Advanced Function Presentation Consortium
Data Stream and Object Architectures

Image Object Content Architecture Reference

AFPC-0003-08

AFPConsortium™
Advanced Function Presentation

Note

Before using this information, read the information in "Notices" on page 181.

AFPC-0003–08
Eighth Edition (May 2022)

This edition applies to the Image Object Content Architecture. It is the second edition produced by the AFP Consortium™ (AFPC™) and replaces and makes obsolete the previous edition (AFPC-0003–07). 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 "Changes in This Edition" on page ix.

Internet

Visit our home page, where this publication and others are available:
`http://www.afpconsortium.org`

Copyright © AFP Consortium 2010, 2022

Preface

This book describes the functions and services associated with Image Object Content Architecture (IOCA). It 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 a product or program to interoperate with other presentation products in an Advanced Function Presentation™ 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™ (AFPCC™). The goal was to extend the object architectures with support for full-color devices including support for comprehensive color management. The idea of doing this via a consortium consisting of the primary AFP architecture users was to build synergism with partners from across the relevant industries, such as hardware manufacturers that produce printers as well as software vendors of composition, work flow, viewer, and transform tools. Quickly more than 30 members came together in regular meetings and work group sessions to create the AFP Color Management Object Content Architecture™ (CMOCA™). A major milestone was reached by the AFP Color Consortium with the initial official release of the CMOCA specification in May 2006.

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

How to Use This Book

This book contains the following sections:

- Chapter 1, "A Presentation Architecture Perspective", provides a brief overview of Presentation Architecture.
- Chapter 2, "Introduction to IOCA", discusses the background of image processing and introduces IOCA.
- Chapter 3, "IOCA Overview", discusses concepts involved in image processing.
- Chapter 4, "Formats and Codes", shows formats used by IOCA, and code points assigned to and reserved for IOCA.
- Chapter 5, "IOCA Image Segment", describes the components of the IOCA entity.

- Chapter 6, "Exception Conditions and Actions", lists exceptions to the IOCA definitions, and standard actions to take when exceptions occur.
- Chapter 7, "Compliance", describes the function sets that IOCA defines.
- Appendix A, "Compression and Recording Algorithms", discusses compression and recording algorithms that IOCA supports.
- Appendix B, "Bilevel, Grayscale, and Color Images", summarizes how to specify these different types of images.
- Appendix C, "IOCA Tile Resource", describes the structure and use of tile resources.
- Appendix D, "MO:DCA Environment", describes how the IOCA Image Segments are carried in the MO:DCA™ data stream controlling environment.
- Appendix E, "IPDS Environment", describes how the IOCA Image Segments are carried in the IPDS™ architecture controlling environment.
- Appendix F, "Notes for IOCA Generators", discusses issues that should be considered when generating efficient IOCA for high speed printing.
- Appendix G, "Retired Architecture", describes parts of IOCA that have been retired.
- Glossary defines terms used in this book.

How to Read the Syntax Diagrams

Throughout this book, syntax for IOCA is shown in tables, laid out as follows:

Offset	Type	Name	Range	Meaning	M/O
Byte offset	Data type	Name, if applicable	Range of valid values, if applicable	Meaning or purpose of the parameter	M or O
Bit offset					

The "M/O" column indicates whether the parameter is mandatory or optional.

The syntax includes the following basic data types:

BITS Bit string
CHAR Character string
CODE Architected constant
UBIN Unsigned binary

The following is an example of IOCA syntax.

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'9B'	IDE Structure parameter	M
1	UBIN	LENGTH	X'06' – X'14'	Length of the parameters to follow	M
2	BITS	FLAGS			M
Bit 0		ASFLAG	B'0' – B'1'	Additive or Subtractive: **B'0'** Additive **B'1'** Subtractive	
Bit 1		GRAYCODE	B'0' – B'1'	Gray coding: **B'0'** Off **B'1'** On	
Bits 2–7			B'000000'	Reserved; should be zero	
3	CODE	FORMAT	X'01', X'02', X'04', X'12', X'8*n*'	Color model: **X'01'** RGB **X'02'** YCrCb **X'04'** CMYK **X'12'** YCbCr **X'8*n*'** nColor (X'2' ≤ *n* ≤ X'F') All other values are reserved.	M
4–6			X'000000'	Reserved; should be zero	M
7	UBIN	SIZE1	X'00' – X'FF'	Number of bits/IDE for component 1	M
8	UBIN	SIZE2	X'00' – X'FF'	Number of bits/IDE for component 2	O
9	UBIN	SIZE3	X'00' – X'FF'	Number of bits/IDE for component 3	O
10	UBIN	SIZE4	X'00' – X'FF'	Number of bits/IDE for component 4	O
11	UBIN	SIZE5	X'00' – X'FF'	Number of bits/IDE for component 5	O
...					
21	UBIN	SIZE15	X'00' – X'FF'	Number of bits/IDE for component 15	O

Notation Conventions

Throughout this document, the following notation conventions apply:

- Bytes are numbered from left to right beginning with byte 0, which is considered the high order byte position. For example, a three-byte field consists of byte 0, byte 1, and byte 2.
- Each byte is composed of eight bits.
- Bits in a single byte are numbered from left to right beginning with bit 0, the most significant bit, and continuing through bit 7, the least significant bit.
- When bits from multiple consecutive bytes are considered together, the first byte, byte 0, contains bits 0 to 7, and byte n contains bits $n×8$ to $n×8+7$.
- A negative number is expressed by the two's-complement form of its positive number. The two's complement of a number is obtained by first inverting every bit of the number and then adding one to the inverted number.

In the syntax summary diagrams, the conventions in the parameter groupings are:

- The identifier is shown for all the parameters. If the identifier is missing, the item is not a parameter, but a grouping of parameters, for example, a tile.
- The following symbols have special meanings:

  [ ] Brackets indicate optional parameters. When a parameter is shown without brackets, it *must* appear if the corresponding grouping is present. For example, if a tile is being specified, Tile Position must appear.

  + Plus signs indicate that a group of successive parameters may appear in any order relative to each other.

  (S) The enclosed (S) indicates that the parameter may be repeated. When it is present on a required parameter, at least one instance of the parameter is required, but multiple instances of it may occur.

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, `http://www.afpconsortium.org`.

Table 1. AFP Consortium Architecture References

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

Table 2. Additional AFP Consortium Documentation

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

Further Reading

The following publications describe image compression algorithms:

- Abramson, Norman. *Information Theory and Coding*. New York: McGraw-Hill, 1963.
- Arps, R., T. Truong, D. Lu, R. Pasco, and T. Friedman, "A multipurpose VLSI chip for adaptive data compression of bilevel images". *IBM Journal of Research and Development*, Volume 32, No. 6 (November 1988).
- "Binary-image-manipulation Algorithms in Image View Facility". *IBM Journal of Research and Development*, vol. 31, no. 1 (January 1987).
- International Organization for Standardization and International Electrotechnical Commission. ISO/IEC International Standard 10918-1. 1994.

- *Composed Page Data Stream Architecture IS & TG Architecture Memorandum*. AR-7262-03-POK. Poughkeepsie, NY: IBM.
- International Telecommunications Union-Telecommunication Standardization Sector. *Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus*. Terminal Equipment and Protocols for Telematic Services Recommendations of the T Series, Recommendation T.6. ITU-TSS Volume VII, Fascicle VII.3.
- ________. *Standardization of Group 3 Facsimile Apparatus for Document Transmission*. Terminal Equipment and Protocols for Telematic Services Recommendations of the T Series, Recommendation T.4. ITU-TSS Volume VII, Fascicle VII.3.
- ________. Terminal Equipment and Protocols for Telematic Services Recommendations of the T Series, Recommendation T.81. 1993.
- Pennebaker, William B., and Joan L. Mitchell. *JPEG: Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1992. ISBN 0-442-01272-01.
- ________. “Standardization of Color Image Data Compression”. Part I. “Sequential Coding”. *Proceedings Electronic Imaging '89 East* (October 2–5, 1989): 109–112.
- *TIFF*. Revision 6.0, Final. Aldus Corp.: June 3, 1992.
- Welch, Terry A. “A Technique for High Performance Data Compression”. *IEEE Computer*, vol. 17, no. 6 (June 1984).

The following publications describe color and grayscale images:

- Commission Internationale de l'Eclairage. *Colorimetry*. CIE Publication no. 15-2.
- Hunt, R. *The Representation of Colour in Photography, Printing and Television* 5th ed. Foundation Press, 1995.
- Lucky, R. W., J. Salz, and E. J. Weldon Jr.. *Principles of Data Communication* (New York: McGraw-Hill, 1968).

Changes in This Edition

The following is a summary of the changes that have been made in this edition:

- The nColor color model was added to the IDE Structure parameter
  - Exception conditions EC-9B10 and EC-9B18 were updated accordingly
- The FS48 function set was added; the main additions to FS48 compared to FS45 are:
  - The nColor color model
  - The TIFF LZW with Differencing Predictor compression algorithm
  - RELRES=1 is allowed for JPEG compressed data
- The FS14 function set was added; the main additions to FS14 compared to FS11 are:
  - Transparency masks
  - The TIFF LZW and TIFF LZW with Differencing Predictor compression algorithms
  - The External Algorithm Specification parameter is explicitly ignored
- The TIFF LZW with Differencing Predictor compression algorithm was added
- Exception condition EC-0002 was retired
- Use of exception condition EC-*xx*10 was recommended, when possible, instead of exception condition EC-0004
- Syntax diagrams, along with some text, were updated to clarify that transparency masks are completely valid in untiled images
- Miscellaneous clarifications for color processing
- Changes were made to correct errors and omissions
- Miscellaneous clarifications
- Many small updates to increase consistency and readability

Technical changes between this edition and the previous edition are shown in green, with a green "|" revision bar in the left margin, as this text is.

Contents

Figures

Tables

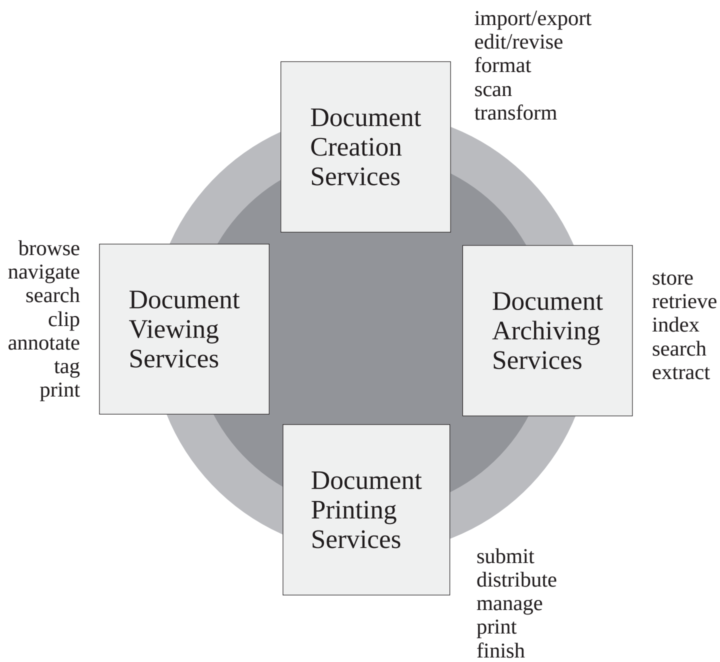
Chapter 1. A Presentation Architecture Perspective

This chapter provides a brief overview of Presentation Architecture.

The Presentation Environment

Figure 1 shows today's presentation environment.

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



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

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

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

Architecture Components

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

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

Data Streams

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

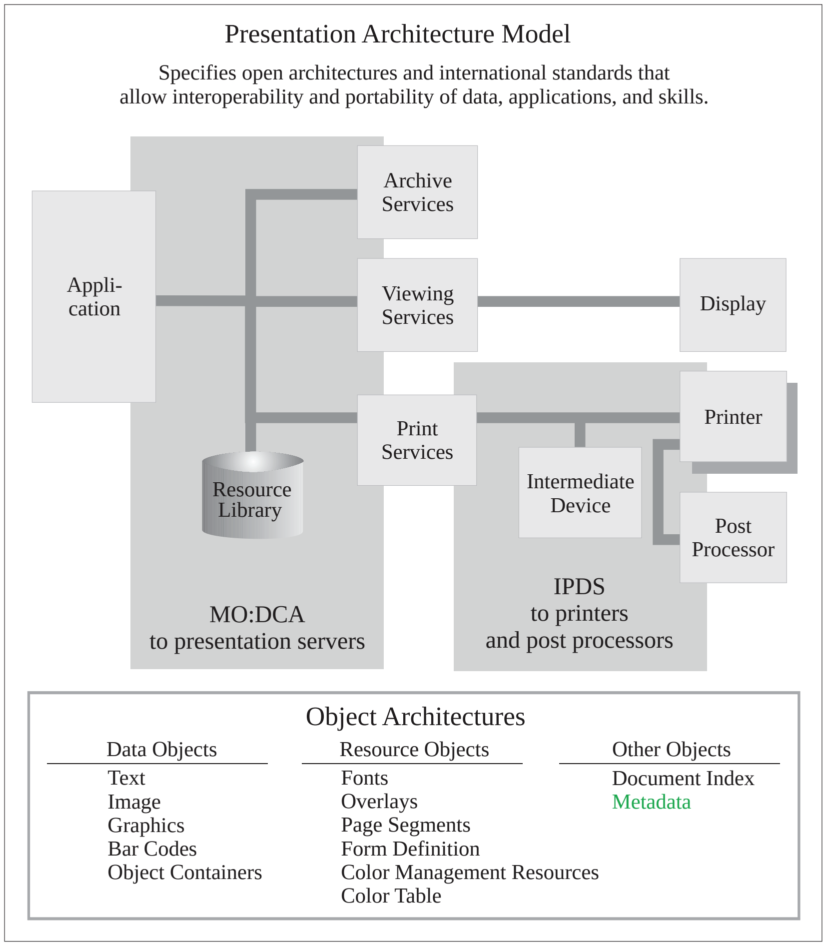
- *Mixed Object Document Content Architecture (MO:DCA)*
- *Intelligent Printer Data Stream (IPDS) Architecture*.

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

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

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

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



Objects

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

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

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

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

The AFPC-defined object content architectures are:

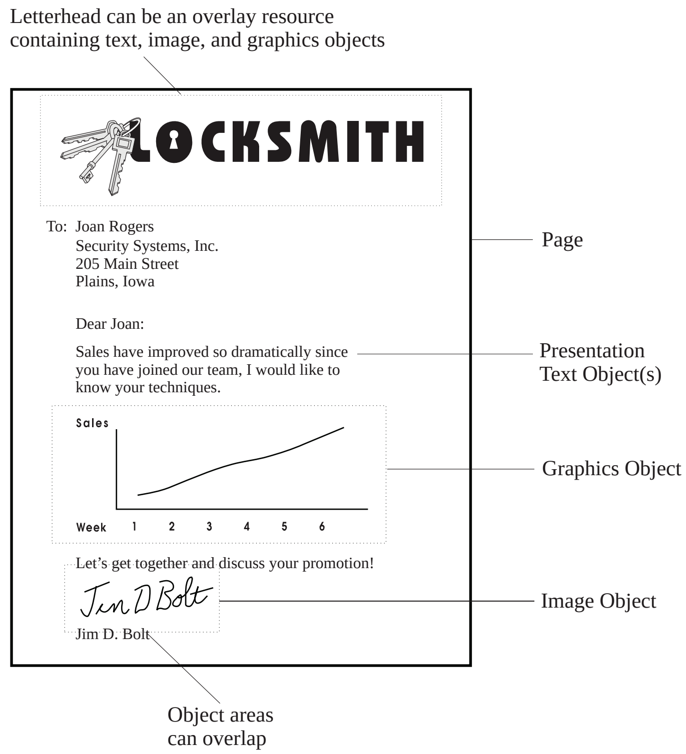
- *Presentation Text Object Content Architecture (PTOCA):* A data architecture for describing text objects that have been formatted for all-points-addressable presentations. Specifications of fonts, text color, and other visual attributes are included in the architecture definition.
- *Image Object Content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

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

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

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

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



Chapter 2. Introduction to IOCA

This chapter outlines:
- The rationale for IOCA
- The scope of IOCA

Background

An *image*, in computer terminology, is an electronic representation of a picture as an array of raster data. Image data can be generated by a computer program, or formed by electronically scanning such items as illustrations, drawings, photographs, and signatures.

The image-processing field is expanding dramatically due to advances in hardware technology. For example:

- Less expensive computer storage and memory are making the handling of larger volumes of image data increasingly more feasible; image databases are now in widespread use.
- Faster processors and techniques such as bit slicing and hardware buffering are improving the efficiency and flexibility of online image processing.
- Higher-resolution image devices are improving the usability of images and image applications. Images can now be printed and displayed in greater detail than ever before.

More and more image applications—most of which involve generating, processing, presenting, and storing images—are emerging to meet the specific needs of various industries. Insurance applications often require high-volume input and single-image manipulation. Banking applications require a verification process for handwritten check endorsements and signatures, with the ability to analyze a specific part of each image. Engineering applications might focus on design analysis systems that deal with drawings. Publishing applications might involve document creation, complex editors with image editing capabilities, and document distribution. The list of potential areas for image applications is very long and continues to grow: medicine, geology, agriculture, manufacturing, and government, to name a few.

To support the diverse image application areas, images are encoded in a number of different formats. As the technology progresses, old formats are extended and refined and new formats are being formulated.

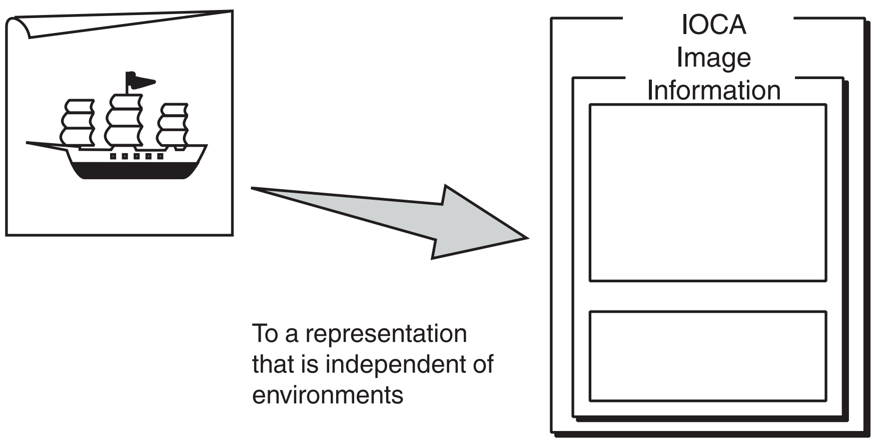
The Image Object Content Architecture (IOCA) has been formulated to provide a format suited for high speed printing. IOCA contains enough flexibility that a wide variety of images can be printed, but formats images in such a way that they can be printed efficiently and with minimal processing.

What is IOCA?

IOCA is an architecture that provides a consistent way to represent images, including conventions and directions for processing and interchanging image information. In other words, this architecture:

- Can be used for scanning, displaying, printing, archiving, and other I/O operations.
- Has an image description that is flexible enough to allow it to exist intact in interactive, printer, and interchange environments that are defined by the following data stream architectures:
  - Intelligent Printer Data Stream (IPDS) for printers
  - Mixed Object Document Content Architecture (MO:DCA)
- Allows the image to be fully described in device- and process-independent terms. Each image object is independent of other data objects and the environment in which it exists.
- Describes images using self-defining fields; that is, each field contains a description of itself along with its contents.

