

character rotation. The alignment of a [character](#) with respect to its [character baseline](#), measured in degrees in a clockwise direction. Examples are 0°, 90°, 180°, and 270°. Zero-degree character rotation exists when a character is in its customary alignment with the baseline. Contrast with [rotation](#).

character set. A finite set of different [graphic characters](#) or [control characters](#) that is complete for a given purpose. For example, the character set in ISO Standard 646, *7-Bit Coded Character Set for Information Processing Interchange*.

character set attribute. An [attribute](#) used to specify a [coded font](#).

character set metrics. The measurements used in a [font](#). Examples are height, width, and [character increment](#) for each [character](#) of the font. See also [character metrics](#) and [font metrics](#).

character shape. The visual representation of a [graphic character](#).

character 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](#).

CIELAB color space. Internationally accepted [color model](#) used as a standard to define color within the graphic arts industry, as well as other industries. L*, a*, and b* are plotted at right angles to one another. Equal distances in the space represent approximately equal color difference.

clipping. Eliminating those parts of a picture that are outside of a clipping boundary such as a viewing window or [presentation space](#). Synonymous with [trimming](#).

CMOCA. See [Color Management Object Content Architecture](#).

CMR. See [color management resource](#).

CMYK color space. (1) The [color model](#) used in four-color printing. Cyan, magenta, and yellow, the subtractive primary colors, are used with black to effectively create a multitude of other colors. (2) The primary colors used together in printing to effectively create a multitude of other colors: cyan, magenta, yellow, and black. Based on the subtractive color theory, the primary colors used in four-color printing processes.

CODE. A data type for architecture [syntax](#) that indicates an [architected](#) constant to be interpreted as defined by the architecture.

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 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 character](#)s 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 character](#)s 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 management. The technology to calibrate the color of input devices (such as scanners or digital cameras), display devices, and output devices (such as printers or offset presses).

Color Management Object Content Architecture (CMOCA). An architected collection of [constructs](#) used for the interchange and presentation of the color management information required to render a print file, document, group of pages or sheets, page, overlay, or data object with color fidelity.

color management resource. An object that provides [color management](#) in presentation environments.

color model. The method by which a color is specified. For example, the RGB color space specifies color in terms of three intensities for red (R), green (G), and blue (B). Also referred to as [color space](#).

color of medium. The color of a [presentation space](#) before any data is added to it. Synonymous with [reset color](#).

color space. The method by which a color is specified. For example, the RGB color space specifies color in terms of three intensities for red (R), green (G), and blue (B). Also referred to as [color model](#).

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](#).

command. (1) In [GOCA](#), a [data-stream construct](#) used to communicate from the [controlling environment](#) to the drawing process. The command introducer is environment dependent. (2) In the [IPDS](#) architecture, a [structured field](#) sent from a [host](#) to a printer. (3) A request for system action.

construct. An [architected](#) set of data such as a [structured field](#) or a [triplet](#).

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](#).

controlling environment. The environment in which an [object](#) is embedded, for example, the [IPDS](#) and [MO:DCA data streams](#).

coordinates. A pair of values that specify a position in a coordinate space. See also [absolute coordinate](#) and [relative coordinate](#).

coordinate system. A Cartesian coordinate system. An example is the [image coordinate system](#) that uses the fourth quadrant with positive values for the Y axis. The [origin](#) is the upper left-hand corner of the fourth quadrant. A pair of (x,y) values corresponds to one image point. Each image point is described by an image data element. See also [character coordinate system](#).

CPGID. See [Code Page Global Identifier](#).

current drawing attributes. The set of [attribute](#)s 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 position. The position identified by the current presentation space coordinates. For example, the coordinate position reached after the execution of a [drawing order](#). Contrast with [given position](#).

custom line type value. A user-defined [line type](#), defined by a series of pairs of a dash/dot length followed by a move length. Contrast with [standard line type value](#).

custom pattern. A user-defined [pattern](#), defined by the picture drawn by a series of [drawing orders](#) between a Begin Custom Pattern drawing order and an End Custom Pattern drawing order. Custom patterns can be either [bilevel custom patterns](#) or [full-color custom patterns](#). Contrast with patterns in the [default pattern set](#).

custom pattern mode. A mode that is entered when a Begin Custom Pattern drawing order is executed and exited when an End Custom Pattern drawing order is executed. While in this mode, drawing is done in a separate, temporary [graphics presentation space](#) rather than in the graphics presentation space of the current [GOCA](#) object.

cut-sheet media. Unconnected [sheets](#). Contrast with [continuous-form media](#).

D

data stream. A continuous stream of data that has a defined format. An example of a defined format is a [structured field](#).

DBCS. See [double-byte character set](#).

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. Synonymous with [drawing defaults](#).

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. Contrast with [current drawing controls](#).

default indicator. A field whose bits are all B'1' indicating that a hierarchical default value is to be used. The value can be specified by an external parameter. See also [external parameter](#).

default pattern set. A set of predefined [patterns](#), like solid, dots, or horizontal lines. Contrast with [custom pattern](#).

deprecated • encoding scheme

deprecated. An [architected construct](#) is marked as “deprecated” to indicate that it should no longer be used because it has been superseded by a newer construct. Use or support of a deprecated construct is permitted but no longer recommended. Constructs are deprecated rather than immediately removed to provide backward compatibility.