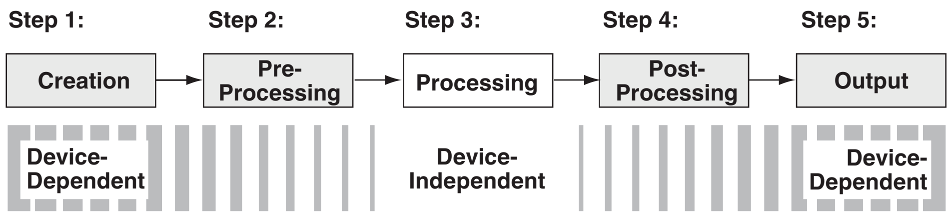
Figure 4. Images and IOCA



IOCA in Image Processing

Figure 5 summarizes the steps typically involved in image processing, and indicates which stages are device dependent. IOCA is involved only in Step 3, device-independent information processing. The term *IOCA Process Model* is used hereafter when referring to this step. The other steps are device dependent, and the interface to them is provided by the *controlling environments*.

Figure 5. Steps in Image Processing



1. *Creation*. An image is created by a program or an input device such as a scanner. The creation step is supported by many types of devices and technologies. The resulting image contains device-dependent information.
2. *Pre-processing*. Pre-processing is the gateway from the input devices. In this step, the device-dependent information is removed from the image. For example, if the image was created by scanning a document, the end-of-scan-line code is removed. After this step, the image, along with its characteristics of resolution and size, is ready to be processed.
3. *Processing*. The image is now processed into an interchangeable form with all device-dependent characteristics removed. In this form, it can be passed to another system or environment and interpreted consistently.
4. *Post-processing*. Post-processing is the gateway to applications that support output devices. The required device-control information is inserted. This step might be different for each type of device.
5. *Output*. This step presents the image to the user. It is controlled locally by the output device, such as a display or a printer.

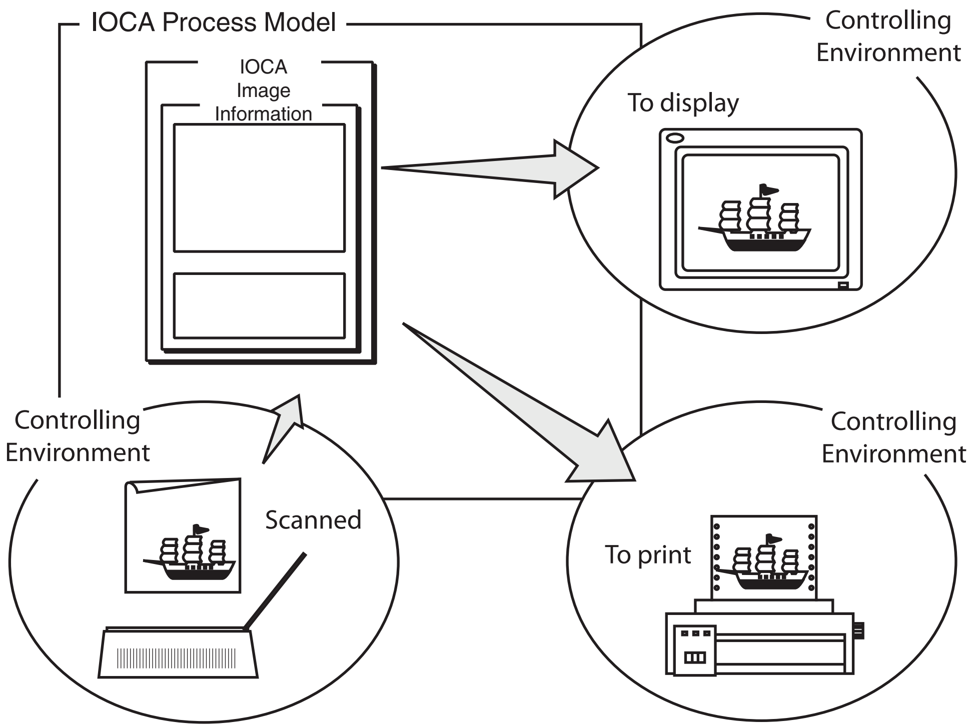
The IOCA Process Model

IOCA uses the *Image Segment* as its base unit for representing an image. An Image Segment consists of image data and the parameters needed to describe that image's characteristics in a universally recognizable way.

The IOCA Process Model communicates with the controlling environment, sending and receiving Image Segments to and from them. It also takes action if irregularities are found in the IOCA Image Segments.

Figure 6 shows the relationship between the IOCA Process Model and the controlling environments that scan, display, and print IOCA Image Segments.

Figure 6. IOCA Process Model and the Controlling Environments



Mixed Object Document Content Architecture (MO:DCA) and Intelligent Printer Data Stream (IPDS) are examples of controlling environments.

Chapter 3. IOCA Overview

This chapter outlines:

IOCA Representation of Images

IOCA provides a way to represent images in a device-independent format, which allows them to be interchangeable across environments. IOCA uses a consistent set of constructs, called *self-defining fields*, to describe the characteristics of the image data. A self-defining field is a field that contains one or two bytes identifying the content of that field.

An image consists of *image points*. Each image point is represented by one or more bits of information, called *Image Data Elements* (IDEs). IDEs are grouped together into *Image Data*. Image Data is known as non-coded information (NCI) since no codes are embedded in it. This characteristic makes Image Data different from either text or graphic data.

Note: Non-coded information does not contain any IOCA codes that would impact the presentation of image points. The data, however, may be carried in compressed format, such as JPEG, that contains codes that specify how the data is compressed.

Certain properties characterize the image, and must be processed in order to interpret the data properly, such as:

- Size (how large)
- Resolution (how sharp)
- Color (whether it is black-and-white, grayscale, or color)
- Recording and compression algorithms (how Image Data is encoded)
- Image Data layout

Image data parameters encapsulate these properties and separate them from the image data. The Image Data and Image Data parameters are collectively referred to as the *Image Content*.

The Image Contents are independent of the controlling environment in which they exist. In every controlling environment, an image can be represented by its Image Contents alone.

When an image is carried in data streams, all of its image components are contained in *Image Segments*.

The Image Segment, a set of self-defining fields, is passed to and from controlling environment, which determine how it is handled. That is, the Image Segment can be presented as a displayed or a printed image in an environment, or can be merged with text and graphics objects into a compound document.

Figure 7. Image Concept and IOCA Representation

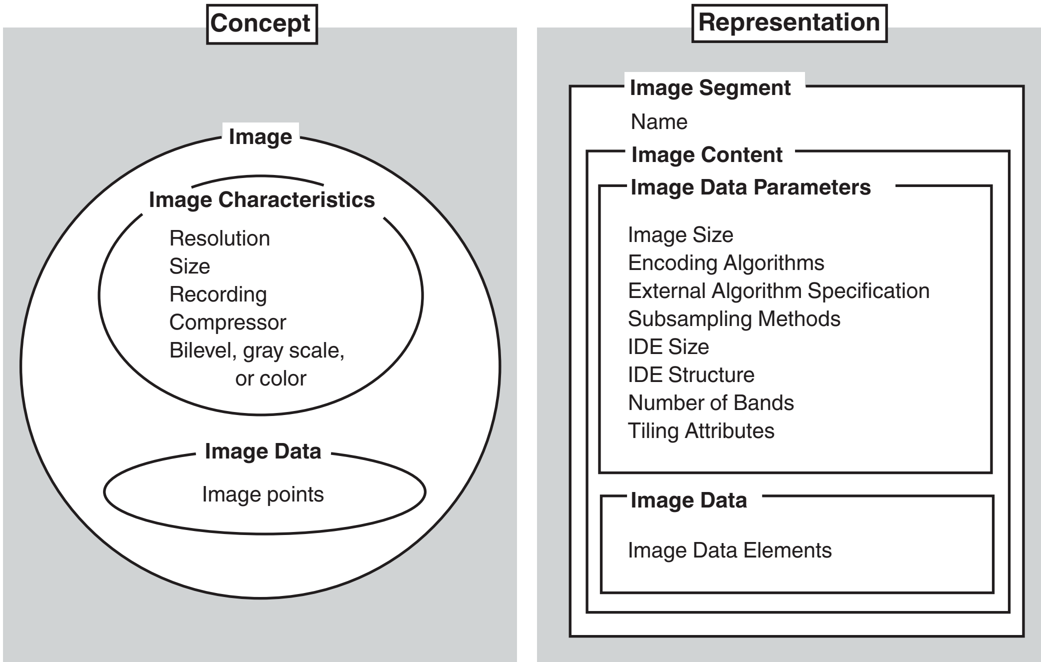


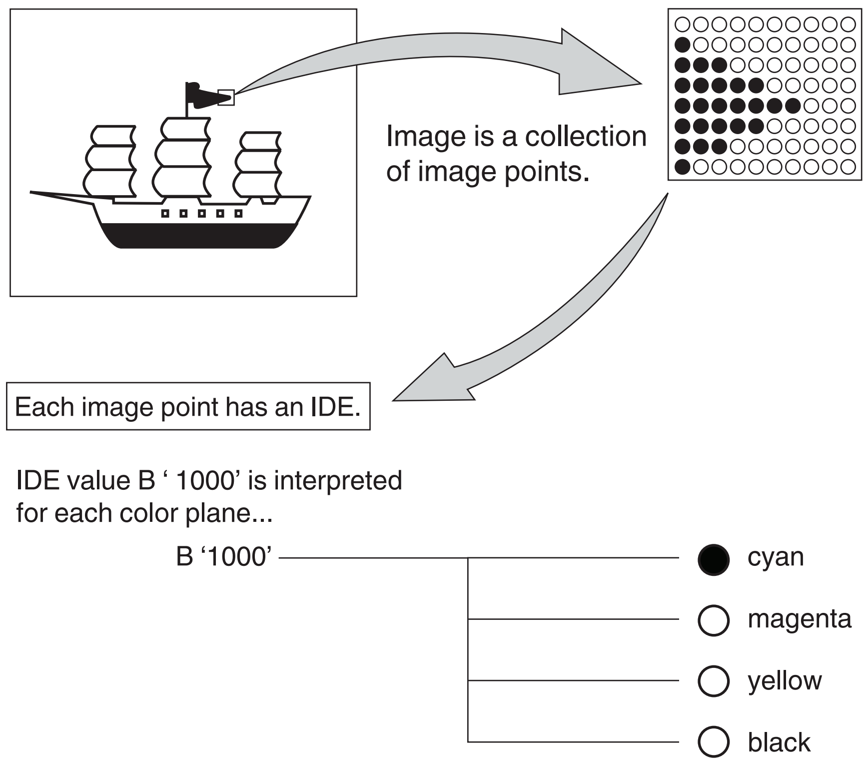
Image Points

When digitized for processing, images are expressed by a two-dimensional array of pixels, called *image points*.

Each image point has information called the *image data element* (IDE). The IDE has one or more bits that are interpreted in the context of the current color space to determine its property, such as black, white, grayscale, or color.

Consider a color image in the CMYK color space that is represented by four bits per IDE. Figure 8 shows how an intensified image point, say an IDE with a binary value of B'1000', is interpreted.

Figure 8. Image Point and IDE



The image foreground and background are defined as follows:

- For bilevel images, the image foreground consists of all those image points whose IDE values are B'1'. The rest of the image points along with the unoccupied areas of the Image Presentation Space (IPS) are considered to be the image background.
- For any other images, the entire image is considered to be foreground. The unoccupied areas in the image presentation space are the image background.

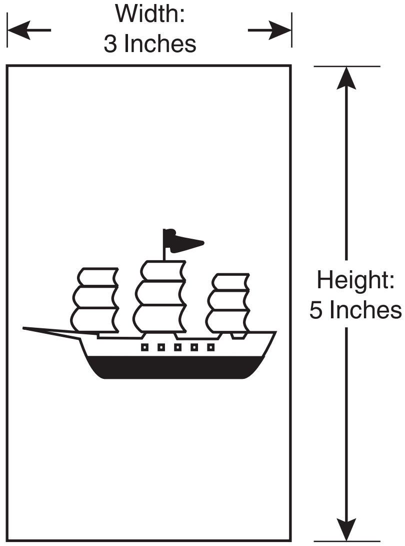
Size and Resolution

In addition to color, images are characterized by their size and resolution.

- The size of an image is expressed in terms of the number of image points in the horizontal and vertical directions.
- The resolution of an image determines its sharpness. It is expressed in terms of the number of image points in the measurement base, in the horizontal and vertical directions. The measurement base, indicated by *unit base*, can be 10 inches or 10 centimeters.

Figure 9 shows how an image's size and resolution are calculated:

Figure 9. Image Resolution



If the image is divided into 600 image points horizontally and 1500 image points vertically, the image is represented as:

- Sizes:

600	Horizontal
1500	Vertical

- Resolutions:

200 image points/inch	Horizontal
300 image points/inch	Vertical

Compression

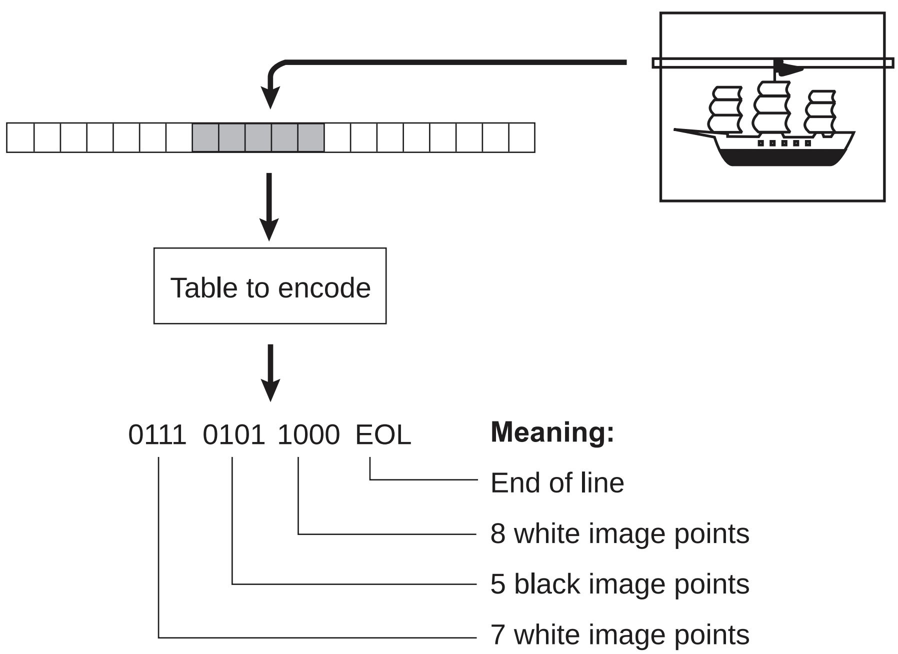
Consider an image that has the dimensions of letter-size paper. If it is represented in black and white (bilevel, represented by one bit per IDE) at 600dpi, its image data would be about 3,366,000 bits long. Such large data volumes are expensive to process, store, and transmit.

The size of an image's data can be reduced by one of many compression techniques. In order to reconstruct a compressed image, an application or device must know which compression technique was used to compress the data. IOCA provides two self-defining fields, the Image Encoding parameter and the External Algorithm Specification parameter, to describe the compression algorithm.

In the Image Data, it is not unusual to find lengthy strings of IDEs that all have the same value. Compression algorithms use codes to represent these strings in the Image Data.

Figure 10 shows a compression example that takes advantage of IDE repetitions in the Image Data. The compression algorithm represents a group of similar IDEs by the length of that group.

Figure 10. Image Compression



The effectiveness of compression algorithms differs depending on the content of the image. The compression algorithm has to be matched to the data type. For example, bilevel text, business graphics such as a pie chart, and a color photograph will each require a different compression algorithm.

Image Coordinate System

Each Image Content, which consists of image data and image characteristics information, has a coordinate system, called the *Image Coordinate System*. This is an X-Y Cartesian system that uses only the fourth quadrant and positive values for the Y-axis. In other words, the origin is top left. Units along the X and Y axes correspond directly to image points that are represented by IDEs in the Image Content.

Figure 11. Image Coordinate System

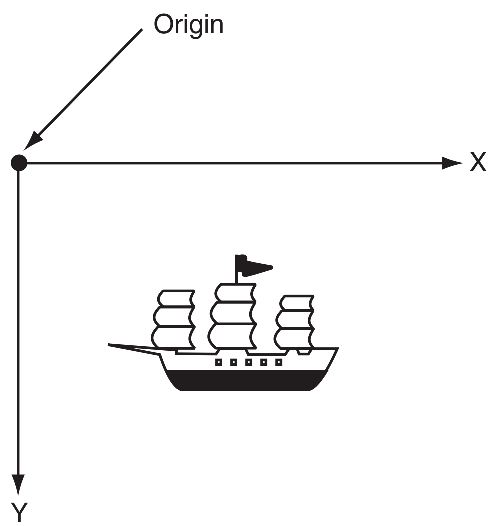


Image points in the horizontal direction are mapped in the X direction of the Image Coordinate System.

Image points in the vertical direction are mapped in the Y direction of the Image Coordinate System.

Image Presentation Space

Before an Image Content can be displayed or printed, it is placed in a conceptual space, called an *Image Presentation Space* (IPS). The physical characteristics of the IPS are defined and provided by the controlling environment. The IPS is two-dimensional, and has an Image Coordinate System. It acts as a bridge between the IOCA Process Model and the controlling environment.

Figure 12. Image Presentation Space

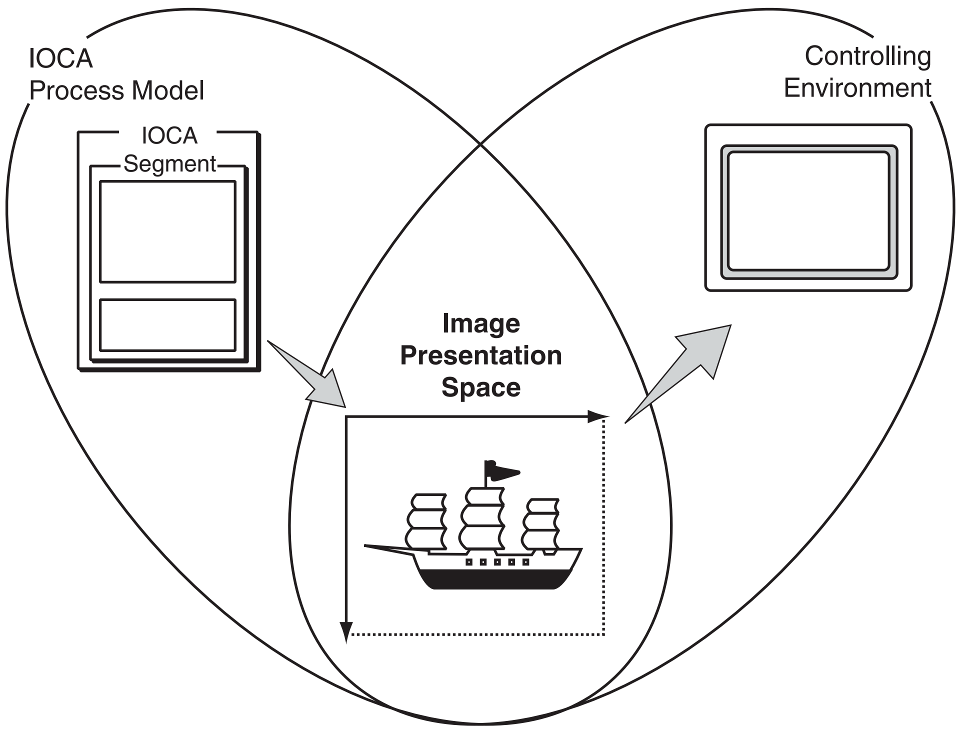


Image Tiling

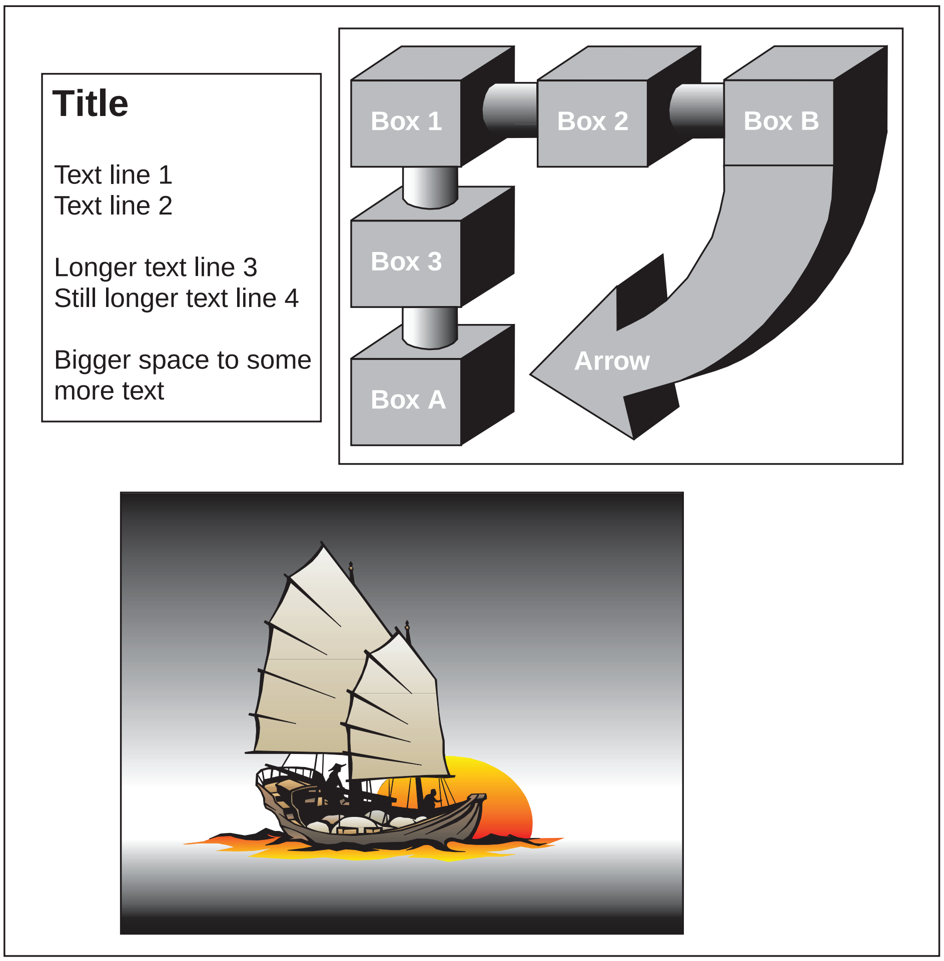
For large images, such as engineering drawings, it is often advantageous to partition the image into smaller non-overlapping rectangular pieces called *tiles*.