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 position](#)s of specific devices.

digital halftoning. A method used to simulate gray levels on a bilevel device.

digital image. An [image](#) whose [image data](#) was sampled at regular intervals to produce a digital representation of the image. The digital representation is usually restricted to a specified set of values.

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](#).

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 content architecture. A family of architectures that define the [syntax](#) and [semantics](#) of the document component.

document element. A self-identifying, variable-length, bounded record, that can have a content portion that provides control information, data, or both. An application or device does not have to understand control information or data to parse a data stream when all the records in the data stream are document elements. See also [structured field](#).

document fidelity. The degree to which a [document presentation](#) preserves the creator’s intent.

document formatting. A method used to determine where information is positioned in [presentation spaces](#) or on [physical media](#).

document presentation. A method used to produce a visible copy of formatted information on [physical media](#).

double-byte character set (DBCS). A [character set](#) that can contain up to 65,536 [characters](#).

double-byte coded font. A [coded font](#) in which the [code points](#) are two bytes long.

drawing control. A control that determines how a picture is drawn. Examples of drawing controls are [arc parameters](#), transforms, and the viewing window.

drawing defaults. In [GOCA](#), the set of attributes adopted at the start of each [segment](#) that is processed. These attributes are set either from standard defaults defined by the controlling environment or from the Set Current Defaults instruction that is contained in the Graphics Data Descriptor. Synonymous with [default drawing attributes](#). Contrast with [current drawing attributes](#).

drawing order. In [GOCA](#), a graphics [construct](#) that the [controlling environment](#) builds to instruct a [drawing processor](#) about what to draw and how to draw it. The order can specify, for example, that a [graphics primitive](#) be drawn, or a change to drawing [attributes](#) or [drawing controls](#) be effected. One or more graphics primitives can be used to draw a picture. Drawing orders can be included in a [structured field](#). Synonymous with [order](#).

drawing order coordinate space (DOCS). A two-dimensional conceptual space in which [graphics primitives](#) are drawn, using [drawing orders](#), to create pictures.

drawing process control. A control used by the [graphics processor](#) that determines how a picture is drawn. Examples of drawing process controls are arc parameters.

drawing processor. A [graphics processor](#) component that executes segments to draw a picture in a [presentation space](#). See also [segment](#) and [graphics presentation space](#).

duplex printing. 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](#). Contrast with [simplex printing](#).

E

EBCDIC. See [Extended Binary-Coded Decimal Interchange Code](#).

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](#).

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](#).

ESID. See [Encoding Scheme Identifier](#).

exception. An invalid or unsupported [data-stream](#) construct.

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](#).

Extended Binary-Coded Decimal Interchange Code (EBCDIC). A coded [character set](#) that consists of eight-bit coded [characters](#).

external parameter. A [parameter](#) for which the current value can be provided by the [controlling environment](#)—for example, the [data stream](#)—or by the [application](#) itself. Contrast with [internal parameter](#).

F

factoring. The movement of a parameter value from one state to a higher-level state. This permits the parameter value to apply to all of the lower-level states unless specifically overridden at the lower level.

FGID. See [Font Typeface Global Identifier](#).

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.

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 [character 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 character set. A [FOCA resource](#) containing descriptive information, [font metrics](#), and the digital representation of [character shapes](#) for a specified graphic character set.

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 local identifier. A binary identifier that is mapped by the [controlling environment](#) to a named [resource](#) to identify a [font](#). See also [local identifier](#).

font metrics. Measurement information that defines individual character values such as height, width, and space, as well as overall font values such as averages and maximums. Font metrics can be expressed in specific fixed units, such as [pels](#), or in relative units that are independent of both the [resolution](#) and the size of the [font](#). See also [character metrics](#) and [character set metrics](#).

font 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 referencing. A method used to identify or characterize a [font](#). Examples of processes that use font referencing are document editing, [document formatting](#), and [document presentation](#).

Font Typeface Global Identifier (FGID). A unique [font](#) identifier that can be expressed as either a two-byte binary or a five-digit decimal value. The FGID is used to identify a [type style](#) and the following characteristics: [posture](#), [weight class](#), and [width class](#).

font width (FW). (1) A characteristic value, parallel to the [character baseline](#), that represents the size of all [graphic](#)

foreground • GOCA

character is 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 with [object data](#). (2) In [GOCA](#), the portion of a [graphics primitive](#) that is mixed into the presentation space under the control of the current value of the [mix](#) and [color attributes](#). See also [pel](#). Contrast with [background](#).

foreground color. A [color attribute](#) used to specify the color of the [foreground](#) of a [primitive](#). Contrast with [background color](#).

foreground mix. An attribute used to determine how the [foreground color](#) of data is combined with the existing color of a [graphics presentation space](#). An example of data is a [graphics primitive](#). Contrast with [background mix](#).

form. Synonymous with [sheet](#).

format. The arrangement or layout of data on a [physical medium](#) or in a [presentation space](#).

formatter. A process used to prepare a [document](#) for presentation.

full arc. A complete circle or ellipse. See also [arc](#).

full-color custom pattern. A [custom pattern](#) that has its colors completely assigned during its definition, and can therefore contain any number of colors. Contrast with [bilevel custom pattern](#).

fully described font. In the [IPDS](#) architecture, an LF1-type raster-font resource containing font metrics, descriptive information, and the raster representation of character shapes, for a specific graphic character set. A fully described font can be downloaded to a printer using the Load Font Control and Load Font commands. An LF1-type coded font or coded-font section is the combination of one fully described font and one font index.