Each tile can be thought of as an individual image. The tiles may differ in the color space, encoding, and compression algorithms, but must have resolution that evenly divides the underlying Image Presentation Space resolution. The tiles need not cover the whole Image Presentation Space.

IOCA provides a series of self-defining fields to encode tiling information.

Figure 13 illustrates an image composed of three tiles, each with a different data type.

Figure 13. Tiles of an Image



Function Sets

For some applications, it is not necessary or feasible to implement all the features in the architecture, or support the entire range of values and parameters in a self-defining field.

Chapter 7, "Compliance", on page 83 defines several subsets of the architecture (called *function sets*) that satisfy some particular common needs. It is the responsibility of the application to determine which function set(s) it must provide to generate and receive IOCA Image Objects.

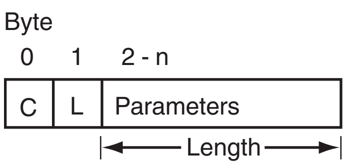
Chapter 4. Formats and Codes

This chapter describes the formats of the IOCA self-defining fields.
- The formats of the IOCA self-defining fields
- The code points used by IOCA

Formats

An IOCA Image Segment is a set of self-defining fields. Each self-defining field is in either long format or extended format. Both formats start with a code for the self-defining field, and the length of the parameters that follow.

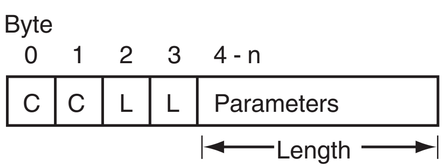
Long Format



where:

C is a one-byte code for the self-defining field.
L is the length of the following parameters, excluding L itself.

Extended Format



where:

CC is a two-byte code for the self-defining field. The first byte is always X'FE'.

This format is used by all of the following:
- Image Data (X'FE92')
- Band Image Data (X'FE9C')
- Include Tile parameter (X'FEB8')
- Tile TOC parameter (X'FEBB')
- Image Subsampling parameter (X'FECE')

Other values for the second byte of CC are reserved.

LL is the length of the parameters, excluding LL itself.

Code Points

Table 3 lists the codes used by IOCA, the names of the associated elements, and the formats used.

Table 3. IOCA Code Points

Code	Name	Format
X'70'	Begin Segment	Long format
X'71'	End Segment	Long format
X'8C'	Begin Tile	Long format
X'8D'	End Tile	Long format
X'8E'	Begin Transparency Mask	Long format
X'8F'	End Transparency Mask	Long format
X'91'	Begin Image Content	Long format
X'93'	End Image Content	Long format
X'94'	Image Size	Long format
X'95'	Image Encoding	Long format
X'96'	IDE Size	Long format
X'97'	Image LUT-ID (Retired)	Long format
X'98'	Band Image	Long format
X'9B'	IDE Structure	Long format
X'9F'	External Algorithm Specification	Long format
X'B5'	Tile Position	Long format
X'B6'	Tile Size	Long format
X'B7'	Tile Set Color	Long format
X'F4'	Set Extended Bilevel Image Color	Long format
X'F6'	Set Bilevel Image Color	Long format
X'F7'	IOCA Function Set Identification	Long format
X'FE92'	Image Data	Extended format
X'FE9C'	Band Image Data	Extended format
X'FEB8'	Include Tile	Extended format
X'FEBB'	Tile TOC	Extended format
X'FECE'	Image Subsampling	Extended format

Chapter 5. IOCA Image Segment

This chapter:

- Briefly describes the IOCA Image Segment
- States the purpose of each IOCA self-defining field in the Image Segment
- Provides the syntax and semantics of each self-defining field, its parameter set, and its exception conditions

For an explanation of the layout of the syntax diagrams in this chapter, see “How to Read the Syntax Diagrams” on page v. For an explanation of the notation conventions, see “Notation Conventions” on page vi.

Image Segment

An Image Segment is represented by a set of self-defining fields, fields that describe their own contents. It starts with a Begin Segment, and ends with an End Segment.

Between the Begin Segment and End Segment is the image information to be processed, called the Image Content.

The Image Content can be either untiled or tiled.

Untiled image content consists of:

- Image Data parameters that describe the characteristics of the image data
- An optional Transparency Mask
- Zero or more image data elements: Image Data and Band Image Data

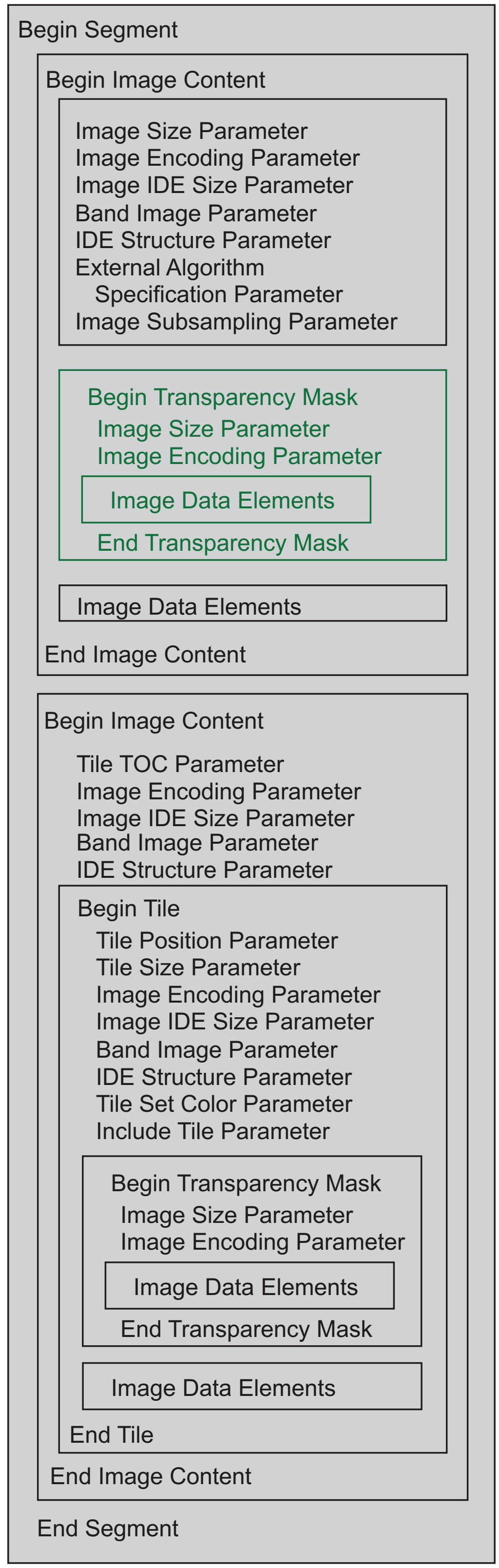
Tiled image content consists of:

- Image Data parameters that describe the characteristics of the image content
- Zero or more Tiles

Each tile consists of:

- Image Data parameters that describe the characteristics of the image data
- An optional Transparency Mask
- Zero or more image data elements: Image Data and Band Image Data

Multiple image contents may exist within a single IOCA image segment. All image contents share the same Image Presentation Space and are presented in the order they appear.



Begin Segment

The Begin Segment parameter defines the beginning of the Image Segment.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'70'	Begin Segment	M
1	UBIN	LENGTH	X'00' – X'04'	Length of the parameters to follow	M
2	UBIN	NAME	X'00000000' – X'FFFFFFFF'	Name of the Image Segment	O

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-0005 Invalid length

Condition: The LENGTH value is not in the valid, function-set specified range. EC-0005 is optional—IOCA receivers can generate EC-0003 instead of EC-0005.

EC-700F Invalid sequence

Condition: A Begin Segment is missing, or it appeared out of sequence or more than once. IOCA receivers can generate an out-of-sequence exception condition—EC-*xx*0F—instead of EC-700F, where *xx* is the one-byte ID code of the IOCA self-defining field encountered in place of the Begin Segment self-defining field.

End Segment

The End Segment parameter defines the end of the Image Segment.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'71'	End Segment	M
1	UBIN	LENGTH	X'00'	Length of the parameters to follow	M

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-710F Invalid sequence

Condition: An End Segment is missing, or it appeared out of sequence.

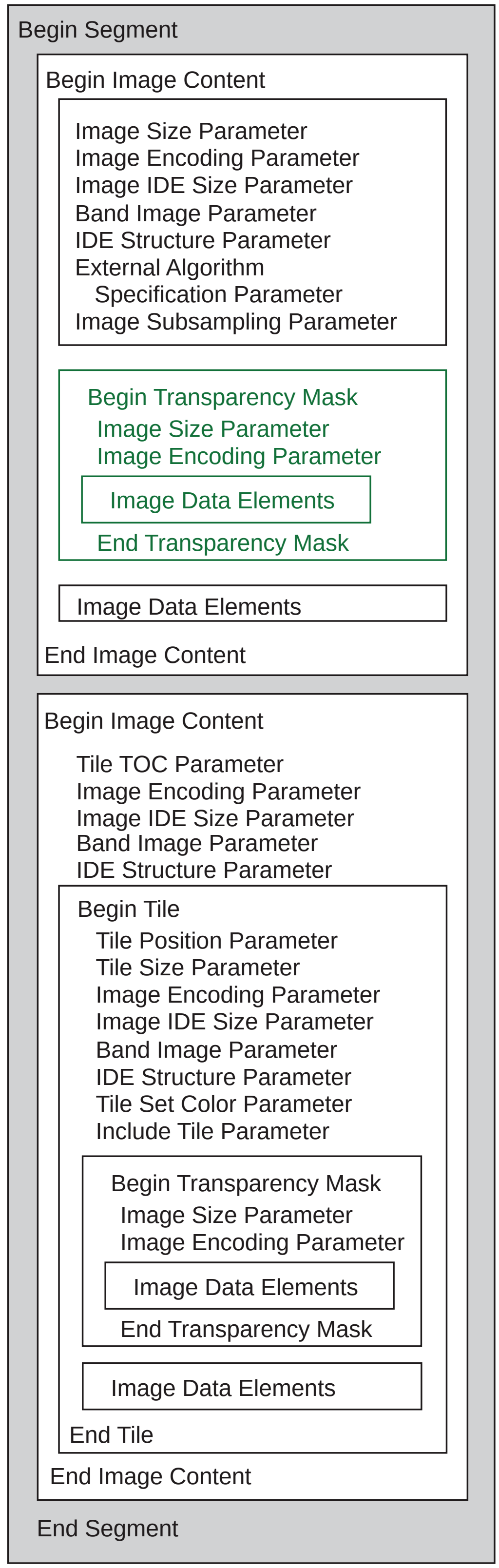
Image Content

An Image Content begins with a Begin Image Content and ends with an End Image Content.

The Image Content can be either untiled or tiled.

If the Image Content is untiled, it contains a number of Image Data parameters, followed by an optional Transparency Mask, followed by the Image Data. The Image Data is contained in one or more self-defining fields. The same Image Data parameter cannot appear more than once within a single Image Content.

If the Image Content is tiled, it starts with a Tile Table of Contents, followed optionally by a number of parameters that set the default values, followed by zero or more Tiles. The structure of each tile is very similar to that inside an untiled Image Content, with Image Data parameters, an optional Transparency Mask, and Image Data.



Begin Image Content

The Begin Image Content parameter defines the beginning of the Image Content.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'91'	Begin Image Content	M
1	UBIN	LENGTH	X'01'	Length of the parameters to follow	M
2	CODE	OBJTYPE	X'FF'	Object type: **X'FF'** IOCA image object All other values are reserved.	M

Notes:

1. IOCA allows multiple image contents in a single Image Segment, but the receivers are not required to support more than one image content in each image segment. If a receiver that does not support multiple image contents in a single image segment receives a second Begin Image Content Parameter in an image segment, exception EC-910F exists.
2. All receivers that support multiple image contents must support at least 128 image contents per image segment.
3. Architecture does not restrict the number of image contents contained within a single image segment. If an image segment contains too many image contents for a receiver to present, the receiver should take the same action as if too many image objects were specified on a page.
4. If a receiver supports multiple image contents, it must support them for any type of image. For example, such a receiver must process multiple image contents containing FS10 data without raising an exception, even though the FS10 definition specifies a single image content in each image segment.
5. Multiple image contents are treated by the receiver as if they were sent as multiple image objects, in the same order in which they appear in the image segment.
6. All of the image contents are presented using the same Image Presentation Space characteristics, as defined in the image data descriptor for the image object.
7. Function Sets 45 and 48 are the only current function sets that require receivers to support multiple image contents in a single image segment.

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-0004 Invalid parameter value

Condition: The OBJTYPE value is not in the valid range.

EC-910F Invalid sequence

Condition: One or more of the following conditions holds:

- A Begin Image Content is missing, or it appeared out of sequence. IOCA receivers can generate an out-of-sequence exception condition—EC-*xx*0F—instead of EC-910F, where *xx* is the one-byte ID code of the IOCA self-defining field encountered in place of the Begin Image Content parameter.
- The Begin Image Content has appeared more than once and the receiver supports only a single image content in each image segment.

End Image Content

The End Image Content parameter defines the end of the Image Content.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'93'	End Image Content	M
1	UBIN	LENGTH	X'00'	Length of the parameters to follow	M

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-930F **Invalid sequence**

Condition: An End Image Content is missing, or it appeared out of sequence. IOCA receivers can generate an out-of-sequence exception condition—EC-*xx*0F—instead of EC-930F, where *xx* is the one-byte ID code of the IOCA self-defining field encountered in place of the End Image Content parameter.

Image Data Parameters

Image Data parameters describe the characteristics of the image data within a particular Image Content. They do not affect the image data in other Image Contents.

This section describes:
- Image Size parameter
- Image Encoding parameter
- IDE Size parameter
- Band Image parameter
- IDE Structure parameter
- External Algorithm Specification parameter
- Image Subsampling parameter

The Image Size parameter must exist in each untiled Image Content; the other Image Data parameters are optional. The Image Size parameter must not exist in a tiled image content. Some optional parameters are not permitted in some Function Sets. If you omit an optional parameter permissible in the function set, its default value is used.

In a tiled Image Content, the Image Data parameters described in this section can appear either within Tiles or before the first tile. Any value set in an Image Data parameter specified before the first tile is used as a default in all the tiles. The same Image Data parameter can appear outside of tiles and within a tile, in which case the values specified within the tile are used.

A function set is a set of self-defining fields that describes an Image Object. For more information on function sets, see "Function Sets" on page 83.

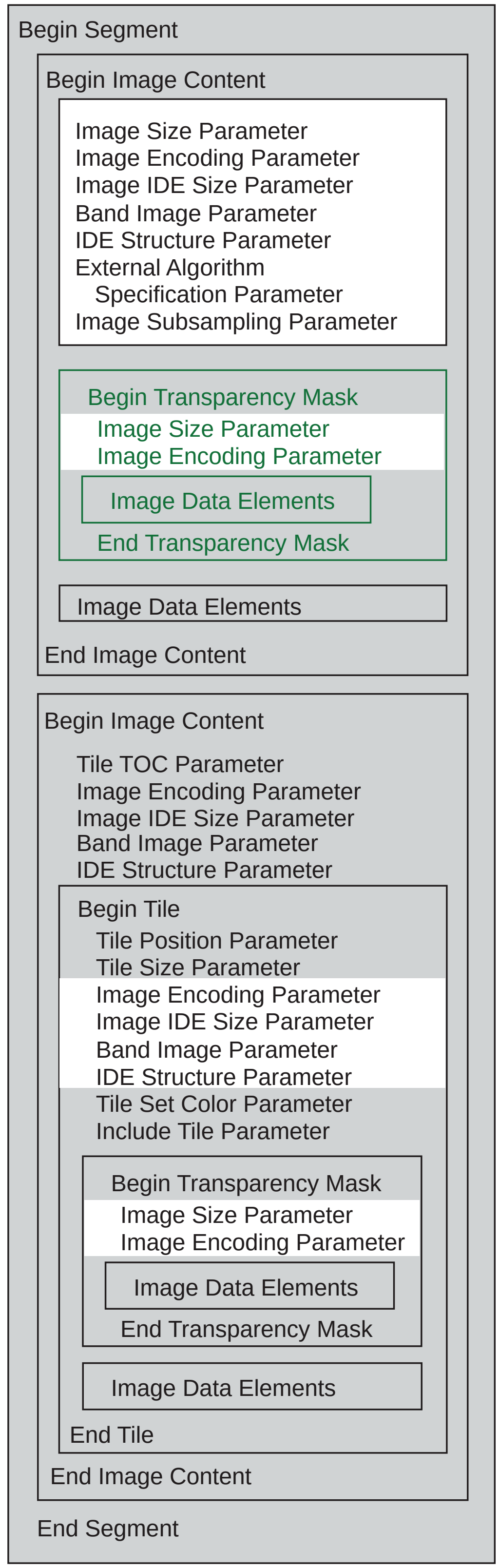


Image Size

This self-defining field, which is mandatory in non-tiled Image Contents, describes the measurement characteristics of the image when it is created. There is no default value.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'94'	Image Size parameter	M
1	UBIN	LENGTH	X'09'	Length of the parameters to follow	M
2	CODE	UNITBASE	X'00' – X'02'	Unit base: **X'00'** 10 inches **X'01'** 10 centimeters **X'02'** Logical (resolution ratio) All other values are reserved.	M
3–4	UBIN	HRESOL	X'0000' – X'7FFF'	Horizontal resolution	M
5–6	UBIN	VRESOL	X'0000' – X'7FFF'	Vertical resolution	M
7–8	UBIN	HSIZE	X'0000' – X'7FFF'	Horizontal size in image points (excluding any padding bit in each scan line)	M
9–10	UBIN	VSIZE	X'0000' – X'7FFF'	Vertical size in image points (excluding any padding scan line)	M

UNITBASE=X'02' (logical) indicates that the following HRESOL and VRESOL specify a ratio of the horizontal and vertical resolutions.