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](#).

GCUID. See [Graphic Character UCS 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). One of the following:

- [Coded Character Set Identifier \(CCSID\)](#)
- [Coded Graphic Character Set Global Identifier \(CGCSGID\)](#)
- [Code Page Global ID \(CPGID\)](#)
- [Font Typeface Global Identifier \(FGID\)](#)
- [Global Resource Identifier \(GRID\)](#)
- [Graphic Character Global Identifier \(GCGID\)](#)
- [Graphic Character Set Global Identifier \(GCSGID\)](#)
- [Graphic Character UCS Identifier \(GCUID\)](#)
- An identifier used by a data object to reference a [resource](#)
- In the [MO:DCA](#) environment, an encoded [graphic character](#) string that provides a reference name for a document element.

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](#).

GPS. See [graphics presentation space](#).

gradient. An area fill where one color gradually changes to another. A gradient is a type of [pattern](#).

graphic character. A member of a set of symbols that represent data. Graphic characters can be letters, digits, punctuation marks, or other symbols. Synonymous with [glyph](#). See also [character](#).

Graphic Character Global Identifier (GCGID). An alphanumeric [character string](#) used to identify a specific [graphic character](#). A GCGID can be from four bytes to eight bytes long.

graphic character identifier. The unique name for a [graphic character](#) in a [font](#) or in a [graphic character set](#). See also [character identifier](#).

Graphic Character Set Global Identifier (GCSGID). A unique [graphic character set](#) identifier that can be expressed as either a two-byte binary or a five-digit decimal value.

Graphic Character UCS Identifier (GCUID). An alphanumeric character string used to identify a specific [graphic character](#). The GCUID naming scheme is used for additional characters and sets of characters that exist in UNICODE; each GCUID begins with the letter *U* and ends with a UNICODE code point. The Unicode Standard is fully compatible with the earlier Universal Character Set (UCS) Standard.

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 object. An object that contains [graphics data](#). See also [object](#).

graphics object area. A rectangular area on a [logical page](#) into which a [graphics presentation space window](#) is mapped.

Graphics Object Content Architecture (GOCA). An architected collection of [constructs](#) used to [interchange](#) and present [graphics data](#). GOCA was originally defined by IBM; this architecture is no longer used in [AFP](#). Instead, a subset of GOCA was defined for use in AFP environments, called [Graphics Object Content Architecture for Advanced Function Presentation \(AFP GOCA\)](#). Usually when the term "GOCA" is used in AFP documentation, it means AFP GOCA.

Graphics Object Content Architecture for Advanced Function Presentation (AFP GOCA). A subset of the GOCA architecture, originally defined by IBM, specifically designed for [AFP](#) environments. See [Graphics Object Content Architecture \(GOCA\)](#).

graphics presentation space (GPS). A two-dimensional conceptual space in which the application user's view of the specified picture is generated. The picture can then be mapped onto an output [medium](#).

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](#).

H

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.

highlight color. A spot color that is used to accentuate or contrast monochromatic areas. See also [spot color](#).

highlighting. The emphasis of displayed or printed information. Examples are increased intensity of selected characters on a display screen and [exception](#) highlighting on an [IPDS](#) printer.

hollow font. A font design in which the graphic character shapes include only the outer edges of the strokes.

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.

horizontal scale factor • kerning

- 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 fonts](#) and [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. 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](#).

host. In the [IPDS](#) architecture, a computer that drives a printer.

I

ID. Identifier.

I direction. Synonymous with [inline direction](#).

IEEE. Institute of Electrical and Electronics Engineers.

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 object. An object that contains [image data](#). See also [object](#).

Image Object Content Architecture (IOCA). An architected collection of [constructs](#) used to [interchange](#) and present [images](#).

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](#).

immediate mode. The mode in which [segments](#) are executed as they are received and then discarded.

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](#).

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](#).

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.

IOCA. See [Image Object Content Architecture](#).

IO Image. An image object containing [IOCA constructs](#). Contrast with [IM Image](#).

IPDS. See [Intelligent Printer Data Stream](#).

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.

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.

keyword. A two-part self-defining parameter consisting of a one-byte identifier and a one-byte value.

L

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.

LID. See [local identifier](#).

ligature. A single [glyph](#) representing two or more [characters](#). Examples of characters that can be presented as ligatures are *ff* and *ffi*.

linear gradient. A [gradient](#) where the color change takes place along a line. Contrast with [radial gradient](#).

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. The line type can either be a [standard line type value](#) or a [custom line type value](#). Contrast with [line width](#).

line width. A [line attribute](#) that controls the appearance of a line. Contrast with [line type](#).

local area network (LAN). A data network located on a user's premises in which serial transmission is used for direct data communication among data stations.

Local Character Set Identifier (LCID). A [local identifier](#) used as a [character](#), [marker](#), or [pattern set](#) attribute.

local identifier (LID). An identifier that is mapped by the [controlling environment](#) to a named [resource](#).

location. A site within a [data stream](#). A location is specified in terms of an offset in the number of [structured fields](#) from the beginning of a data stream, or in the number of bytes from another location within the data stream.

logical page. A [presentation space](#). One or more [object areas](#) can be mapped to a logical page. A logical page has specifiable characteristics, such as size, shape, [orientation](#), and offset. The shape of a logical page is the shape of a rectangle. Orientation and offset are specified relative to a [medium coordinate system](#).

logical unit. A unit of linear measurement expressed with a unit base and units per unit-base value. For example, in [MO:DCA](#) and [IPDS](#) architectures, the following logical units are used:

- 1 logical unit = 1/1440 inch (unit base = 10 inches, units per unit base = 14,400)
- 1 logical unit = 1/240 inch (unit base = 10 inches, units per unit base = 2400)

Synonymous with [L-unit](#).

look-up table (LUT). (1) A table used to map one or more input values to one or more output values. (2) A logical list of colors or intensities. The list has a name and can be referenced to select a color or intensity. See also [color table](#).

lowercase. Pertaining to small letters as distinguished from capital letters. Examples of small letters are *a*, *b*, and *g*. Contrast with [uppercase](#).

L-unit. Synonymous with [logical unit](#).

LUT. See [look-up table](#).

M

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 cell-size and color.

marker cell. A conceptual rectangular box that can include a [marker symbol](#) and the space surrounding that symbol.

marker precision. A method used to specify the degree of influence that [marker attributes](#) have on the appearance of a [marker](#); this method has been made [obsolete](#).

marker set. In [GOCA](#), a set of graphic symbols used to indicate a position.

marker symbol. A symbol that is used for a [marker](#).

meaning • offset

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](#).

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 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](#).

metadata. Descriptive information that is associated with and augments other data.

Metadata Object Content Architecture (MOCA). A resource object architecture to carry metadata that serves to provide context or additional information about an [AFP](#) object or other AFP data.

mil. 1/1000 inch.

mix. A method used to determine how the color of a [graphics primitive](#) is combined with the existing color of a [graphics presentation space](#). See also [foreground mix](#) and [background mix](#).

Mixed Object Document Content Architecture (MO:DCA). An [architected](#), device-independent [data stream](#) for [interchanging documents](#).

MOCA. See [Metadata Object Content Architecture](#).

MO:DCA. See [Mixed Object Document Content Architecture](#).

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

name. A table heading for architecture [syntax](#). The entries under this heading are short names that give a general indication of the contents of the [construct](#).

named color. A color that is specified with a descriptive name. An example of a named color is “green”.

neutral white. A [color attribute](#) that gives a device-dependent [default](#) color, typically white on a screen and black on a printer. Note that neutral white and color of medium are two different colors.

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](#).

N-up. The partitioning of a [side](#) of a [sheet](#) into a fixed number of equal size partitions. For example, 4-up divides each side of a sheet into four equal partitions.

O

object. (1) A collection of [structured fields](#). The first structured field provides a begin-object function, and the last structured field provides an end-object function. The object can contain one or more other structured fields whose content consists of one or more data elements of a particular data type. An object can be assigned a name that can be used to reference the object. Examples of objects are [presentation text](#), [font](#), [graphics](#), and [image](#) objects. (2) Something that a user works with to perform a task.

object area. A rectangular area in a [presentation space](#) into which a data [object](#) is mapped. The presentation space can be for a [page](#) or an [overlay](#). Examples are a graphics object area, an image object area, and a bar code object area.

object data. A collection of related data elements bundled together. Examples of object data include [graphic characters](#), image data elements, and [drawing orders](#).

obsolete. Removed from the architecture, and thus ignored by receivers.

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.

online. A device state in which the device is under the direct control of a [host](#). Contrast with [offline](#).

order. Synonymous with [drawing order](#).

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 [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.

overlay. (1) A [resource object](#) that can contain 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. (2) The final representation of such an object on a [physical medium](#). Contrast with [page segment](#).

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 such an object on a [physical medium](#).

page segment. (1) In the [IPDS](#) architecture, a [resource object](#) that can contain [text](#), [image](#), [graphics](#), and bar code data. Page segments do not define their own environment, but are processed in the existing environment. (2) In the [MO:DCA](#) architecture, a resource object that can contain any mixture of bar code objects, graphics objects, and [IOCA](#) image objects. A page segment does not contain an active environment group. The environment for a page segment is defined by the active environment group of the including page or overlay. (3) The final representation of such an object on a [physical medium](#). Contrast with [overlay](#).

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.

pattern. A graphic symbol used repeatedly to fill an [area](#).

pattern attributes. The characteristics that specify the appearance of a [pattern](#).

pattern reference point. A position in the [graphics presentation space](#) to be used as the origin of a [custom pattern](#); the pattern is tiled in all directions from this position.

pattern set. In [GOCA](#), a set of graphic symbols used to fill the interior of an area.

pattern symbol. A graphic symbol that is used for a [pattern](#).

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](#).

picture element. Synonymous with [pel](#).

pixel. Synonymous with [pel](#).