The combinations of UNITBASE, HRESOL, and VRESOL have the following meanings:

- When UNITBASE=X'00' or X'01':
  - When HRESOL or VRESOL (or both) is zero, the resolution of the Image Content in that direction is *undefined*. Image Contents with undefined resolutions are written with each image point mapped onto one point in the Image Presentation Space.
  - Nonzero HRESOL or VRESOL values, divided by 10, yield the number of image points per inch or per centimeter in the corresponding direction.

    Example: If the distance between image points is 1/200th of an inch, the resolutions are specified as X'0007D007D0'. This means that there are 2000 image points per 10 inches in both the horizontal and vertical directions.
- With UNITBASE=X'02':
  - When either HRESOL or VRESOL is zero, the Image Content's resolutions in both directions are undefined. Image Contents with undefined resolutions are written with each image point mapped on a point in the Image Presentation Space.
  - Dividing a nonzero HRESOL value by a nonzero VRESOL value yields the ratio of the horizontal and vertical resolutions.

    Example: X'0200010002' means that the vertical resolution is twice the horizontal resolution, and that the image is sharper in the vertical direction than in the horizontal direction. To keep this ratio, the controlling environment allows you to define the Image Presentation Space so as to have the doubled resolution in the vertical direction.

The total number of image points, excluding any padding bit and padding scan line, in the image data can be obtained by multiplying the nonzero HSIZE and VSIZE values.

For non-tiled images, HSIZE=X'00' means that the image data has an *unknown* horizontal size, and VSIZE= X'00' means that it has an unknown vertical size. These are valid only for compression algorithms where the IOCA Process Model can determine the width or height of the image from the image data during decompression time.

Note: The width or height determined by the IOCA Process Model may be larger than the actual image width or height, as the image data may include padding bits or padding scan lines.

HSIZE=X'00' or VSIZE=X'00' for other compression algorithms raises exception condition EC-9411. See Appendix A, “Compression and Recording Algorithms”, on page 137 for details.

When VSIZE=X'00', the actual vertical size of such image data is determined after all image data is received. For example, with IBM MMR-Modified Modified Read, the vertical size is determined when the end-of-page (EOP) condition is detected. See Appendix A, “Compression and Recording Algorithms”, on page 137 for details.

Note: IOCA generators should set HSIZE and VSIZE to the image's actual width and height regardless of the compression algorithm used. Setting either HSIZE or VSIZE to zero might cause some IOCA receivers to abort prematurely.

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-0004 **Invalid parameter value**

Condition: The HRESOL, VRESOL, HSIZE, or VSIZE value is not in the valid range.

Note: It is recommended that IOCA receivers generate exception EC-9410 instead of exception EC-0004 for this condition.

EC-940F **Invalid sequence**

Condition: An Image Size parameter is missing, or it appeared out of sequence or more than once.

EC-9410 **Invalid or unsupported Image Data parameter value**

Condition: The Image Size parameter contains an invalid or unsupported value.

EC-9411 **Inconsistent Image Data parameters, or inconsistent Image Data parameter and Image Data**

Condition: HSIZE or VSIZE is zero (X'0000'), and the size in that direction cannot be determined from the image data.

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

EC-9401 **Inconsistent Image Size parameter value and Image Data**

Condition: The size detected in the image data is different from the HSIZE or VSIZE value of the Image Size parameter.

System action: The size detected from the image data is used.

Image Encoding

This optional self-defining field describes the algorithms by which the image data is encoded. See Appendix A, “Compression and Recording Algorithms”, on page 137 for details.

Syntax

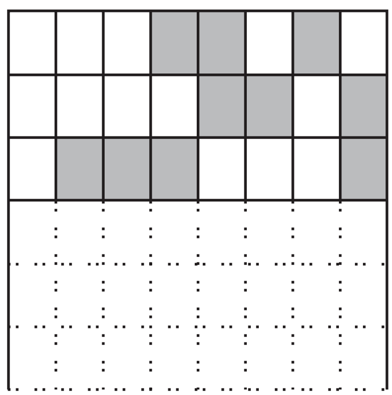
Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'95'	Image Encoding parameter	M
1	UBIN	LENGTH	X'02' – X'03'	Length of the parameters to follow	M
2	CODE	COMPRID	X'00' – X'0E', X'20', X'80' – X'84', X'A0' – X'AF', X'FE'	Compression algorithm: **X'00'** (Retired) **X'01'** IBM MMR-Modified Modified Read **X'02'** (Retired) **X'03'** No compression **X'04'** (Retired) **X'05'** (Retired) **X'06'** RL4 (Run Length 4) **X'07'** (Retired) **X'08'** ABIC (Bilevel Q-Coder) **X'09'** TIFF algorithm 2 **X'0A'** Concatenated ABIC **X'0B'** Color compression used by OS/2 Image Support **X'0C'** TIFF PackBits **X'0D'** TIFF LZW **X'0E'** TIFF LZW with Differencing Predictor **X'20'** Solid Fill Rectangle **X'80'** G3 MH-Modified Huffman (ITU-TSS T.4 Group 3 one-dimensional coding standard for facsimile) **X'81'** G3 MH-Modified READ (ITU-TSS T.4 Group 3 two-dimensional coding option for facsimile) **X'82'** G4 MMR-Modified Modified READ (ITU-TSS T.6 Group 4 two-dimensional coding standard for facsimile) **X'83'** JPEG algorithms (See the External Algorithm Specification parameter for detail) **X'84'** JBIG2 **X'A0'– X'AF'** (Retired) **X'FE'** User-defined algorithms (see the External Algorithm Specification parameter for details) All other values are reserved.	M
3	CODE	RECID	X'00' – X'04', X'FE'	Recording algorithm: **X'00'** (Retired) **X'01'** RIDIC (Recording Image Data Inline Coding) **X'03'** Bottom-to-Top **X'04'** Unpadded RIDIC **X'FE'** See the External Algorithm Specification parameter for details All other values are reserved.	M
4	CODE	BITORDR	X'00' – X'01'	Bit order within each image data byte: **X'00'** Left-to-right **X'01'** Right-to-left All other values are reserved.	O

Notes:

1. When COMPRID or RECID are X'FE', the External Algorithm Specification parameter must also be present within the same Image Content, otherwise exception condition EC-9F01 exists.
2. The External Algorithm Specification Parameter is no longer required when COMPRID is X'83'. If the decompressor in the receiver fails because the compressed datastream requires a feature unimplemented in the decoder, exception EC-9511 occurs.
3. The Solid Fill Rectangle compression algorithm can be used only within tiled images, for bilevel tiles. Otherwise, exception EC-9510 occurs. This compression algorithm indicates that all the image points in the tile are set to the same color and that the tile does not contain any actual image data.
4. JBIG2 is a toolkit with many different capabilities. The standard recognizes a number of profiles that serve the same function as Function Sets in IOCA. Receivers declaring the JBIG2 support must support at least one JBIG2 profile, but are not obliged to support all of them. If a receiver encounters JBIG2-compressed data encoding unsupported function, exception EC-9511 occurs.
5. LZW encoders sometimes terminate the data early. For either the TIFF LZW or the TIFF LZW with Differencing Predictor compression algorithms, if the LZW decoder does not produce the expected number of bytes, no exception should be raised and the receiver should fill the remaining data with binary zeros.

BITORDR indicates the bit order within each image data byte. Figure 14 on page 35, for example, shows a bilevel image with a width of eight image points:

Figure 14. Top Three Lines of a Bilevel Image



The uncompressed serial bit stream for the top three lines would be:

```
B'00011010 00001101 01110001 ...'
```

When the bits are packed into image data bytes, with BITORDR=X'00', the first three bytes would be as follows:

```
B'00011010 00001101 01110001 ...'
```

For BITORDR=X'01', the first three bytes of the image data would be:

```
B'01011000 10110000 10001110 ...'
```

If the image data is compressed, the BITORDR parameter denotes the bit order within each compressed image data byte prior to decompression.

Zero is the default for BITORDR if it is absent.

If the Image Encoding parameter is not present, the defaults are X'03' for the compression algorithm, X'01' for the recording algorithm, and zero for the bit order.

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-0005 Invalid length

Condition: The LENGTH value is not in the valid, function-set specified range. EC-0005 is optional—IOCA receivers can generate EC-0003 instead of EC-0005.

EC-950F Invalid sequence

Condition: The Image Encoding parameter is required in some function sets but missing, or it appeared out of sequence or more than once.

EC-9510 Invalid or unsupported Image Data parameter value

Condition: The Image Encoding parameter contains an invalid or unsupported value.

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

EC-9511 Inconsistent Image Data parameters, or inconsistent Image Data parameter and Image Data

Condition: The decoder encountered one of the following conditions when decompressing the image data:

- The image data is not encoded according to the compression or recording algorithm specified in the Image Encoding parameter.
- The image data cannot be decoded successfully using the size values specified in the Image Size parameter. This condition applies to compression or recording algorithms that do not permit the image size to be encoded in the image data.
- The image data is not in complete accordance with the compression algorithm specified in the Image Encoding parameter.
- Image is encoded using the algorithm specified in the Image Encoding Parameter, but uses a function of the algorithm that is unsupported by the receiver.

System action: Receivers should attempt to present or make use of all successfully decompressed image data. Note, however, that the resulting partial image might differ from the original image.

IDE Size

This optional self-defining field specifies the number of bits that comprise each Image Data Element (IDE) in the image data, before any subsampling or compression method is performed on the IDEs.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'96'	IDE Size parameter	M
1	UBIN	LENGTH	X'01'	Length of the parameters to follow	M
2	UBIN	IDESZ	X'01' – X'FF'	Number of bits in each IDE	M

If the IDE Size parameter is not present, the default value for IDESZ is 1 (bilevel image).

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-0004 Invalid parameter value

Condition: The IDESZ value is not in the valid range.

Note: It is recommended that IOCA receivers generate exception EC-9610 instead of exception EC-0004 for this condition.

EC-960F Invalid sequence

Condition: The IDE Size parameter appeared out of sequence or more than once.

EC-9610 Invalid or unsupported Image Data parameter value

Condition: The IDE Size parameter contains an invalid or unsupported value.

EC-9611 Inconsistent Image Data parameters, or inconsistent Image Data parameter and Image Data

Condition: The compression scheme specified in the Image Encoding parameter does not support the IDE size specified in the IDE Size parameter.

Band Image

This optional self-defining field describes the format of one or more *bands* that represent an image. A band is a plane where, typically, image data of similar attributes is placed. Certain bits of an IDE can be placed into separate bands, for example the bits that represent the red, green, and blue color components of each IDE.

If the Band Image parameter is present, then the image data must be carried by the Band Image Data parameter. Each band of the image IDEs is carried by one or more Band Image Data parameters. The Band Image Data parameter is described in "Band Image Data" on page 72.

Syntax

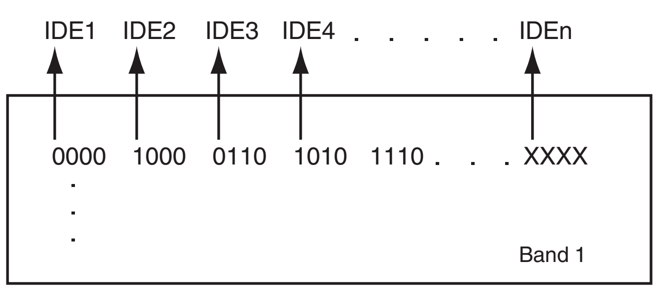
Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'98'	Band Image parameter	M
1	UBIN	LENGTH	X'02' – X'FE'	Length of the parameters to follow	M
2	UBIN	BCOUNT	X'01' – X'FD'	Number of bands	M
One or more repeating groups in the following format:					
0	UBIN	BITCNT	X'01' – X'FF'	Bit count for the band	M

BITCNT specifies how many bits of the IDE comprise one band, and BCOUNT specifies how many bands comprise the image data. The number of BITCNTs in the self-defining field must equal the BCOUNT value. The BITCNTs appear in the order in which the bits were placed into the band. For boundary alignment purposes, BITCNT can include padding bits inserted into the data. If BITCNT contains no padding bits, then the sum of all the BITCNT values equals the IDE size specified by the IDE Size parameter.

Example 1: For a single-band image with an IDE size of four with no padding bit, the first four bits of data represent the first IDE, the next four represent the second IDE, and so on.

Figure 15 illustrates the layout of image bits in this image.

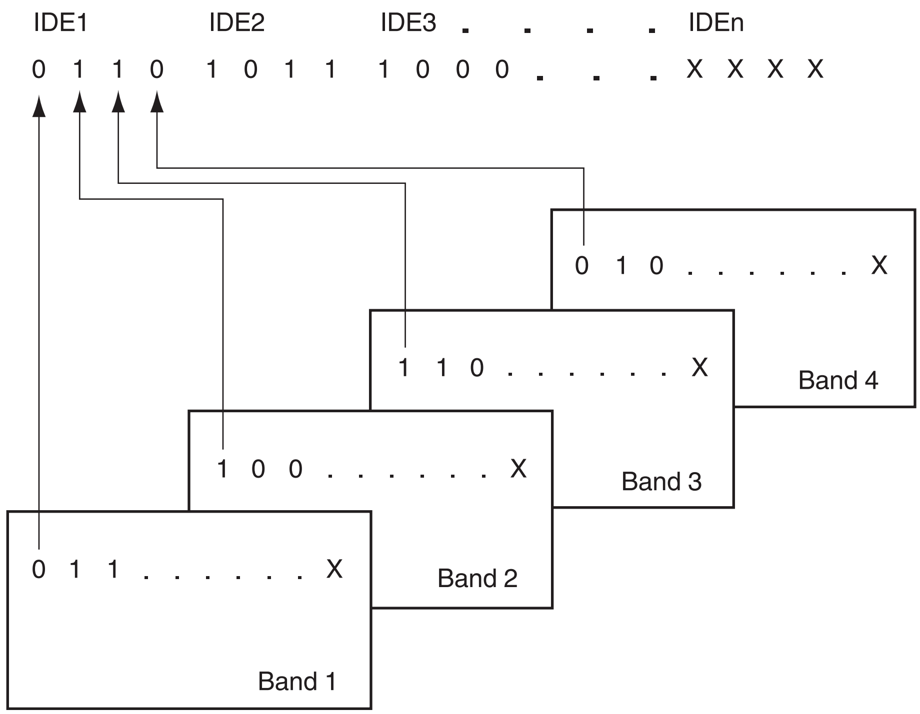
Figure 15. Example of a Four-Bit Single-Band Image with No Padding Bit



Example 2: For an image with an IDE size of four that is represented by four bands with no padding bit, the first bit in each of the four bands represents the first IDE, the second bit represents the second IDE, and so on.

Figure 16 illustrates the layout of image bits in this image.

Figure 16. Example of a Four-Bit Four-Band Image with No Padding Bit



Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-0004 **Invalid parameter value**

Condition: The BCOUNT or BITCNT value is not in the valid range.

Note: It is recommended that IOCA receivers generate exception EC-9810 instead of exception EC-0004 for this condition.

EC-0005 **Invalid length**

Condition: The LENGTH value is not in the valid, function-set specified range. EC-0005 is optional—IOCA receivers can generate EC-0003 instead of EC-0005.

EC-9801 **Invalid Band Image parameter and Image Subsampling parameter coexistence**

Condition: In some function sets, the Band Image parameter and the Image Subsampling parameter cannot coexist in the same Image Content.

EC-980F **Invalid sequence**

Condition: The Band Image parameter appeared out of sequence or more than once.

EC-9810 Invalid or unsupported Image Data parameter value

Condition: The Band Image parameter contains an invalid or unsupported value.

EC-9814 Invalid number of bands and bit counts

Condition: The number of BITCNT parameters is not equal to the BCOUNT in the Band Image parameter.

EC-9815 Invalid IDE size

Condition: The IDE size, determined by the Band Image parameter, does not match the IDE Size parameter.

IDE Structure

This optional self-defining field describes the structure of each IDE for a bilevel, grayscale, or color image.

If the IDE Structure parameter is not present, each IDE of the image data consists of a single component whose size is dependent on the IDE Size parameter. If the IDE Size is 1, the IDE value of B'1' represents a significant (toned) pel, while the value of B'0' represents an insignificant (untoned) pel. If the IDE Size is more than 1, the color model is YCbCr and the value is expressed using the Y component. This is a grayscale color model, where the value of zero represents black, while the maximum value represents white.

With this self-defining field, color images are expressed by using the RGB, YCrCb, YCbCr, CMYK, or nColor model, while grayscale images are expressed by using only the Y component of the YCrCb or YCbCr model.

See Appendix B, “Bilevel, Grayscale, and Color Images”, on page 149 for details on the relationship with the IDE Size parameter.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'9B'	IDE Structure parameter	M
1	UBIN	LENGTH	X'06' – X'14'	Length of the parameters to follow	M
2	BITS	FLAGS			M
Bit 0		ASFLAG	B'0' – B'1'	Additive or Subtractive: **B'0'** Additive **B'1'** Subtractive	
Bit 1		GRAYCODE	B'0' – B'1'	Gray coding: **B'0'** Off **B'1'** On	
Bits 2–7			B'000000'	Reserved; should be zero	
3	CODE	FORMAT	X'01', X'02', X'04', X'12', X'8*n*'	Color model: **X'01'** RGB **X'02'** YCrCb **X'04'** CMYK **X'12'** YCbCr **X'8*n*'** nColor (X'2' ≤ *n* ≤ X'F') All other values are reserved.	M
4–6			X'000000'	Reserved; should be zero	M
7	UBIN	SIZE1	X'00' – X'FF'	Number of bits/IDE for component 1	M
8	UBIN	SIZE2	X'00' – X'FF'	Number of bits/IDE for component 2	O
9	UBIN	SIZE3	X'00' – X'FF'	Number of bits/IDE for component 3	O
10	UBIN	SIZE4	X'00' – X'FF'	Number of bits/IDE for component 4	O
11	UBIN	SIZE5	X'00' – X'FF'	Number of bits/IDE for component 5	O
...					
21	UBIN	SIZE15	X'00' – X'FF'	Number of bits/IDE for component 15	O

You can specify whether increasing IDE values correspond to brighter or darker levels of gray or color, with the ASFLAG=0 (*additive*) or ASFLAG=1 (*subtractive*) parameters, respectively. ASFLAG applies to all three components of the RGB model, all four components of the CMYK model, and to the Y component of the YCrCb and YCbCr models.

- *Additive* means that the maximum color value represents full intensity of that color, while the minimum color value represents zero intensity. For example, in a black-and-white system, the minimum color value (usually zero) means black, and the maximum value means white.
- *Subtractive* means that the minimum color value represents full intensity of that color, while the maximum color value represents zero intensity. For example, in a black-and-white system, the minimum color value (usually zero) means white, and the maximum value means black.

See Appendix B, "Bilevel, Grayscale, and Color Images", on page 149 for more information on the use of ASFLAG. Note in particular that ASFLAG is ignored for bilevel images and for images that use the nColor color model.

FORMAT specifies the breakdown format for each IDE value:

- RGB means that each value is to be treated as a set of red, green, blue intensity values, and the set is in the order red, green, blue.
- YCrCb means that each value is to be treated as a set of Y, Cr, Cb values, and the set is in the order Y, Cr, Cb, where Y is the intensity, and Cr and Cb are the chrominance differences.
- YCbCr means that each value is to be treated as a set of Y, Cb, Cr values, and the set is in the order Y, Cb, Cr, where Y is the intensity, and Cb and Cr are the chrominance differences.
- CMYK means that each value is to be treated as a set of cyan, magenta, yellow, black intensity values and the set is in the order cyan, magenta, yellow, black.
- nColor means that each value is to be treated as a set of *n* separate intensity values. A color management resource from the controlling environment will be required to process the *n* values.

GRAYCODE specifies whether or not the Gray coding scheme is used to encode the image data. Gray code is a type of binary code that is applied to the entire IDE whose size is specified in the IDE Size parameter self-defining field, not just to each individual bit plane of the IDE. Gray code is constructed such that two successive codes always differ by just one bit. Table 4 shows the series of gray codes from 0 to 15 in decimal.[1]

Table 4. Gray Code Values (Decimal)

Decimal	Gray Code
0	B'0000'
1	B'0001'
2	B'0011'
3	B'0010'
4	B'0110'
5	B'0111'
6	B'0101'
7	B'0100'
8	B'1100'
9	B'1101'
10	B'1111'
11	B'1110'

1. Source: R. W. Lucky, J. Salz, and E. J. Weldon Jr., (New York: McGraw-Hill, 1968).

Table 4 Gray Code Values (Decimal) (cont'd.)

Decimal	Gray Code
12	B'1010'
13	B'1011'
14	B'1001'
15	B'1000'

Refer to R. Hunt, *The Representation of Colour in Photography, Printing and Television* (Foundation Press, 1995), for an explanation of each color model.

SIZE1, SIZE2, ..., SIZE15 specify the number of bits required to express each color component of an IDE before any subsampling or compression method is performed on the IDEs. The maximum possible value of a particular color component is equal to $2^{SIZE\mathit{m}}$-1, where $1 \le m \le 15$.

The SIZE parameter values must appear in the sequence of the color components whose size they specify. For an RGB image, this sequence is R, G, and B; for a YCrCb image, it is Y, Cr, and Cb; for a YCbCr image, it is Y, Cb, and Cr; and for a CMYK image, it is C, M, Y, and K. For an nColor image, since such an image must be matched with a color management resource (CMR) in the controlling environment, the color components must be in the order expected by the CMR.

Other than nColor, the number of SIZE parameters varies from one to four, depending on the color components that are used to express each IDE.

For bilevel and grayscale images, expressed by the YCrCb or YCbCr color model, specifying SIZE1 is sufficient; SIZE2 and SIZE3 can be omitted, or you can specify zero for them. However, any preceding SIZE parameter must be included, and zero must be specified. For example, if an image uses only the third component of a color model, then SIZE1=0 and SIZE2=0 must be specified.

Other than nColor, the SIZE4 field is only allowed for the CMYK color model (IDE Size of 4 or 32), where it is mandatory. If SIZE4 is missing for the CMYK color model, or if it appears for any other non-nColor color model, exception EC-9B18 occurs.

For the CMYK color model, 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. SIZE1, SIZE2, SIZE3, and SIZE4 in the IDE Structure parameter are nonzero and define the number of bits used in 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 $2^{SIZE\mathit{m}}$-1, where *m*=1,2,3,4. This is a device-dependent color model.

For the nColor color model, the number of SIZE parameters must be equal to the second half of the X'8*n*' value specified in the FORMAT byte of the IDE Structure; if not, exception EC-9B18 occurs. For example, if FORMAT is specified as X'87', there must be exactly 7 SIZE parameters. Each SIZE parameter defines the number of bits used by that component.

Note: If the IDE Structure parameter is not present, the defaults are ASFLAG of B'0' (additive), GRAYCODE of B'0' (off), FORMAT of YCbCr, and SIZE1 the same as the IDE size specified in the IDE Size parameter.

Exception Conditions

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

EC-0003 The LENGTH value is not in the valid range

Condition: The LENGTH value is not in the valid range.

EC-0005 Invalid length

Condition: The LENGTH value is not in the valid, function-set specified range. EC-0005 is optional—IOCA receivers can generate EC-0003 instead of EC-0005.

EC-9B0F Invalid sequence

Condition: The IDE Structure parameter is required but missing, or it appeared out of sequence or more than once.

EC-9B10 Invalid or unsupported IDE Structure parameter value

Condition: One or more of the following conditions has been encountered:

- The value of ASFLAG is invalid or unsupported.
- The value of GRAYCODE is invalid or unsupported.
- The value of FORMAT is invalid or unsupported.
- The value of a SIZE field is invalid or unsupported.

EC-9B18 Invalid IDE Structure parameter

Condition: One of the following conditions has been encountered:

- The sum of the SIZE values does not match the IDE size specified by the IDE Size parameter.
- Color model is CMYK and SIZE4 is missing.
- SIZE4 is present and the color model is not CMYK or nColor.
- More than four SIZE parameters are present and the color model is not nColor.
- Color model is nColor and the number of SIZE parameters is not equal to the second half of the FORMAT byte.

External Algorithm Specification

This optional self-defining field provides complementary information about the algorithm specified in the Image Encoding parameter. It can be used only in conjunction with that parameter.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'9F'	External Algorithm Specification parameter	M
1	UBIN	LENGTH	X'03' – X'FF'	Length of the parameters to follow	M
2	CODE	ALGTYPE	X'00', X'10'	Type of algorithm specified: **X'00'** Recording algorithm **X'10'** Compression algorithm All other values are reserved.	M
3			X'00F'	Reserved; should be zero	M
4–*n*	CODE	ALGSPEC		Recording Algorithm Specification or Compression Algorithm Specification	M

Recording Algorithm Specification

This subparameter is carried by the External Algorithm Specification parameter.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	BITS	DIRCTN		Direction of IDEs	M
Bits 0–1		IDEPTH	B'11'	Direction of successive IDEs along a line (clockwise from X-axis): **B'11'** 270 degrees All other values are reserved.	
Bits 2–3		LINEPRG	B'00'	Direction of progression of successive lines (clockwise from X-axis): **B'00'** 0 degrees All other values are reserved.	
Bit 4		RNDTRIP	B'0'	Direction of successive IDEs relative to the previous line (clockwise from the previous line): **B'0'** 0 degrees All other values are reserved.	
Bits 5–7			B'000'	Reserved; should be zero	
1	CODE	PADBDRY	X'03'	Boundary length for padding: **X'03'** 32-bit boundary All other values are reserved.	M
2	CODE	PADALMT	X'00'	Alignment for padding: **X'00'** Data left-aligned within boundary All other values are reserved.	M

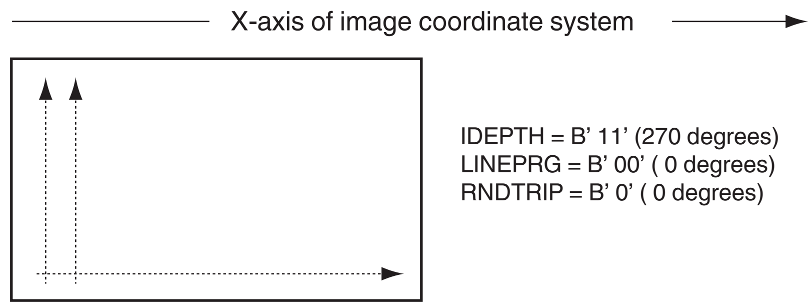
DIRCTN specifies how the IDEs are positioned in a set of image data self-defining fields. The following subparameters are defined:

- IDEPTH specifies how successive IDEs proceed along a line in relation to the X-axis of the Image Coordinate System. Degrees are measured clockwise from the X-axis of the Image Coordinate System.
- LINEPRG specifies how successive lines of IDEs proceed in relation to the X-axis of the Image Coordinate System. Degrees are measured clockwise from the X-axis of the Image Coordinate System.
- RNDTRIP specifies how the next line of IDEs proceeds in relation to the previous line. Degrees are measured clockwise from the previous line.

PADBDRY specifies if each line of the IDEs is padded with zeros where necessary for boundary alignment purposes.

PADALMT specifies whether the padding bits used for alignment purposes are located at the beginning or at the end of each line of the IDEs.

Figure 17. IDE Progression



Compression Algorithm Specification

This subparameter is carried by the External Algorithm Specification parameter. The syntax table specifies the JPEG compression algorithm that conforms to the following publications:

- ITU-TSS Recommendation T.81
- ISO/IEC International Standard 10918-1

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	COMPRID	X'83'	JPEG algorithms	M
1			X'00'	Reserved; should be zero	M
2	CODE	VERSION	X'00'	Version	M
3			X'00'	Reserved; should be zero	M
4	CODE	MARKER	X'C0' – X'C3', X'C5' – X'C7', X'C9' – X'CB', X'CD' – X'CF'	Marker code: *Non-differential Huffman coding:* **X'C0'** Baseline DCT **X'C1'** Extended sequential DCT **X'C2'** Progressive DCT **X'C3'** Lossless (sequential) *Differential Huffman coding:* **X'C5'** Differential sequential DCT **X'C6'** Differential progressive DCT **X'C7'** Differential lossless *Non-differential arithmetic coding:* **X'C9'** Extended sequential DCT **X'CA'** Progressive DCT **X'CB'** Lossless (sequential) *Differential arithmetic coding:* **X'CD'** Differential sequential DCT **X'CE'** Differential progressive DCT **X'CF'** Differential lossless All other values are reserved.	M
5–7			X'000000'	Reserved; should be zero	M

JPEG algorithms have the following restrictions:

- They cannot be applied to images whose IDE size is 1 bit/IDE.
- The baseline DCT-based algorithm is applicable only to images with 8-bits/component. The other DCT-based algorithms are applicable only to images with 8-bits/component or 12-bits/component. The IDE of the image can consist of at most four components.
- The lossless algorithms are applicable only to images with *n*-bits/component, where $2 \leq n \leq 16$. The IDE of the image can consist of at most four components.

Syntax of a User-defined Compression algorithm

Offset	Type	Name	Range	Meaning	M/O
0	CODE	COMPRID	X'FE'	User-defined compression algorithm	M
1	UBIN	LENGTH	X'04' – X'FF'	Length of the parameters to follow	M
2–5	CODE	USRCPID		Architecture-assigned user compression algorithm code point The assignment of compression code points is controlled by the IOCA data stream architecture group.	M
6–*n*		COMPSPEC	Any	User-defined specification	O

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-0005 Invalid length

Condition: The LENGTH value is not in the valid, function-set specified range. EC-0005 is optional—IOCA receivers can generate EC-0003 instead of EC-0005.

EC-9F01 Missing External Algorithm Specification parameter or Image Encoding parameter

Condition: An External Algorithm Specification parameter exists without a corresponding Image Encoding parameter, or an Image Encoding parameter exists that requires an External Algorithm Specification parameter that cannot be found.

EC-9F0F Invalid sequence

Condition: An External Algorithm Specification parameter appeared out of sequence.

EC-9F10 Invalid or unsupported Image Data parameter value

Condition: The External Algorithm Specification parameter contains an invalid or unsupported value.

EC-9F11 Inconsistent Image Data parameters, or inconsistent Image Data parameter and Image Data

Condition: An External Algorithm Specification parameter is present, but the Image Encoding parameter does not require it.

Image Subsampling

This optional self-defining field describes the *subsampling* methods used to encode the uncompressed IDEs within the image data. The methods are encoded in self-defining fields.

Subsampling is a technique of reducing the amount of image data, resulting in lower storage and processing requirements. This is accomplished by combining the color information of adjacent IDEs. If done properly, there is little or no visual degradation of the image quality.

Subsampling relies on the fact that in color images the difference between adjacent IDEs is small for certain color components. For example, in the YCrCb and YCbCr color models, most of the image information is concentrated in the Y component; hence it is fairly common to store only the average values of the Cr and Cb components of two adjacent IDEs.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0–1	CODE	ID	X'FECE'	Image Subsampling parameter	M
2–3	UBIN	LENGTH	X'0000' – X'FFFF'	Length of the parameters to follow	M
4–*n*	CODE	SDF		Zero or more self-defining fields that specify the subsampling methods	O

If the Image Subsampling parameter is not present, the default is that the IDEs have not been subsampled.

Sampling Ratios

This optional self-defining field is carried by the Image Subsampling parameter. It specifies the number of horizontal and vertical samples that make up each component of the IDEs.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'01'	Sampling Ratios	M
1	UBIN	LENGTH	X'02' – X'FE'	Length of the parameters to follow	M
1 to 127 repeating groups in the following format:					
0	UBIN	HSAMPLE	X'00' – X'7F'	Number of horizontal samples that make up the component of the IDEs	M
1	UBIN	VSAMPLE	X'00' – X'7F'	Number of vertical samples that make up the component of the IDEs	M

If the HSAMPLE and VSAMPLE group for a particular component of the IDEs is not present, the default value is 1 for both HSAMPLE and VSAMPLE. However, any preceding HSAMPLE and VSAMPLE group must be included. For example, a color image with only its third component subsampled must have HSAMPLE1, VSAMPLE1, HSAMPLE2, and VSAMPLE2 specified as equal to 1.

Example: For a 24-bit YCrCb uncompressed color image that has eight bits per component using the following sampling ratios:

```
HSAMPLE1=2    VSAMPLE1=1
HSAMPLE2=1    VSAMPLE2=1
HSAMPLE3=1    VSAMPLE3=1
```

the resulting image data layout would be as follows:

Offset	Content
0	The Y component value of the first IDE
1	The Y component value of the second IDE
2	The average of the first and second IDEs' Cr component values
3	The average of the first and second IDEs' Cb component values
4	The Y component value of the third IDE
5	The Y component value of the fourth IDE
6	The average of the third and fourth IDEs' Cr component values
7	The average of the third and fourth IDEs' Cb component values
...	...

Exception Conditions

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

EC-0003 The LENGTH value is not in the valid range

Condition: The LENGTH value is not in the valid range.

EC-0004 Invalid parameter value

Condition: The HSAMPLE or VSAMPLE value is not in the valid range.

Note: It is recommended that IOCA receivers generate exception EC-CE10 instead of exception EC-0004 for this condition.

EC-0005 Invalid length

Condition: The LENGTH value is not in the valid, function-set specified range. EC-0005 is optional—IOCA receivers can generate EC-0003 instead of EC-0005.

EC-CE01 Invalid Band Image parameter and Image Subsampling parameter coexistence

Condition: In some function sets, the Band Image parameter and the Image Subsampling parameter cannot coexist in the same Image Content.

EC-CE0F Invalid sequence

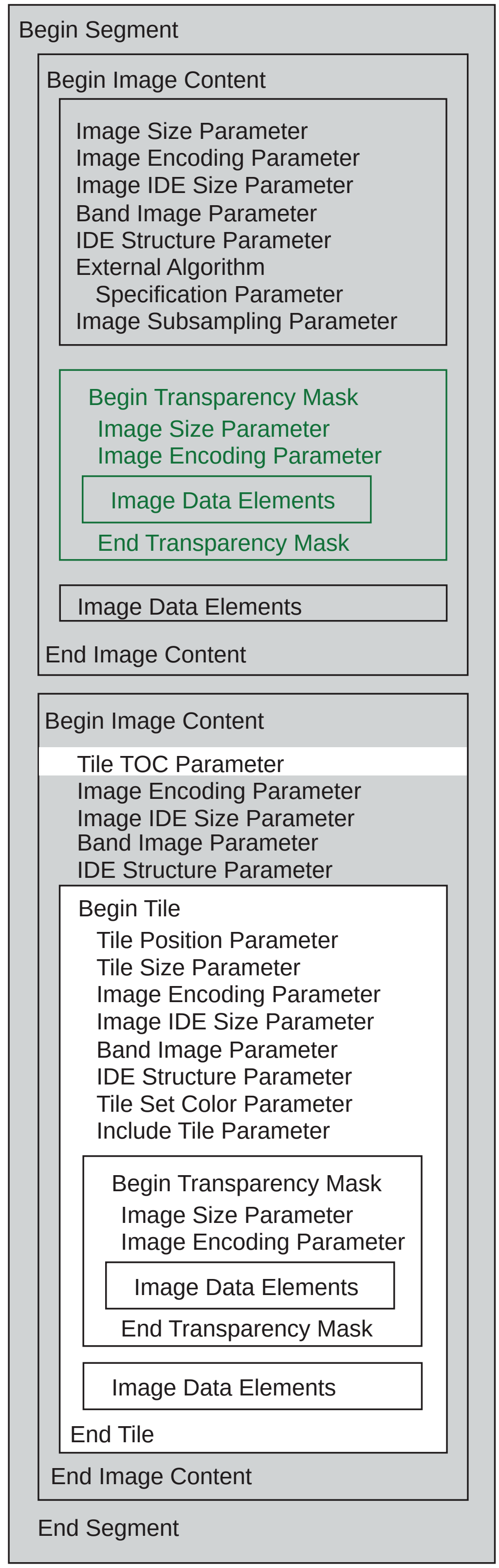
Condition: The Image Subsampling parameter appeared out of sequence or more than once.

EC-CE10 Invalid or unsupported Image Data parameter value

Condition: The Image Subsampling parameter contains an invalid or unsupported value.

Tiles

Tiles are used when different parts of an image are described using different color spaces, resolutions, and compression algorithms. Tiles can also be used as resources (see Appendix C, “IOCA Tile Resource”, on page 151).



Tiles

The tiling scheme used in IOCA has the following features:

- Each Image Content can be either tiled or untiled. In an untiled Image Content, no tiles may appear. Tiled Image Contents are indicated by the presence of the Tile TOC parameter immediately following the Begin Image Content parameter. In a tiled Image Content, no image data elements may appear outside of the tiles.
- Tiles can use different color spaces and compression algorithms.
- Each tile must either have the resolution of the underlying Image Presentation Space, or be subsampled by the same integer factor in both horizontal and vertical dimensions.
- Tiles must be non-overlapping and must also be specified in top-down, left-to-right order.
- Tiles do not have to cover the whole Image Presentation Space. The part of the Image Presentation Space not covered by tiles is treated as background. Tiles must be fully contained in the Image Presentation Space.
- Within tiles, foreground and background are determined based on the color space used.
- A tile can be either a *data tile* (that is, a fully defined tile with all the data present), or a *referencing tile*. A referencing tile contains an invocation, positioning, and merging instruction for a tile resource and is intended to save bandwidth and processing time when processing multiple images that have some areas in common.

Begin Tile

The Begin Tile parameter defines the beginning of a tile.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'8C'	Begin Tile parameter	M
1	UBIN	LENGTH	X'00'	Length of the parameters to follow	M

Notes:

1. In tiled images, all of the image data must be contained in tiles. That is, no Image Data or Band Image Data can appear outside of the sequence delimited by the Begin Tile/End Tile pairs.
2. The Begin Tile Parameter can appear in all of the contexts where Image Data and Band Image Data can appear in non-tiled images.
3. If the Begin Tile Parameter is encountered, the first parameter after the Begin Image Content parameter must be the Tile TOC parameter. Otherwise, exception EC-8C0F occurs.

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-8C0F **Invalid sequence**

Condition: A Begin Tile has appeared out of sequence.

End Tile

The End Tile parameter defines the end of a tile.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'8D'	End Tile parameter	M
1	UBIN	LENGTH	X'00'	Length of the parameters to follow	M

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-8D0F **Invalid sequence**

Condition: An End Tile is missing after a Begin Tile has been encountered, or it appeared out of sequence.

Tile Position

The Tile Position parameter determines the position of the upper-left corner of the tile in the Image Presentation Space.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'B5'	Tile Position parameter	M
1	UBIN	LENGTH	X'08'	Length of the parameters to follow	M
2–5	UBIN	XOFFSET	X'00000000' – X'7FFFFFFF'	Horizontal offset of the tile origin, relative to the presentation space origin	M
6–9	UBIN	YOFFSET	X'00000000' – X'7FFFFFFF'	Vertical offset of the tile origin, relative to the presentation space origin	M

Notes:

1. XOFFSET and YOFFSET are specified in presentation space image points. If subsampling is specified in the Tile Size parameter, it does not apply to XOFFSET and YOFFSET.
2. The upper-left corner of the tile must be contained in the presentation space; that is, XOFFSET and YOFFSET must be less than XSIZE and YSIZE, respectively, as specified in the Image Data Descriptor. For the definition of the Image Data Descriptor, see "Image Data Descriptor (IDD)" on page 154.
3. If the current tile is not the first tile specified, the YOFFSET value must be at least as large as any specified for the previous tiles. If YOFFSET is identical to the previous YOFFSET, XOFFSET must be greater than the previous XOFFSET. This requirement forces the tile order of top down (primary key) and left to right (secondary key). This condition applies only if the Tile TOC parameter does not contain the tile table of contents.