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

polyline • reserved

within the drawing order coordinate space. See also [drawing order coordinate space](#).

polyline. A sequence of connected lines.

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](#).

posture. Inclination of a letter with respect to a vertical axis. Examples of inclination are upright and inclined.

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](#), [logical page](#), and [medium presentation space](#).

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.

process color. A color that is specified as a combination of the components, or primaries, of a color space. A process color is rendered by mixing the specified amounts of the primaries. An example of a process color is C=0.1, M=0.8, Y=0.2, K=0.1 in the cyan/magenta/yellow/black (CMYK) color space. Contrast with [spot color](#).

prolog. The first portion of a [segment](#)'s data. Prologs are optional. They contain [attribute](#) settings and [drawing controls](#). Synonymous with [segment prolog](#).

proportionally spaced font. A [font](#) with [graphic characters](#) that have varying [character increments](#). Proportional spacing can be used to provide the appearance of even spacing between presented characters and to eliminate excess blank space around narrow characters. An example of a narrow character is the letter *i*. Synonymous with [typographic font](#). Contrast with [monospaced font](#) and [uniformly spaced font](#).

Proportional Spacing Machine font (PSM font). A [font](#) originating with the electric typewriter and having character increment values that are integer multiples of the narrowest character width.

PSM font. See [Proportional Spacing Machine font](#).

PTOCA. See [Presentation Text Object Content Architecture](#).

R

radial gradient. A [gradient](#) where the color change takes place between two full arcs. Contrast with [linear gradient](#).

range. A table heading for architecture [syntax](#). The entries under this heading give numeric ranges applicable to a [construct](#). The ranges can be expressed in binary, decimal, or [hexadecimal](#). The range can consist of a single value.

raster pattern. A rectangular array of [pels](#) arranged in [rows](#) called [scan lines](#).

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 positioning. The establishment of a position within a [coordinate system](#) as an offset from the [current position](#). Contrast with [absolute positioning](#).

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](#).

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.

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 the [MO:DCA](#) architecture, resource objects can be contained within a resource group. Examples of resources are [fonts](#), [overlays](#), and [page segments](#).

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

RGB. Red, green and blue, the additive primary colors.

RGB color space. The basic additive [color model](#) used for color video display, as on a computer monitor.

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.

rule. A solid line of any [line width](#).

S

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. The ratio of an image-object-area size to its image-presentation-space size.

scan line. A series of picture elements. Scan lines in raster patterns form [images](#). See also [picture element](#) and [raster pattern](#).

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](#).

segment. (1) In [GOCA](#), a set of graphics [drawing orders](#) contained within a Begin Segment [command](#). See also [graphics segment](#). (2) In [IOCA](#), [image content](#) bracketed by Begin Segment and End Segment self-defining fields. See also [image segment](#).

segment chain. A string of [segments](#) that defines a picture.

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 and segment length.

semantics • triplet

semantics. The meaning of the [parameters](#) of a [construct](#). See also [pragmatics](#) and [syntax](#).

shear. The angle of slant of a character cell that is not perpendicular to a baseline. Synonymous with [character shear](#).

sheet. A division of the [physical medium](#); multiple sheets can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a sheet. Envelopes are an example of a physical medium that comprises only one sheet. The [IPDS](#) architecture defines four types of sheets: [cut-sheet media](#), [continuous-form media](#), envelopes, and computer output on microfilm. Each type of sheet has a top edge. A sheet has two [sides](#), a front side and a back side. Synonymous with [form](#).

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.

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.

spot color. A color that is specified with a unique identifier such as a number. A spot color is normally rendered with a custom colorant instead of with a combination of process color primaries. See also [highlight color](#). Contrast with [process color](#).

standard action. The architecture-defined action to be taken on detecting an [exception condition](#), when the [controlling environment](#) specifies that processing should continue.

standard line type value. A predefined [line type](#), like [solid](#), [invisible](#), or [dash-dot](#). Contrast with [custom line type value](#).

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 the [MO:DCA](#) architecture, 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](#).

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.

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](#).

syntax. The rules governing the structure of a [construct](#). See also [pragmatics](#) and [semantics](#).

T

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 orientation. A description of the appearance of text as a combination of inline direction and baseline direction. See also [baseline direction](#), [inline direction](#), and [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](#).

toned. Containing marking agents such as toner or ink. Contrast with [untoned](#).

trimming. Eliminating those parts of a picture that are outside of a clipping boundary such as a viewing window or [presentation space](#). Synonymous with [clipping](#).

triplet. A three-part self-defining variable-length parameter consisting of a length byte, an identifier byte, and parameter-value bytes.

triplet identifier. A one-byte type identifier for a [triplet](#).

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.