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-B50F **Invalid sequence**

Condition: A Tile Position is missing, or it appeared out of sequence.

EC-B510 **Invalid Tile Position Parameters**

Condition: XOFFSET, YOFFSET, or both are outside of the valid range or outside of the Image Presentation Space.

EC-B511 **Inconsistent Tile Position Parameters**

Condition: One of the following conditions has been encountered:

- Tiles are specified out of order. This exception can occur only if the Tile TOC parameter does not contain the table of contents. If the Tile TOC Parameter does contain the table of contents, the tiles themselves can be specified in any order.
- Offset mismatch: the tile table of contents has been specified, but the XOFFSET or YOFFSET given for this tile does not match the values specified in the Tile Position Parameter.

Tile Size

The Tile Size parameter defines the size and resolution of a tile.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'B6'	Tile Size parameter	M
1	UBIN	LENGTH	X'08' – X'09'	Length of the parameters to follow	M
2–5	UBIN	THSIZE	X'00000000' – X'7FFFFFFF'	Horizontal size in image points, excluding any padding bits in each scan line	M
6–9	UBIN	TVSIZE	X'00000000' – X'7FFFFFFF'	Vertical size in image points, excluding any padding scan lines	M
10	UBIN	RELRES	X'01' – X'02'	Relative resolution of the tile	O

Notes:

1. If RELRES has not been specified, the tile resolution is the same as the resolution of the Image Presentation Space.
2. A RELRES value of 1 means that the tile has the same resolution as the Image Presentation Space. A RELRES value of 2 means the resolution of the tile is half the resolution of the Image Presentation Space. For example, if the Image Presentation Space has a resolution of 600 dpi, a tile with a RELRES of 2 has a resolution of 300 dpi. The default value of RELRES is 1.
3. The tile dimensions THSIZE and TVSIZE are specified in the tile resolution. To get the size of the tile in presentation space points, multiply the THSIZE and TVSIZE by RELRES.
4. The tile must be wholly contained in the presentation space; that is, (XOFFSET + RELRES × THSIZE) must not exceed the XSIZE specified in the Image Data Descriptor and (YOFFSET + RELRES × TVSIZE) must not exceed the YSIZE specified in the Image Data Descriptor.
5. Tiles must be non-overlapping. If X1, Y1, H1, V1, S1 and X2, Y2, H2, V2, S2 describe the offset, size, and subsampling of any two tiles, at least one of the following relationships must hold:
   ```
   X1 + S1 × H1 ≤ X2
   X2 + S2 × H2 ≤ X1
   Y1 + S1 × V1 ≤ Y2
   Y2 + S2 × V2 ≤ Y1
   ```
   Note that, in this example, tiles 1 and 2 are not necessarily sorted. That is, the origin of tile 1 need not be above or left of the origin of tile 2.
6. The JPEG compression algorithm works on 8-by-8-pixel blocks. Depending on the JPEG subsampling (note that this is different from RELRES in Tile Size), the Minimum Coded Units (MCUs) used by JPEG might be larger. The most common MCU size is 16 by 16 pixels. The image must be padded before compression to the MCU boundary and the decompressor discards the padding pixels. To help receivers merge JPEG-compressed tiles efficiently, the tile data must be padded to the left and top to the nearest 8-pixel boundary in the tile resolution, after applying tile subsampling and before compression. After padding on the left and top, the tile is padded as usual on the right and bottom. On decompression, the decompressor discards the right and bottom padding pixels. The receiver then must discard any left and top padding pixels. The number of pad pixels on the left and top can be computed by dividing the XOFFSET and YOFFSET by RELRES×8 and taking the remainder. Note that padding is done in the Image Presentation Space image points, before subsampling. Otherwise, images with odd XOFFSET or YOFFSET could not be aligned.

Example

This example shows how to construct, compress, and decompress a tile with JPEG and RELRES of 2.

Let the area of the image that we wish to use as a tile have the origin of XOFFSET = 21 and YOFFSET = 36. Let the area be 100 presentation space points wide and 211 presentation space points high. Assume that we use no JPEG subsampling. XOFFSET and YOFFSET can be used to indicate the tile origin in the Tile Position parameter. The tile size is set to THSIZE = 50 and TVSIZE = 105.

To compress the data, start at the image point with the horizontal offset of 16 and the vertical offset of 32 in the presentation space. Select the region 112 pixels wide and 224 pixels high. If the presentation space is not large enough, pad at the right and bottom, until these dimensions are reached. Subsample by the factor of two, which yields an image 56 pixels wide and 112 pixels high. Since the image sizes are even multiples of 8 and no JPEG subsampling is desired, the data can be compressed with JPEG without further padding.

To merge the tile into the presentation space, decompress the tile with JPEG. Upsample by a factor of two, yielding a tile that is 112 by 224 pixels. Since XOFFSET is 21, we know that the leftmost five pixels have to be discarded. Similarly, the YOFFSET value of 36 indicates a top pad of 4 pixels. From the THSIZE and TVSIZE, after upsampling, the actual tile is 100 pixels wide and 210 pixels high. Thus, the left 5 pixels, top 4 pixels, right 7 pixels and bottom 10 pixels are discarded to yield the unpadded tile. Note that a scanline on the bottom was lost due to downsampling.

In this example, the right and bottom are padded before the data is passed to the compressor. If you do not pad first, the compressor does the padding and the decompressor strips it. Manual padding, however, allows control over how the padding is done. If the tiles are constructed so that a single continuous tone image is broken into multiple adjoining tiles, selecting the actual image data for padding eliminates edge artifacts when the tiles are joined.

If the compressor allows the caller to specify the padding data, manual padding is not necessary. Note that manual padding also assumes that the receiver checks the image returned by the decompressor and discards not only the top and left pads, but also the bottom and right pads. The receiver can compute the pad sizes from the values of RELRES, XOFFSET, YOFFSET, THSIZE, and TVSIZE.

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-A902 A portion of the extracted image is written outside the Image Presentation Space

Condition: The tile is not wholly contained in the Image Presentation Space.

EC-B60F Invalid sequence

Condition: A Tile Size is missing, or it appeared out of sequence.

EC-B610 Invalid Tile Size parameters

Condition: The tile size or relative resolution are outside valid ranges or are invalid for the function set.

EC-B611 Inconsistent Tile Size parameters

Condition: At least one of the following conditions is true:

- The tile overlaps a previously specified tile.
- Subsampling mismatch: the RELRES value in the table of contents does not match the RELRES value in the Tile Size parameter.

Tile Size

- Size mismatch: the THSIZE or TVSIZE specified in the table of contents does not match the corresponding value in the Tile Size parameter.

Tile Set Color

The Tile Set Color parameter specifies the color used to paint significant pels of a bilevel tile.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'B7'	Tile Set Color parameter	M
1	UBIN	LENGTH	X'0A' – X'0C'	Length of the parameters to follow	M
2	CODE	CSPACE	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	M
3–5			X'000000'	Reserved; should be zero	M
6	UBIN	SIZE1	X'01' – X'08', X'10'	Number of bits/IDE for component 1; see color space definitions	M
7	UBIN	SIZE2	X'00' – X'08'	Number of bits/IDE for component 2; see color space definitions	M
8	UBIN	SIZE3	X'00' – X'08'	Number of bits/IDE for component 3; see color space definitions	M
9	UBIN	SIZE4	X'00' – X'08'	Number of bits/IDE for component 4; see color space definitions	M
10–*n*		Color		Color specification; see "Tile Set Color Semantics" on page 59 for details	M

Notes:

1. The Tile Set Color Parameter serves two purposes. One purpose is to define the color of the significant pels in a bilevel tile. The other is to paint the whole tile with the specified color. In the second use, the tile does not contain any image data.
2. If the Tile Set Color Parameter is present, the significant image pels are painted with the specified color. Insignificant image pels are treated according to the rules for bilevel images.
3. If all pels are significant (that is, if the whole tile is to be painted), the compression algorithm must be set to Solid Fill Rectangle. In this case (solid fill), Image Data and Band Image Data cannot appear, or the exceptions EC-920F and EC-9C0F occur.
4. The Image Encoding parameter and IDE Structure parameter can appear for the tile, but must specify a bilevel image (the IDE size must be 1). The color space given in the IDE Structure parameter must be either YCbCr or YCrCb.

Tile Set Color Semantics

CSPACE Is a code that defines the color space and the encoding for the color specification.

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. SIZE1, SIZE2, and SIZE3 are nonzero and define the number of bits used to specify each component. SIZE4 is reserved and should be set to zero.

The intensity range for the R,G,B components is 0 to 1, which is mapped to the binary value range 0 to (2^{SIZEN} - 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* D_{6500} or simply *D65*. The R,G,B components are assumed to be gamma-corrected (non-linear) 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. SIZE1, SIZE2, SIZE3, and SIZE4 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^{SIZEN} - 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. SIZE1 = 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. SIZE2 = 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-coverage)%. If the sum of percent coverage plus percent shading is less than 100%, the remaining coverage is achieved with color of medium. SIZE3 = 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.

SIZE4 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' to X'00FF' are reserved for existing highlight color applications and devices. That is, indexed color values in the range X'0000' to 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' to 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. SIZE1, SIZE2, and SIZE3 are nonzero and define the number of bits used to specify each component. SIZE4 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^{SIZE1} - 1). The range for the a and b components is -127 to +127, which is mapped to the binary range -($2^{\text{SIZE}N\text{-1}}$ - 1) to +($2^{\text{SIZE}N\text{-1}}$ - 1), where N=2,3.

For color fidelity, 8-bit encoding should be used for each component, that is, SIZE1, SIZE2, and SIZE3 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*. SIZE1 = X'10' and defines the number of bits used to specify component 1. SIZE2, SIZE3, and SIZE4 are reserved and should be set to zero. This is a device-dependent color space.

All others Reserved

SIZE1 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 SIZE1 = X'06', the first color component has two padding bits.

SIZE2 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.

SIZE3 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.

SIZE4 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.

Color Specifies the color value in the defined format and encoding. Note that the number of bytes specified for this parameter depends on the color space. For example, when using 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 self-defining field length is valid.

To illustrate the syntax for the color value specified in the Color field, the following table shows examples of various ways that a light-green color can be specified. Note that the light-green color value is approximated in each of the color spaces.

CSPACE	SIZE1	SIZE2	SIZE3	SIZE4	Color
RGB	X'08'	X'08'	X'08'	N/A	X'00B761'
RGB	X'05'	X'05'	X'05'	N/A	X'00160B'
CMYK	X'08'	X'08'	X'08'	X'08'	X'FF489E00'
CMYK	X'01'	X'02'	X'04'	X'08'	X'01010900'
Highlight (see note)	X'10'	X'08'	X'00'	N/A	X'000264'
CIELAB	X'08'	X'08'	X'08'	N/A	X'A8CC21'
Standard OCA	X'10'	N/A	N/A	N/A	X'0004'
Note: This example assumes that the light-green color is loaded in the printer as highlight color X'0002'.					

Architecture Note: 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.

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-B70F **Invalid sequence**

Condition: The Tile Set Color parameter appears out of sequence, or more than once within a single tile.

EC-B710 **Invalid Tile Set Color parameter**

Condition: At least one of the following values is not valid:
- CSPACE
- SIZE
- Color

EC-B711 **Inconsistent Tile Set Color parameter**

Condition: The IDESZ field in the IDE Size parameter has a value other than 1, or the color space specified in the IDE Structure parameter is not YCbCr or YCrCb.

Include Tile

The Include Tile parameter defines the tile as a referencing tile. The tile does not contain any image data, except possibly a Transparency Mask, and is instead read from the referenced resource.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0–1	CODE	ID	X'FEB8'	Include Tile parameter	M
2–3	UBIN	LENGTH	X'0004'	Length of the parameters to follow	M
4–7	CODE	TIRID	X'00000000' – X'FFFFFFFF'	Tile Resource local identifier	M

Notes:

1. If a tile contains the Include Tile parameter, it must contain a Tile Position parameter and can also contain a transparency mask. Any other parameters cause one of the Invalid Sequence (EC-*xx*0F) exceptions to be raised.
2. The Tile Position parameter in the included tile is ignored. The Tile Position parameter specified in the referencing tile is used instead.
3. If a referencing tile contains a transparency mask and the included tile also contains a transparency mask, the two masks are combined by using the logical AND operation. That is, a pixel is foreground if it is defined as foreground in both masks.
4. Tile resources do not contain any references to the Image Presentation Space. Each included tile is interpreted according to the current Image Presentation Space.
5. Except for the Tile Position and transparency mask, the included tile is treated exactly as if it was specified entirely locally. All defaulting and override rules for tile data apply.
6. The included tile must not contain another Include Tile parameter (that is, no nested references are allowed). There are no other constraints on the tile content.
7. Any other errors, such as the tile not being contained in the Image Presentation Space, are treated by raising the same exceptions as if the tile were specified locally.

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-B80F **Invalid sequence**

Condition: An Include Tile parameter has appeared out of sequence or more than once.

EC-B811 **Inconsistent Include Tile parameter**

Condition: The included tile resource contains an Include Tile parameter.

Tile TOC

The Tile Table of Contents (TOC) parameter defines the image as a tiled image. Optionally, it also defines the size and position of each tile.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0–1	CODE	ID	X'FEBB'	Tile TOC parameter	M
2–3	UBIN	LENGTH	X'0002' – X'7FFF'	Length of the parameters to follow; this value must be a multiple of 26, plus 2	M
4–5			X'0000'	Reserved; should be zero	M
Zero or more repeating groups, with tile table of contents. If any TOC entries are present, then an entry for each tile must be present. The groups have the following format:					
0–3	UBIN	XOFFSET	X'00000000' – X'7FFFFFFF'	Horizontal offset of the tile origin, relative to the image origin	M
4–7	UBIN	YOFFSET	X'00000000' – X'7FFFFFFF'	Vertical offset of the tile origin, relative to the image origin	M
8–11	UBIN	THSIZE	X'00000000' – X'7FFFFFFF'	Horizontal size in image points, excluding any padding bits in each scan line	M
12–15	UBIN	TVSIZE	X'00000000' – X'7FFFFFFF'	Vertical size in image points, excluding any padding scan lines	M
16	UBIN	RELRES	X'01' – X'02'	Relative resolution of the tile	M
17	CODE	COMPR		Compression algorithm; see Image Encoding parameter	M
18–25	UBIN	DATAPOS	Any	Offset, in bytes, from the start of the Begin Segment parameter of the current image, to the start of the Begin Tile parameter starting the tile	M

Notes:

1. Tiles in the table of contents must be specified in top-down, left-to-right order. If the table of contents is specified, the tiles themselves can be specified in any order (that is, the order restriction described for the Tile Position parameter is lifted).
2. The Tile TOC parameter must appear immediately after the Begin Image Content parameter; otherwise exception EC-BB0F occurs. If a Begin Tile parameter is encountered without a Tile TOC Parameter having been specified, exception EC-8C0F occurs.
3. If the image contains the Tile TOC parameter, no Image Data or Band Image Data may appear outside of the tiles (Begin Tile/End Tile pairs). Otherwise, exception EC-9201 (Image Data) or EC-9C01 (Band Image Data) occurs.
4. The presence of the Tile TOC parameter does not require that any tiles be actually specified. An empty image (no tiles present) is valid and all the image points are treated as background.
5. In terms of the DATAPOS, the first byte of the Begin Segment parameter has the offset zero.
6. If the Tile TOC parameter contains the table of contents, the values in the table of contents entry for each tile must match the values specified in the Tile Position parameter and Tile Size parameter for that tile. Otherwise, exception EC-B511 or EC-B611, respectively, occurs when inconsistent values are encountered in the Tile Position parameter and the Tile Size parameter.

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-BB0F Invalid sequence

Condition: A Tile TOC parameter appeared somewhere other than immediately after the Begin Image Content parameter or appeared more than once.

EC-BB10 Invalid Tile TOC values

Condition: One or more values specified in the Tile TOC Parameter is outside of the valid range.

EC-BB11 Inconsistent Tile TOC parameter

Condition: The parameter contains the tile table of contents and one or more of the following conditions is true:

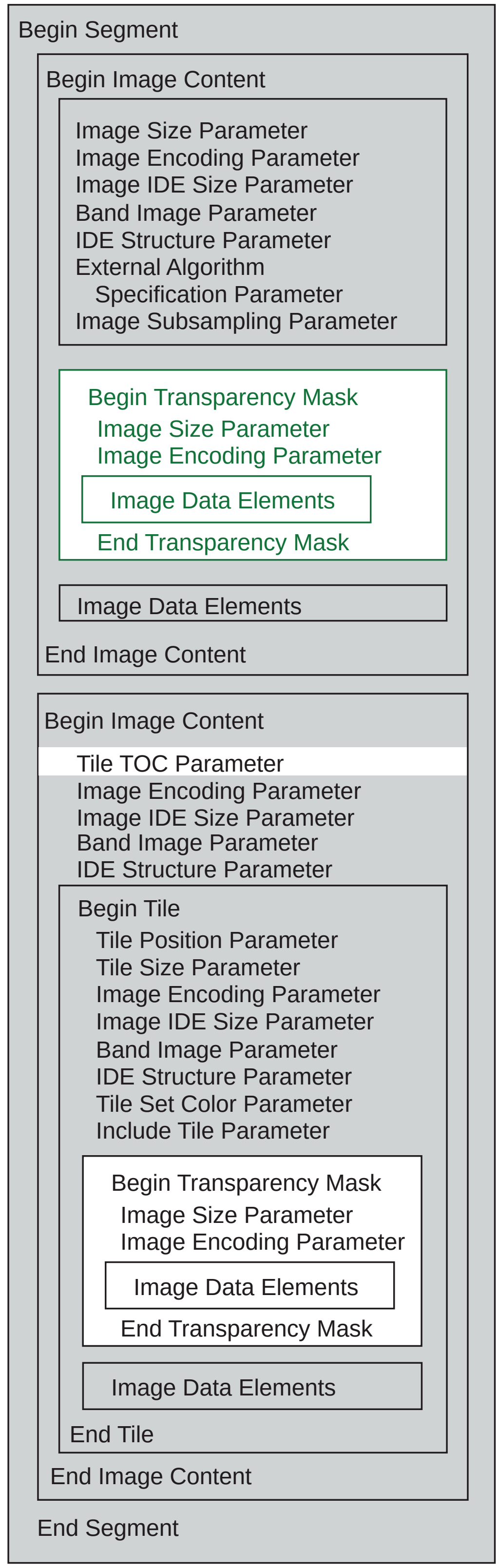
- Not all tiles are listed in the table of contents, even though the table of contents contains at least one tile.
- The table of contents lists a non-existent tile.
- Invalid tile order: two or more tiles in the table of contents have identical sort keys, or the sort keys are out of sequence.

  Note: The primary sort key is YOFFSET. The secondary sort key is XOFFSET.
- Invalid DATAPOS: the specified offset for one or more tiles does not point to a position where a Begin Tile parameter starts.

Transparency Masks

Transparency masks are bilevel images that are used to turn some image points into background.

Function Sets 14, 45, and 48 are currently the only function sets that include transparency masks. For more information on function sets, see "Function Sets" on page 83.



The transparency mask is a restricted bilevel image in the sense that it must have the same size in pels as the underlying image or tile. If the transparency mask is specified within a tile, the mask has the same resolution as the presentation space; that is, the relative resolution specified in RELRES does not apply. Transparency mask dimensions are carried explicitly using the Image Size parameter. These dimensions must match dimensions obtained by multiplying the tile dimensions by RELRES; otherwise exception EC-9411 occurs.

The transparency mask, if present, must immediately precede the first Image Data or Band Image Data. Images that are not tiled can have at most one transparency mask. In tiled images, the transparency masks must be contained in tiles and each tile can contain at most one transparency mask. Note that tiled images can thus contain multiple transparency masks, each contained in and applying to a different tile.

If the transparency mask is specified in a tile that contains the Include Tile parameter, it must be specified after both the Tile Position and Include Tile parameters.

Tiles using the Include Tile parameter to invoke tile resources can have two transparency masks, one in the calling tile and one in the resource tile itself. The two transparency masks are combined using the logical AND operation; that is, an image point is in the foreground if it is in foreground in both masks. In other words, the caller can declare some of the resource image foreground points as background, but not the reverse.

The transparency mask has a point for each underlying image or presentation space point. If the transparency mask has been specified, the receiver should apply it on a point by point basis. If, at an image point, the mask contains B'0', the point is treated as background. Otherwise, if the mask contains B'1', the image point is treated according to the rules of the current color space, as if no transparency mask has been specified.

Table 5. Transparency Mask Structure

	X'8E'	Begin Transparency Mask parameter		
	X'94'	Image Size parameter		
[	X'95'	Image Encoding parameter		]
	X'FE92'	Image Data	(S)	
	X'8F'	End Transparency Mask parameter		

Transparency masks can be described using the following parameters:
- Begin Transparency Mask
- Image Size
- Image Encoding
- Image Data
- End Transparency Mask

Notes:

1. All recording algorithms and compression algorithms allowed for bilevel images in the IOCA Function Set specified for the image can be used. If the datastream does not specify the function set, any architecturally valid Image Encoding parameter values can be used, except Solid Fill Rectangle. The Solid Fill Rectangle algorithm is not needed, since omitting the transparency mask achieves the same effect as setting all the transparency mask image points to 1. Completely removing the image achieves the same effect as setting all transparency mask image points to 0.
2. If the Image Encoding parameter is missing, the default encoding (no compression and RIDIC) applies.

Begin Transparency Mask

The Begin Transparency Mask defines the beginning of the transparency mask.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'8E'	Begin Transparency Mask parameter	M
1	UBIN	LENGTH	X'00'	Length of the parameters to follow	M

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-8E0F **Invalid sequence**

Condition: A Begin Transparency Mask has appeared out of sequence or more than once within a tile or a non-tiled image.

End Transparency Mask

The End Transparency Mask defines the end of a transparency mask.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'8F'	End Transparency Mask parameter	M
1	UBIN	LENGTH	X'00'	Length of the parameters to follow	M

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-8F0F Invalid sequence

Condition: An End Transparency Mask is missing after a Begin Transparency Mask has been encountered, or it appeared out of sequence.

Image Data Elements

This section describes the parameters used to carry the Image Data Elements.

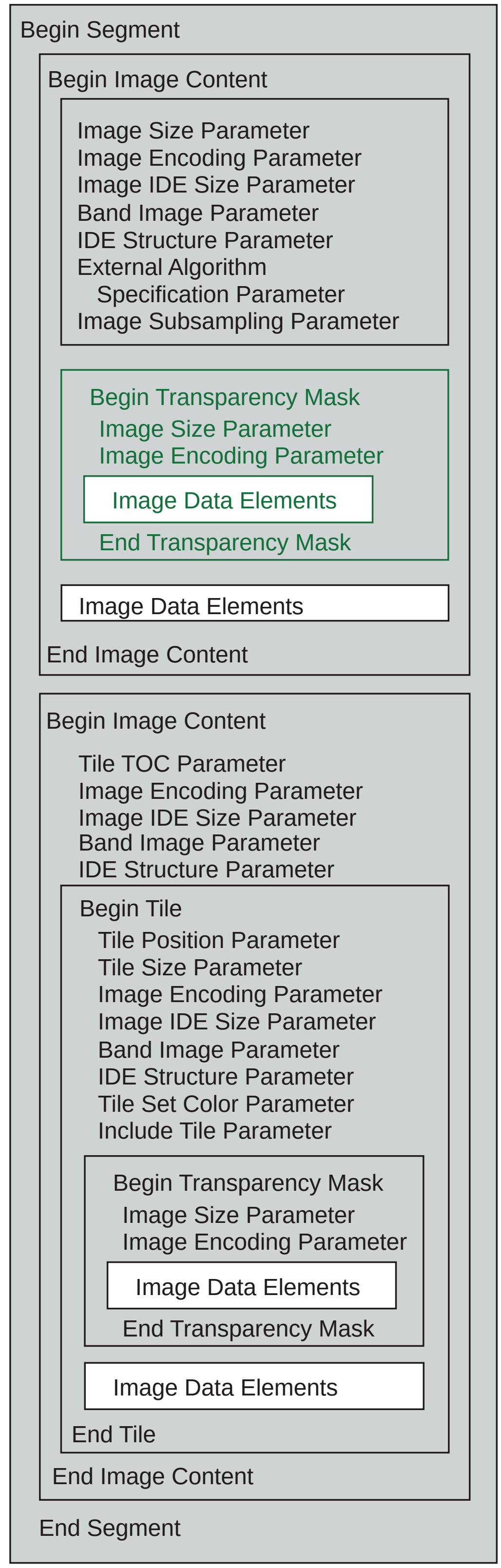


Image Data

The Image Data is an element of the Image Content, and is a set of IDEs. Multiple Image Data self-defining fields of the same type can be contained in a single untiled Image Content, a single tile, or a single transparency mask.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0–1	CODE	ID	X'FE92'	Image Data parameter	M
2–3	UBIN	LENGTH	X'0001' – X'FFFF'	Length of the image data to follow	M
4–*n*		DATA	Any	Image Data Elements. The data in this self-defining field is recorded, compressed, and ordered as specified by the Image Encoding parameter. For some function sets, the data can also be subsampled as described by the Image Subsampling parameter.	M

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-9201 Invalid existence of Image Data

Condition: Image Data should not be present, as in the case when the image data is in bands.

EC-920F Invalid sequence

Condition: Image Data is missing, or it appeared out of sequence.

Band Image Data

The Band Image Data is an element of the Image Content. It is a set of IDEs typically having similar attributes to each other.

Band Image Data must appear within an Image Content or a tile for each band defined by the Band Image parameter. The bands must appear in the sequential order of their band numbers. The Band Image parameter must exist in the Image Content if this parameter is used. See "Band Image" on page 38 for more detail.

If the data for a particular band exceeds the length of a single Band Image Data, the remaining data is contained in one or more Band Image Data parameters having the same band number, and following in sequence.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0–1	CODE	ID	X'FE9C'	Band Image Data parameter	M
2–3	UBIN	LENGTH	X'0003' – X'FFFF'	Length of the parameters to follow	M
4	CODE	BANDNUM	X'01' – X'FD'	Band number	M
5–6			X'0000'	Reserved; should be zero	M
7–*n*		DATA	Any	Image Data Elements for the band. The data in this self-defining field is recorded, compressed, and ordered as specified by the Image Encoding parameter. For some function sets, the data can also be subsampled as described by the Image Subsampling parameter.	O

Notes:

1. If the Band Image Data contains no data (the length is X'0003') for a certain band, all the uncompressed image data elements in the band are zero. For such a band, only a single Band Image Data parameter can appear; otherwise exception EC-9C0F occurs.
2. If the color space of the image is CMYK and the bands 1, 2, and 3 (cyan, magenta, and yellow) contain no data, the receivers should color-manage the image as monochrome.

Exception Conditions

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

EC-0003 Invalid length

Condition: The LENGTH value is not in the valid range.

EC-0004 Invalid parameter value

Condition: The BANDNUM value is not in the valid range.

EC-9C01 Invalid existence of Band Image Data

Condition: Band Image Data should not be present, as in the case when the Image Data is not in bands.

EC-9C0F Invalid sequence

Condition: Band Image Data is missing, or it appeared out of sequence.

EC-9C17 Invalid number/sequence of Band Image Data

Condition: The band numbers of a band image do not appear for the number of bands defined in the Band Image parameter, or do not appear in succeeding order.

Chapter 6. Exception Conditions and Actions

This chapter outlines the exception conditions and exception actions for IOCA, and summarizes:
- Exception conditions causing the common standard action
- Exception conditions causing unique standard actions

An exception condition is either mandatory or optional. If a function is supported and a mandatory exception condition for the function is detected, the process must notify the controlling environment. If an optional exception condition for the function is detected, the process may or may not need to notify the controlling environment.

The table in "Mandatory or Optional Exception Conditions" on page 81 summarizes for each IOCA exception condition whether it is mandatory or optional. Also shown in the table is whether the two primary IOCA controlling environments–MO:DCA and IPDS–would consider the exception condition to be mandatory or optional.

Exception Conditions

Exception conditions include violations of the following:
- Syntax
- Parameter value
- Self-defining field sequence

Exceptions are represented in the following format:
eee-xxxx

where:

eee identifies the exception group. The exception group can be one of the following:
EC Image Segment exception

xxxx is the exception code (two-byte hexadecimal value).

There are two types of exception conditions:
- Those that cause the common standard action
- Those that cause unique standard actions

The exception conditions are summarized in the following sections. Unique exception conditions and actions that are related to a specific element are described in the corresponding section for the element.

Image Segment Exception Conditions

This exception occurs when a violation of format, parameter, or sequence of execution to this architecture is detected in the Image Segment.

The exception is represented in the following format:
EC-*xxxx*

where:

EC identifies an Image Segment exception condition.

xxxx is the Image Segment exception condition code (two-byte hexadecimal value).

The following Image Segment exception conditions are defined:
- Exception conditions that cause the common standard action

- Exception conditions that cause unique standard actions

The IOCA Process Model recovers the exception condition according to the severity of the exception. The severity of an exception depends on the Image Data parameter or the Image Data.

If an exception action is defined in the corresponding section, the action is taken first, and the exception condition is kept until it is reported to the controlling environment.

If the action is not defined in the corresponding section, the rest of the Image Segment is not processed and the exception condition is reported immediately to the controlling environment.

Exception Actions

The IOCA Process Model responds with an exception action when it detects an exception condition.

An exception condition prompts either of the following kinds of actions:
- An exception action defined by IOCA
- An alternative action that is defined outside the IOCA Process Model

The controlling environment governs which way the IOCA Process Model responds to the exception conditions. For example, the IPDS architecture has a command to specify whether the IPDS-defined page continuation action or the IOCA-defined exception action is to be taken.

IOCA Process Model Actions

The IOCA Process Model recovers the exception condition according to the severity of the exception. The severity depends on the condition where the exception is detected.

The exception conditions are reset after the controlling environment has been notified.

Unique Standard Actions

If a unique exception action is defined for the exception condition (such as for EC-9401 and EC-9511), the IOCA Process Model:
- Notifies the controlling environment of the condition
- Performs the defined unique action

Common Standard Action

If no unique exception action is defined, the IOCA Process Model:
- Notifies the controlling environment of the condition
- Does not process the rest of the Image Segment

The exception conditions are reset after the controlling environment has been notified.

Exception Conditions Causing the Common Standard Action

EC-0001 Invalid or unsupported code within an Image Segment

Condition: An invalid or unsupported self-defining field is detected within the Image Segment.

EC-0002 Retired

Condition: Retired. See Appendix G, “Retired Architecture”, on page 175 for information about this retired exception condition.

EC-0003 The LENGTH value is not in the valid range

Condition: The LENGTH value is not in the valid range.

EC-0004 Invalid parameter value

Condition: A parameter value is not in the valid range.

Note: In cases where this exception is being generated for self-defining fields for which an EC-*xx*10 exception is available, it is recommended that IOCA receivers generate the EC-*xx*10 exception instead of exception EC-0004. For example, for the IDE Size parameter, EC-9610 would be generated and not EC-0004.

EC-0005 Invalid length

Condition: The LENGTH value is not in the valid, function-set specified range. EC-0005 is optional—IOCA receivers can generate EC-0003 instead of EC-0005.

EC-*xx*0F Invalid sequence

Condition: The sequence of self-defining fields is not correct within an Image Segment. This exception condition is also raised when a mandatory or necessary self-defining field is missing, or when a self-defining field other than Image Data, or Band Image Data, appears more than once in an Image Segment. Value *xx* in the exception code depends on the parameter in which the exception occurred. The parameter code is placed in this *xx* position.

For example:
- **EC-700F** for Begin Segment
- **EC-710F** for End Segment (see Note below)
- **EC-8C0F** for Begin Tile
- **EC-8D0F** for End Tile
- **EC-8E0F** for Begin Transparency Mask
- **EC-8F0F** for End Transparency Mask
- **EC-910F** for Begin Image Content
- **EC-920F** for Image Data (see Note below)
- **EC-930F** for End Image Content
- **EC-940F** for Image Size
- **EC-950F** for Image Encoding
- **EC-960F** for IDE Size
- **EC-970F** (Retired)
- **EC-980F** for Band Image
- **EC-9B0F** for IDE Structure
- **EC-9C0F** for Band Image Data (see Note below)
- **EC-9F0F** for External Algorithm Specification
- **EC-B50F** for Tile Position
- **EC-B60F** for Tile Size
- **EC-B70F** for Tile Set Color
- **EC-B80F** for Include Tile
- **EC-BB0F** for Tile TOC
- **EC-CE0F** for Image Subsampling

Note: This exception condition is not raised when the indicated self-defining field appears more than once.

EC-*xx*10 Invalid or unsupported Image Data parameter value

Condition: The specified value for a parameter is not valid in the architecture or function sets, or is not supported by the implementation. Value *xx* in the exception code depends on the parameter in which the exception occurred. The parameter code is placed in this *xx* position.

For example:
- **EC-9410** for Image Size
- **EC-9510** for Image Encoding
- **EC-9610** for IDE Size
- **EC-9710** (Retired)
- **EC-9810** for Band Image
- **EC-9B10** for IDE Structure
- **EC-9F10** for External Algorithm Specification
- **EC-B510** for Tile Position
- **EC-B610** for Tile Size
- **EC-B710** for Tile Set Color
- **EC-BB10** for Tile TOC
- **EC-CE10** for Image Subsampling

Note: Some controlling environments, such as IPDS, define an exception for an invalid value in the Set Extended Bilevel Image Color self-defining field. Therefore, exception condition EC-F410 is reserved in IOCA for this use.

EC-*xx*11 Inconsistent Image Data parameters, or inconsistent Image Data parameter and Image Data

Condition: The specified value for a parameter is not consistent with a value specified in another parameter or with the image data following it. Value *xx* in the exception code depends on the parameter in which the exception occurred. The parameter code is placed in this *xx* position.

For example:
- **EC-9411** for Image Size
- **EC-9611** for IDE Size
- **EC-9F11** for External Algorithm Specification
- **EC-B511** for Tile Position
- **EC-B611** for Tile Size
- **EC-B711** for Tile Set Color
- **EC-B811** for Include Tile
- **EC-BB11** for Tile TOC

EC-9201 Invalid existence of Image Data

Condition: Image Data should not be present, as in the case when the image data is in bands.

EC-9801 Invalid Band Image parameter and Image Subsampling parameter coexistence

Condition: In some function sets, the Band Image parameter and the Image Subsampling parameter cannot coexist in the same Image Content.

EC-9814 Invalid number of bands and bit counts

Condition: The number of BITCNT parameters is not equal to the BCOUNT in the Band Image parameter.

EC-9815 Invalid IDE size

Condition: The IDE size, determined by the Band Image parameter, does not match the IDE Size parameter.

EC-9B18 Invalid IDE Structure parameter

Condition: One of the following conditions occurs in the IDE Structure parameter:

- The sum of the SIZE values does not match the IDE size specified by the IDE Size parameter.
- The color model is CMYK and SIZE4 is missing.
- SIZE4 is present and the color model is not CMYK or nColor.
- More than four SIZE parameters are present and the color model is not nColor.
- The color model is nColor and the number of SIZE parameters is not equal to the second half of the FORMAT byte.

EC-9C01 Invalid existence of Band Image Data

Condition: Band Image Data should not be present, as in the case when the image data is not in bands.

EC-9C17 Invalid number/sequence of Band Image Data

Condition: The band numbers in a Band Image Data do not appear for the number of bands defined in the Band Image parameter, or do not appear in succeeding order.

EC-9F01 Missing External Algorithm Specification parameter or Image Encoding parameter

Condition: An External Algorithm Specification parameter exists without a corresponding Image Encoding parameter, or an Image Encoding parameter exists that requires an External Algorithm Specification parameter that cannot be found.

EC-CE01 Invalid Band Image parameter and Image Subsampling parameter coexistence

Condition: In some function sets, the Band Image parameter and the Image Subsampling parameter cannot coexist in the same Image Content.

Exception Conditions Causing Unique Standard Actions

EC-9401 Inconsistent Image Size parameter value and Image Data

Condition: The size detected in the image data is different from the HSIZE or VSIZE value of the Image Size parameter.

System action: The size detected from the image data is used.

EC-9511 Inconsistent Image Data parameters, or inconsistent Image Data parameter and Image Data

Condition: The decoder encountered one of the following conditions when decompressing the image data:

- The image data is not encoded according to the compression or recording algorithm specified in the Image Encoding parameter.
- The image data cannot be decoded successfully using the size values specified in the Image Size parameter. This condition applies to compression or recording algorithms that do not permit the image size to be encoded in the image data.
- The image data is not in complete accordance with the compression algorithm specified in the Image Encoding parameter.
- The image data is encoded using the algorithm specified in the Image Encoding parameter, but uses a function of the algorithm that is unsupported by the receiver.

System action: Receivers should attempt to present or make use of all successfully decompressed image data. Note that the resulting partial image might differ from the original image.

EC-A902 Write outside of the Image Presentation Space

Condition: A portion of the extracted image is written outside the Image Presentation Space.

System action: The portion inside the Image Presentation Space is presented, and the rest is discarded.

Mandatory or Optional Exception Conditions

Exception	IOCA	MO:DCA	IPDS
EC-0001	Mandatory	Same as IOCA	Same as IOCA
EC-0003	Mandatory	Same as IOCA	Same as IOCA
EC-0004	Mandatory	Same as IOCA	Same as IOCA
EC-0005	Optional*	Same as IOCA	Same as IOCA
EC-*xx*0F	Mandatory	Same as IOCA	Same as IOCA
EC-*xx*10	Mandatory	Same as IOCA	Same as IOCA
EC-*xx*11	Mandatory	Same as IOCA	Same as IOCA
EC-9201	Mandatory	Same as IOCA	Same as IOCA
EC-9401	Mandatory	Same as IOCA	Same as IOCA
EC-9801	Mandatory	Same as IOCA	Same as IOCA
EC-9814	Mandatory	Same as IOCA	Same as IOCA
EC-9815	Mandatory	Same as IOCA	Same as IOCA
EC-9B18	Mandatory	Same as IOCA	Same as IOCA
EC-9C01	Mandatory	Same as IOCA	Same as IOCA
EC-9C17	Mandatory	Same as IOCA	Same as IOCA
EC-9F01	Mandatory	Same as IOCA	Same as IOCA
EC-A902	Mandatory	Same as IOCA	Same as IOCA
EC-CE01	Mandatory	Same as IOCA	Same as IOCA
*Receiver can generate EC-0003 instead of EC-0005.			

Chapter 7. Compliance

This chapter:

- Describes the concept of IOCA function sets
- Lists the product compliance rules
- Defines IOCA Function Sets FS10, FS11, FS14, FS40, FS42, FS45, and FS48

Function Sets

A *function set* is a set of self-defining fields that describes an Image Object. Specifically, it is a definition of the Image Segment: which parameters the Image Segment should consist of, and what values each parameter should have. The Image Object described in the function set can thus be processed in different controlling environments.

Each function set has an identification. With that identification, products determine the level of support they must provide to generate or receive IOCA Image Objects.

Table 6. Function Set Identification

ID	Description	Function Sets Currently Defined
0*x*	Stand-alone	
1*x*	Carried by presentation-level data stream, non-tiled	FS10, FS11, FS14
2*x*	Library/resource	FS20 (Retired)
3*x*	Reserved	
4*x*	Carried by presentation-level data stream, tiled	FS40, FS42, FS45, FS48
Note: *x* is generally assigned in ascending numerical order from zero.		

IOCA defines seven function sets: FS10, FS11, FS14, FS40, FS42, FS45, and FS48.