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](#), and [UNDF](#).

typeface. All [character](#)s of a single [type family](#) or style, [weight class](#), [width class](#), and [posture](#), regardless of size. For example, Helvetica Bold Condensed [Italics](#), in any point size.

type family. All [character](#)s of a single design, regardless of [attributes](#) such as width, weight, [posture](#), and size. Examples are Courier and Gothic.

type structure. Attributes of [character](#)s other than [type family](#) or [typeface](#). Examples are solid shape, hollow shape, and overstruck.

type style. The form of [character](#)s within the same [font](#), for example, Courier or Gothic.

type weight. A parameter indicating the degree of boldness of a [typeface](#). A [character](#)'s [stroke](#) thickness determines its type weight. Examples are light, medium, and bold. Synonymous with [weight class](#).

type width. A parameter indicating a relative change from the [font](#)'s normal width-to-height ratio. Examples are normal, condensed, and expanded. Synonymous with [width class](#).

typographic font. A [font](#) with [graphic character](#)s that have varying [character increment](#)s. 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](#).

underscore. A method used to create an underline beneath the [character](#)s in a specified text field. An example of underscore is the line presented under one or more

characters. Also a special [graphic character](#) used to implement the underscoring function.

UNDF. A data type for architecture [syntax](#), indicating one or more bytes that are undefined by the architecture.

Unicode. A character encoding standard for information processing that includes all major scripts of the world. Unicode defines a consistent way of encoding multilingual [text](#). Unicode specifies a numeric value, a name, and other attributes—such as directionality—for each of its characters; for example, the name for \$ is “dollar sign” and its numeric value is X'0024'. This Unicode value is called a Unicode [code point](#) and is represented as U+*nnnn*. Unicode provides for three encoding forms (UTF-8, UTF-16, and UTF-32), described as follows:

UTF-8 A byte-oriented form that is designed for ease of use in traditional ASCII environments. Each UTF-8 code point contains from one to four bytes. All Unicode code points can be encoded in UTF-8 and all 7-bit ASCII characters can be encoded in one byte.

UTF-16 The default Unicode encoding. A fixed, two-byte Unicode encoding form that can contain surrogates and identifies the byte order of each UTF-16 code point via a Byte Order Mark in the first 2 bytes of the data. Surrogates are pairs of Unicode code points that allow for the encoding of as many as 1 million additional characters without any use of escape codes.

UTF-16BE UTF-16 that uses big endian byte order; this is the byte order for all multi-byte data within AFP data streams. The Byte Order Mark is not necessary when the data is externally identified as UTF-16BE (or UTF-16LE).

UTF-16LE UTF-16 that uses little endian byte order.

UTF-32 A fixed, four-byte Unicode encoding form in which each UTF-32 code point is precisely identical to the Unicode code point.

UTF-32BE UTF-32 serialized as bytes in most-significant-byte-first order (big endian). UTF-32BE is structurally the same as UCS-4.

UTF-32LE UTF-32 serialized as bytes in least-significant-byte-first order (little endian).

uniformly spaced font. A [font](#) with [graphic character](#)s 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 • X_{pg}, Y_{pg} coordinate system

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.

untoned. Unmarked portion of a [physical medium](#). Contrast with [toned](#).

uppercase. Pertaining to capital letters. Examples of capital letters are A, B, and C. Contrast with [lowercase](#).

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](#).

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](#), and [user printable area](#).

vector graphics. A vector has a defined starting point, a designated direction, and a specified distance. Vector graphics is line-based [graphics data](#), where vectors determine how straight and curved lines are shaped between specific points. A picture consists of lines and colors to fill the areas enclosed by the lines.

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](#).

VPA. See [valid printable area](#).

W

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](#).

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](#).

X

X_g, Y_g coordinate system. The [graphics presentation space](#) (GPS) [coordinate system](#).

X_m, Y_m coordinate system. (1) In the [IPDS](#) architecture, the [medium presentation space coordinate system](#). (2) In the [MO:DCA](#) architecture, the [medium](#) coordinate system.

X_{oa}, Y_{oa} coordinate system. The [object area coordinate system](#).

X_{oi}, Y_{oi} coordinate system. The [overlay coordinate system](#).

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](#).

Index

A

AFPC iii, vii, 4
arc parameters 23, 130
arc primitives 24
areas 36
areas primitive 36
 valid definitions 38
attributes 12
 character 13, 56
 drawing 12
 image 59
 line 12, 29
 marker 13, 58
 obsolete 195
 pattern 13, 50
attributes, current 70
attributes, primitives
 color 14
 color tables 14
 line end 32–33
 line join 32, 34
 line type 29
 line width 32
 mix 16
 mix and background mix values 17

B

background mix values 17
Begin Segment command 75
box primitive 28
boxes 28

C

chained segments 61
character attributes 13, 56
character precision
 character strings 53, 138
character precision attribute 51, 138
character strings 51
 device-specific (character)
 precision 53
 device-specific (string) precision 56
coexistence functions 195
color attribute 14
color table 14
commands 11, 199
 Begin Segment 75
communication with the controlling
 environment 65
 control instructions 65
compliance 175
computer graphics 7
concepts of GOCA 7
control instructions 11, 65, 199
 coordinate data 79
 drawing orders 79
 offset data 79
coordinate space definitions
 Drawing Order Coordinate Space
 (DOCS) 13

 Graphics Presentation Space
 (GPS) 13
 Usable Area (UA) 14
 cross-references 199
 cubic bezier curve primitive 27
 current attributes 70
 current defaults 79
 current position 19
 in drawing orders 79
 curved lines 23
 Cubic Bezier Curve primitive 27
 Fillet primitive 26
 Full Arc primitive 24
 Partial Arc primitive 25
 custom patterns 41–42
 bilevel 42
 full-color 42

D

default pattern set 39
default values, drawing orders 79
device-specific (character) precision 53, 138
device-specific (string) precision 56, 138
DR/2V0 subset 17, 176
drawing attributes 12
drawing control instructions
 Set Current Defaults 66
drawing defaults 70, 79
drawing order alignment 79
Drawing Order Coordinate Space
 (DOCS) 13
Drawing Order Subset parameter
 (retired) 181, 197
drawing orders 11, 61, 73, 200, 202
 Begin Area (GBAR) order 82
 Begin Custom Pattern (GBCP)
 order 84
 Begin Image (GBIMG, GCBIMG)
 orders 87
 Box (GBOX, GCBOX) orders 90
 Character String (GCHST, GCCHST)
 orders 92
 Comment (GCOMT) order 95
 coordinate data 79
 offset data 79
 Cubic Bezier Curve (GCBEZ, GCCBEZ)
 orders 96
 current position 79
 default values 79
 definition of 12
 Delete Pattern (GDPT) order 99
 End Area (GEAR) order 101
 End Custom Pattern (GECP)
 order 102
 End Image (GEIMG) order 103
 End Prolog (GEPROL) order 104
 example of syntax v
 exception conditions 168
 Fillet (GFLT, GCFLT) orders 105
 formats
 extended 78
 fixed 1-byte 78
 fixed 2-byte 78

 long 78
 order alignment 79
Full Arc (GFARC, GCFARC)
 orders 107
Image Data (GIMD) order 109
Line (GLINE, GCLINE) orders 110
Linear Gradient (GLGD) order 112
Marker (GMRK, GCMRK) orders 116
No-Operation (GNOP1) order 118
obsolete 196
Partial Arc (GPARC, GCPARC)
 orders 119
Radial Gradient (GRGD) order 122
Relative Line (GRLINE, GCRLINE)
 orders 127
Segment Characteristics (GSGCH)
 order 129
Set Arc Parameters (GSAP) 130
Set Background Mix (GSBMX)
 order 132
Set Character Angle (GSCA)
 order 133
Set Character Cell (GSCC) order 134
Set Character Direction (GSCD)
 order 136
Set Character Precision (GSCR)
 order 138
Set Character Set (GSCS) order 140
Set Character Shear (GSCH)
 order 141
Set Color (GSCOL) order 142
Set Current Position (GSCP)
 order 143
Set Custom Line Type (GSCLT)
 order 144
Set Extended Color (GSECOL)
 order 146
Set Fractional Line Width (GSFLW)
 order 147
Set Line End (GSLE) order 148
Set Line Join (GSLJ) order 149
Set Line Type (GSLT) order 150
Set Line Width (GSLW) order 151
Set Marker Cell (GSMC) order 152
Set Marker Precision (GSMP) order
 (obsolete) 196
Set Marker Set (GSMS) order 154
Set Marker Symbol (GSMT) order 155
Set Mix (GSMX) order 156
Set Pattern Reference Point (GSPRP)
 order 157
Set Pattern Set (GSPS) order 158
Set Pattern Symbol (GSPT) order 159
Set Process Color (GSPCOL)
 order 161
drawing process check 168
drawing process controls 70
 current position 19
drawing processing environment 61
drawing processor 11
drawing processor facilities
 drawing process controls 70

E

environment interface	11
exception conditions	17, 19, 168
for Begin Area (GBAR) order	83
for Begin Custom Pattern (GBCP) order	85
for Begin Image (GBIMG, GCBIMG) orders	89
for Box (GBOX, GCBOX) orders	91
for Character String (GCHST, GCCHST) orders	93
for Comment (GCOMT) order	95
for Cubic Bezier Curve (GCBEZ, GCCBEZ) orders	98
for Delete Pattern (GDPT) order	99
for End Area (GEAR) order	101
for End Custom Pattern (GECP) order	102
for End Image (GEIMG) order	103
for End Prolog (GEPROL) order	104
for Fillet (GFLT, GCFLT) orders	106
for Full Arc (GFARC, GCFARC) orders	108
for Image Data (GIMD) order	109
for Line (GLINE, GCLINE) orders	111
for Linear Gradient (GLGD) order	115
for Marker (GMRK, GCMRK) orders	117
for No-Operation (GNOP1) order	118
for Partial Arc (GPARC, GPCARC) orders	120
for Radial Gradient (GRGD) order	126
for Relative Line (GRLINE, GCRLINE) orders	128
for Segment Characteristics (GSGCH) order	129
for Set Arc Parameters (GSAP) order	131
for Set Background Mix (GSBMX) order	132
for Set Character Angle (GSCA) order	133
for Set Character Cell (GSCC) order	135
for Set Character Direction (GSCD) order	137
for Set Character Precision (GSCR) order	139
for Set Character Set (GSCS) order	140
for Set Character Shear (GSCH) order	141
for Set Color (GSCOL) order	142
for Set Current Position (GSCP) order	143
for Set Custom Line Type (GSCLT) order	145
for Set Extended Color (GSECOL) order	146
for Set Fractional Line Width (GSFLW) order	147
for Set Line End (GSLE) order	148
for Set Line Join (GSLJ) order	149
for Set Line Type (GSLT) order	150
for Set Line Width (GSLW) order	151
for Set Marker Cell (GSMC) order	153
for Set Marker Precision (GSMP) order (obsolete)	196
for Set Marker Set (GSMS) order	154

for Set Marker Symbol (GSMT) order	155
for Set Mix (GSMX) order	156
for Set Pattern Reference Point (GSPRP) order	157
for Set Pattern Set (GSPS) order	158
for Set Pattern Symbol (GSPT) order	160
for Set Process Color (GSPCOL) order	164
IPDS	192
obsolete	197
retired	197
summary of	167
command process checks	167
exceptions with standard actions	171
exceptions without standard actions	169
instruction process checks	167
extended order format	78

F

facilities for drawing processor	70
drawing process controls	70
facilities for graphics processor attributes	70
fillet primitive	26
fixed 1-byte format	78
fixed 2-byte format	78
foreground mix values	17
full arc primitive	24
fully described fonts	21

G

GOCA overview	9
gradients	43
linear	44
radial	46
graphics	7
graphics commands Begin Segment	75
graphics coordinate spaces	13
graphics in IPDS environment	189
Graphics Presentation Space (GPS)	13
graphics process facilities attributes	70
graphics processor model	9
GRS2 subset	17, 175–176
GRS3 subset	17, 178

I

image attributes	59
image primitive image definition	59
image resolution	88, 182
images	59
immediate mode	61
introduction to GOCA	7
IPDS environment	189
IPDS exceptions	192
IPDS Graphics Command Set Write Graphics	191

Write Graphics Control Graphics Area Position	190
Graphics Data Descriptor	191
Graphics Output Control	190

L

levels of subset Base (mandatory), Version 0	175
drawing order levels, Version 0	176, 178
Level 2	176
Level 3 (GRS3)	178
line attributes	12, 29
line end attribute	32–33
line join attribute	32, 34
line primitive	22
line type attribute	29
line width attribute	32
lines boxes Box primitive	28
curved	23, 26
Cubic Bezier Curve primitive	27
Fillet primitive	26
Full Arc primitive	24
Partial Arc primitive	25
straight	22
Line primitive	22
Relative Line primitive	22
long format	78

M

marker attributes	13, 58
markers	57
precision	195
migration and coexistence support	195
migration functions	195
mix and background mix attribute values	17
mix attribute	16
MO:DCA environment	179
modes immediate	61
Move Type orders	30

N

notation used in formulas	vi
notices	205

O

obsolete attributes	195
obsolete drawing orders	196
obsolete exceptions	197
obsolete functions	195
orders, summary of	80
output control definitions GPS window	190
graphics object areas	190
mapping control options	190
mapping defaults	190

output primitives	19
areas	36
valid definitions	38
attributes	12
boxes	
Box	28
curved lines	23
Cubic Bezier Curve	27
Fillet	26
Full Arc	24
Partial Arc	25
images	59
overflow of	60
straight lines	22
Line	22
Relative Line	22
symbol	20
character strings	51
markers	57
patterns	39
overflow of output primitives	60

P

parameter type	71
partial arc primitive	25
pattern attributes	13, 50
patterns	39
custom	41–42
bilevel	42
full-color	42
default pattern set	39
gradients	43
linear	44
radial	46
precision value	
character	
character strings	53, 138
marker	196
string	
character strings	56, 138
stroke	138
presentation spaces	13
primitive attributes	12
primitives	19
process controls	
parameter type	71
processing of segments	
current attributes	70
segment prolog	62
processor facilities	70
prolog segment semantics	63
properties	
segments	62

R

raster symbol sets	21
receiving data	61
relative line primitive	22
reserved fields	v
retired exceptions	197
retired functions	195, 197
retired parameters	197

S

scope of GOCA	7
segment chain	61
segment commands	
Begin Segment	75
segment processing sequence	61
segment prolog	62
segment types	61
segments	17, 61
appended	61
chained	61
current attributes	70
properties	62
segment prolog	62
sequence of segment processing	61
Set Current Defaults instruction	66
standard actions	171
standard defaults	70, 79
start and sweep angles	25
straight lines	22
Line primitive	22
Relative Line primitive	22
string precision	
character strings	56, 138
stroke precision	
character strings	138
subset levels	
Base (mandatory) Version 0	175
drawing order levels, Version 0	176, 178
Level 2	176
Level 3 (GRS3)	178
within IPDS	189
within the MO:DCA environment	188
subsetting	17
summary of exception conditions	167
those with standard actions	171
those without standard actions	169
symbol primitive	20
character strings	51
device-specific (character)	
precision	53
device-specific (string) precision	56
markers	57
patterns	39
symbol sets	
fonts	51
raster	21
syntax diagrams	
how to read	v

T

trademarks	206
------------	-----

U

Usable Area (UA)	14
------------------	----

V

valid definitions of area primitive	38
Version 0 subset levels	
Base (mandatory)	175
drawing order levels	176, 178
Level 2	176

Level 3 (GRS3)	178
----------------	-----

W

Write Graphics Control	189
------------------------	-----

Advanced Function Presentation Consortium

Graphics Object Content Architecture for Advanced Function Presentation Reference

AFPC-0008-03