- Function Set 10 is intended for bilevel images.
- Function Set 11 covers bilevel, grayscale, and color images.
- Function Set 14 covers bilevel, grayscale, and color images that allow use of transparency masks.
- Function Set 40 covers tiled bilevel images.
- Function Set 42 covers tiled bilevel images and tiled CMYK images with one bit per spot (SIZE1=SIZE2=SIZE3=SIZE4=1, IDESZ=4).
- Function Set 45 carries tiled bilevel and CMYK images. CMYK images can be either one or eight bits per spot (IDESZ=4 or IDESZ=32).
- Function Set 48 carries tiled bilevel, CMYK, and nColor images. CMYK images can be either one or eight bits per spot (IDESZ=4 or IDESZ=32). nColor images are eight bits per spot (IDESZ is a multiple of eight).

Note: Function Set 20 is used only in MO:DCA-L and has been retired. See Appendix G, “Retired Architecture”, on page 175.

Function Set 14 is a superset of Function Set 11. Function Set 11 is a superset of Function Set 10. Function Set 48 is a superset of Function Set 45. Function Set 45 is a superset of Function Set 42. Function Set 42 is a superset of Function Set 40. There are no other relationships among the function sets.

Function Sets

Products that conform to an IOCA function set must meet the following criteria:

- Products that generate IOCA Image Objects can only use the IOCA self-defining fields and parameter values that are defined in the corresponding Function Set definition.
- Products that receive IOCA Image Objects must be capable of accepting any IOCA Image Object that conforms to the corresponding Function Set definition.

The following sections show the self-defining fields that each function set contains and the acceptable values for each field.

IOCA Function Set 10 (IOCA FS10)

Function Set 10 describes bilevel images. This function set is carried by the MO:DCA and IPDS controlling environments. The permissible parameter groupings in FS10 are defined as follows:

Table 7. Function Set 10 Structure

		X'70'	Begin Segment parameter		
		X'91'	Begin Image Content parameter		
+		X'94'	Image Size parameter		
+	[	X'95'	Image Encoding parameter		]
+	[	X'96'	IDE Size parameter		]
+	[	X'97'	Retired (Image LUT-ID parameter)		]
		X'FE92'	Image Data	(S)	
		X'93'	End Image Content parameter		
		X'71'	End Segment parameter		

The self-defining fields and values acceptable for FS10 are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Image Size	ID (1)	X'94'	
	LENGTH (1)	X'09'	
	UNITBASE (1)	X'00' – X'02'	
	HRESOL (2)	X'0000' – X'7FFF'	
	VRESOL (2)	X'0000' – X'7FFF'	
	HSIZE (2)	X'0000' – X'7FFF'	
	VSIZE (2)	X'0000' – X'7FFF'	
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02'	
	COMPRID (1)	X'01', X'03', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No compression **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01'	RIDIC
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01'	1bit/IDE
Retired	RESERVED (3)	X'970100'	Retired Image LUT-ID parameter

Function Set 10

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs (see Note)
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	

Note: IDE value 0 represents an *insignificant* image point, and 1 represents a *significant* image point. The controlling environment determines how to interpret each value.

IOCA Function Set 11 (IOCA FS11)

Function Set 11 is a superset of Function Set 10, and describes bilevel, grayscale, and color images. This function set is carried by the MO:DCA and IPDS controlling environments. The permissible parameter groupings in FS11 are defined as follows:

Table 8. Function Set 11 Structure

		X'70'	Begin Segment parameter		
		X'91'	Begin Image Content parameter		
+		X'94'	Image Size parameter		
+	[	X'95'	Image Encoding parameter		]
+	[	X'96'	IDE Size parameter		]
+	[	X'97'	Retired (Image LUT-ID parameter)		]
+	[	X'98'	Band Image parameter		]
+	[	X'9B'	IDE Structure parameter		]
+	[	X'9F'	External Algorithm Specification parameter		]
+	[	X'FECE'	Image Subsampling parameter		]
			Image Data or Band Image Data	(S)	
		X'93'	End Image Content parameter		
		X'71'	End Segment parameter		

Note: The External Algorithm Specification parameter is part of FS11, but in IOCA is no longer required for JPEG compression, as described in Note 2 in the description of the Image Encoding parameter on page 35. Thus, an FS11 receiver can ignore the External Algorithm Specification parameter if desired.

The self-defining fields and values acceptable for FS11 are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
		Initial parameters:	
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Image Size	ID (1)	X'94'	
	LENGTH (1)	X'09'	
	UNITBASE (1)	X'00' – X'02'	
	HRESOL (2)	X'0000' – X'7FFF'	
	VRESOL (2)	X'0000' – X'7FFF'	
	HSIZE (2)	X'0000' – X'7FFF'	
	VSIZE (2)	X'0000' – X'7FFF'	

Function Set 11

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'0A', X'82', X'83'	**X'01'** IBM MMR-Modified Modified Read (see Note 1) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) (see Note 1) **X'0A'** Concatenated ABIC (see Note 2) **X'82'** G4 MMR-Modified Modified READ (see Note 1) **X'83'** JPEG algorithms (see Note 3)
	RECID (1)	X'01'	RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01', X'04', X'08', X'18'	**X'01'** 1 bit/IDE **X'04'** 4 bits/IDE **X'08'** 8 bits/IDE **X'18'** 24 bits/IDE
External Algorithm Specification	ID (1)	X'9F'	
	LENGTH (1)	X'0A'	
	ALGTYPE (1)	X'10'	Compression algorithm specification
	RESERVED (1)	X'00'	Should be zero
	COMPRID (1)	X'83'	JPEG algorithms
	RESERVED (3)	X'000000'	Should be zero
	MARKER (1)	X'C0' – X'C2', X'C9' – X'CA'	Non-differential Huffman coding: **X'C0'** Baseline DCT **X'C1'** Extended sequential DCT **X'C2'** Progressive DCT Non-differential arithmetic coding: **X'C9'** Extended sequential DCT **X'CA'** Progressive DCT See Note 3
	RESERVED (3)	X'000000'	Should be zero

Notes on the initial parameters:

1. ABIC (Bilevel Q-Coder), IBM MMR-Modified Modified Read, and G4 MMR-Modified Modified READ are applicable only to images whose IDE size is 1 bit/IDE, otherwise exception condition EC-9611 is raised.
2. Concatenated ABIC is applicable only to images whose IDE size is 4 or 8 bits/IDE, otherwise exception condition EC-9611 is raised.
3. The JPEG baseline DCT-based algorithm is applicable only to images whose IDE size is 8 bits/IDE, while the other DCT-based algorithms are applicable only to images whose IDE size is 8 or 24 bits/IDE; otherwise exception condition EC-9611 is raised.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=1:			
Retired	RESERVED (3)	X'970100'	Retired Image LUT-ID parameter
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'01'	1 bit/IDE
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE
Image Subsampling (see General Note at the end of the table)	ID (2)	X'FECE'	
	LENGTH (2)	X'0000', X'0004'	
	ID (1)	X'01'	Sampling ratios
	LENGTH (1)	X'02'	
	HSAMPLE (1)	X'01'	
	VSAMPLE (1)	X'01'	

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=4 or IDESZ=8			
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'04', X'08'	**X'04'** 4 bits/IDE **X'08'** 8 bits/IDE
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0' – B'1'	**B'0'** No gray coding **B'1'** Gray coding (see Note 1)
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb (see Note 2) **X'12'** YCbCr (see Note 2)
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'04', X'08'	**X'04'** 4 bits/IDE **X'08'** 8 bits/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE
Image Subsampling (see General Note at the end of the table)	ID (2)	X'FECE'	
	LENGTH (2)	X'0000', X'0004'	
	ID (1)	X'01'	Sampling ratios
	LENGTH (1)	X'02'	
	HSAMPLE (1)	X'01'	
	VSAMPLE (1)	X'01'	
Notes on parameters used when IDESZ=4 or IDESZ=8: 1. Gray coding is valid only for the Concatenated ABIC algorithm, otherwise exception condition EC-9B10 is raised. 2. Grayscale images only. Grayscale IDEs are composed of the Y component only of the YCrCb or YCbCr color model.			

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=24:			
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'18'	24 bits/IDE
or:			
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'04'	
	BCOUNT (1)	X'03'	3 bands: R,G,B or Y,Cr,Cb or Y,Cb,Cr
	BITCNT (1)	X'08'	8 bits/IDE for R or Y band
	BITCNT (1)	X'08'	8 bits/IDE for G or Cr or Cb band
	BITCNT (1)	X'08'	8 bits/IDE for B or Cb or Cr band
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'01', X'02', X'12'	**X'01'** RGB **X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'08'	8 bits of the IDE for the R or Y component
	SIZE2 (1)	X'08'	8 bits of the IDE for the G or Cr or Cb component
	SIZE3 (1)	X'08'	8 bits of the IDE for the B or Cb or Cr component
Image Subsampling (see General Note at the end of the table)	ID (2)	X'FECE'	
	LENGTH (2)	X'0000', X'0004', X'0006', X'0008'	
	ID (1)	X'01'	Sampling ratios
	LENGTH (1)	X'02', X'04', X'06'	
	HSAMPLE1 (1)	X'01' – X'02'	**X'02'** YCrCb and YCbCr color models only
	VSAMPLE1 (1)	X'01'	
	HSAMPLE2 (1)	X'01'	
	VSAMPLE2 (1)	X'01'	
	HSAMPLE3 (1)	X'01'	
	VSAMPLE3 (1)	X'01'	

Function Set 11

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Final parameters:			
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs
or:			
Band Image Data (BCOUNT=1 only)	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'01'	One band
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	IDEs
or:			
Band Image Data (BCOUNT=3 only)	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'01'	Band containing the R or Y component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	R or Y component of the IDEs
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'02'	Band containing the G or Cr or Cb component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	G or Cr or Cb component of the IDEs
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'03'	Band containing the B or Cb or Cr component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	B or Cb or Cr component of the IDEs
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	
General note: In this function set, the Image Subsampling parameter and the Band Image parameter cannot coexist within the same Image Content; otherwise exception condition EC-9801 or EC-CE01 is raised.			

IOCA Function Set 14 (IOCA FS14)

Function Set 14 is a superset of Function Set 10 and Function Set 11, and describes bilevel, grayscale, and color images that allow use of transparency masks, as well as some additional compression algorithm options. This function set is carried by the MO:DCA and IPDS controlling environments. The permissible parameter groupings in FS14 are defined as follows:

Table 9. Function Set 14 Structure

		X'70'	Begin Segment parameter		
		X'91'	Begin Image Content parameter		
+		X'94'	Image Size parameter		
+	[	X'95'	Image Encoding parameter		]
+	[	X'96'	IDE Size parameter		]
+	[	X'97'	Retired (Image LUT-ID parameter) (ignored)		]
+	[	X'98'	Band Image parameter		]
+	[	X'9B'	IDE Structure parameter		]
+	[	X'9F'	External Algorithm Specification parameter (ignored)		]
+	[	X'FECE'	Image Subsampling parameter		]
	[		Transparency Mask		]
			Image Data or Band Image Data	(S)	
		X'93'	End Image Content parameter		
		X'71'	End Segment parameter		

Table 10. Transparency Mask Structure

	X'8E'	Begin Transparency Mask parameter	
	X'94'	Image Size parameter	
[	X'95'	Image Encoding parameter	]
	X'FE92'	Image Data	
	X'8F'	End Transparency Mask parameter	

Notes:

1. The Image LUT-ID parameter has been retired and is thus not used in FS14. However, to keep FS14 a superset of FS11, the parameter will be allowed, but ignored.
2. The External Algorithm Specification parameter is not needed in FS14, as there are no restrictions on which codings can be used in the JPEG compression. The parameter is thus allowed, but ignored, as in FS45.

Function Set 14

The self-defining fields and values acceptable for FS14 are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters:			
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Image Size	ID (1)	X'94'	
	LENGTH (1)	X'09'	
	UNITBASE (1)	X'00' – X'02'	
	HRESOL (2)	X'0000' – X'7FFF'	
	VRESOL (2)	X'0000' – X'7FFF'	
	HSIZE (2)	X'0000' – X'7FFF'	
	VSIZE (2)	X'0000' – X'7FFF'	
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'0A', X'0D', X'0E', X'82', X'83'	**X'01'** IBM MMR-Modified Modified Read (see Note 1) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) (see Note 1) **X'0A'** Concatenated ABIC (see Note 2) **X'0D'** TIFF LZW **X'0E'** TIFF LZW with Differencing Predictor **X'82'** G4 MMR-Modified Modified READ (see Note 1) **X'83'** JPEG algorithms (see Note 3)
	RECID (1)	X'01'	RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01', X'04', X'08', X'18'	**X'01'** 1 bit/IDE **X'04'** 4 bits/IDE **X'08'** 8 bits/IDE **X'18'** 24 bits/IDE

Notes on the initial parameters:

1. ABIC (Bilevel Q-Coder), IBM MMR-Modified Modified Read, and G4 MMR-Modified Modified READ are applicable only to images whose IDE size is 1 bit/IDE, otherwise exception condition EC-9611 is raised.
2. Concatenated ABIC is applicable only to images whose IDE size is 4 or 8 bits/IDE, otherwise exception condition EC-9611 is raised.
3. The JPEG algorithms are applicable only to images whose IDE size is 8 or 24 bits/IDE; otherwise exception condition EC-9611 is raised.

Function Set 14

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=1:			
Retired	RESERVED (3)	X'970100'	Retired Image LUT-ID parameter
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'01'	1 bit/IDE
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE
Image Subsampling (see General Note at the end of the table)	ID (2)	X'FECE'	
	LENGTH (2)	X'0000', X'0004'	
	ID (1)	X'01'	Sampling ratios
	LENGTH (1)	X'02'	
	HSAMPLE (1)	X'01'	
	VSAMPLE (1)	X'01'	

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=4 or IDESZ=8			
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'04', X'08'	**X'04'** 4 bits/IDE **X'08'** 8 bits/IDE
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0' – B'1'	**B'0'** No gray coding **B'1'** Gray coding (see Note 1)
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb (see Note 2) **X'12'** YCbCr (see Note 2)
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'04', X'08'	**X'04'** 4 bits/IDE **X'08'** 8 bits/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE
Image Subsampling (see General Note at the end of the table)	ID (2)	X'FECE'	
	LENGTH (2)	X'0000', X'0004'	
	ID (1)	X'01'	Sampling ratios
	LENGTH (1)	X'02'	
	HSAMPLE (1)	X'01'	
	VSAMPLE (1)	X'01'	

Notes on parameters used when IDESZ=4 or IDESZ=8:

1. Gray coding is valid only for the Concatenated ABIC algorithm, otherwise exception condition EC-9B10 is raised.
2. Grayscale images only. Grayscale IDEs are composed of the Y component only of the YCrCb or YCbCr color model.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=24:			
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'18'	24 bits/IDE
or:			
Band Image (see General Note at the end of the table)	ID (1)	X'98'	
	LENGTH (1)	X'04'	
	BCOUNT (1)	X'03'	3 bands: R,G,B or Y,Cr,Cb or Y,Cb,Cr
	BITCNT (1)	X'08'	8 bits/IDE for R or Y band
	BITCNT (1)	X'08'	8 bits/IDE for G or Cr or Cb band
	BITCNT (1)	X'08'	8 bits/IDE for B or Cb or Cr band
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'01', X'02', X'12'	**X'01'** RGB **X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'08'	8 bits of the IDE for the R or Y component
	SIZE2 (1)	X'08'	8 bits of the IDE for the G or Cr or Cb component
	SIZE3 (1)	X'08'	8 bits of the IDE for the B or Cb or Cr component
Image Subsampling (see General Note at the end of the table)	ID (2)	X'FECE'	
	LENGTH (2)	X'0000', X'0004', X'0006', X'0008'	
	ID (1)	X'01'	Sampling ratios
	LENGTH (1)	X'02', X'04', X'06'	
	HSAMPLE1 (1)	X'01' – X'02'	**X'02'** YCrCb and YCbCr color models only
	VSAMPLE1 (1)	X'01'	
	HSAMPLE2 (1)	X'01'	
	VSAMPLE2 (1)	X'01'	
	HSAMPLE3 (1)	X'01'	
	VSAMPLE3 (1)	X'01'	

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Final parameters:			
Begin Transparency Mask	ID (1)	X'8E'	
	LENGTH (1)	X'00'	
Image Size	ID (1)	X'94'	
	LENGTH (1)	X'09'	
	UNITBASE (1)	X'00' – X'01'	**X'00'** 10 inches **X'01'** 10 centimeters
	HRESOL (2)	X'0001' – X'7FFF'	
	VRESOL (2)	X'0001' – X'7FFF'	
	HSIZE (2)	X'0001' – X'7FFF'	
	VSIZE (2)	X'0001' – X'7FFF'	
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00', X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs (bilevel only)
End Transparency Mask	ID (1)	X'8F'	
	LENGTH (1)	X'00'	

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs
or:			
Band Image Data (BCOUNT=1 only)	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'01'	One band
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	IDEs
or:			
Band Image Data (BCOUNT=3 only)	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'01'	Band containing the R or Y component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	R or Y component of the IDEs
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'02'	Band containing the G or Cr or Cb component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	G or Cr or Cb component of the IDEs
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'03'	Band containing the B or Cb or Cr component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	B or Cb or Cr component of the IDEs
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	

General note: In this function set, the Image Subsampling parameter and the Band Image parameter cannot coexist within the same Image Content; otherwise exception condition EC-9801 or EC-CE01 is raised.

IOCA Function Set 40 (IOCA FS40)

Function Set 40 is a subset of Function Set 42, Function Set 45, and Function Set 48. It describes tiled images with one bit per spot (color space YCbCr or YCrCb, IDESZ=1). This function set is carried by the MO:DCA and IPDS controlling environments. The permissible parameter groupings in FS40 are defined as follows:

Table 11. Function Set 40 Structure

	X'70'	Begin Segment parameter		
	X'91'	Begin Image Content parameter		
	X'FEBB'	Tile TOC parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'9B'	IDE Structure parameter		]
[		Tiles	(S)	]
	X'93'	End Image Content parameter		
	X'71'	End Segment parameter		

Table 12. Tile Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'B6'	Tile Size parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'9B'	IDE Structure parameter		]
[	X'FE92'	Image Data	(S)	]
	X'8D'	End Tile parameter		

Notes:

1. Note that the parameters in the above diagram must come in the specified order. Even though the general IOCA architecture allows different ordering for some of the parameters, the FS40 specification is more restrictive. If the parameters are given in a different order, an out-of-sequence exception is raised.
2. In the context of FS40, the Image Size parameter, Image Subsampling parameter, and External Algorithm Specification parameter cause the EC-0001 exception (Invalid parameter) to occur. If the first parameter after the Begin Image Content parameter is not the Tile TOC parameter, the image is not a tiled image and any of the tile-specific parameters (Tile TOC parameter, Begin Tile parameter, etc.) cause EC-0001 to occur.
3. The Image Encoding parameter, IDE Size parameter, Band Image parameter, and IDE Structure parameter are shown as optional and can possibly be specified in two places. The specification within a tile takes precedence over a specification outside of the tile.
4. If the IDE Size parameter is not present, the default IDE size is one bit per pel (bilevel image).
5. If the Image Encoding parameter is not present, the default compression algorithm is X'03' (No Compression), the recording algorithm defaults to X'01' (RIDIC), and the bit order defaults to zero.

Function Set 40

The self-defining fields and values acceptable for FS40 are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters in Function Set 40:			
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Tile TOC	ID (2)	X'FEBB'	
	LENGTH (2)	X'0002' – X'7FFF'	
	RESERVED (2)	X'0000'	Reserved; should be set to zero
	Either zero repeating groups, or one per tile in the following format:		
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01'	Relative resolution
	COMPR (1)		Compression algorithm
	DATAPOS (8)		File offset to the beginning of the tile
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read (see General Note) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) (see General Note) **X'82'** G4 MMR-Modified Modified READ (see General Note)
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01'	1 bit/IDE

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters in a tile:			
Begin Tile	ID (1)	X'8C'	
	LENGTH (1)	X'00'	
Tile Position	ID (1)	X'B5'	
	LENGTH (1)	X'08'	
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
Tile Size	ID (1)	X'B6'	
	LENGTH (1)	X'08' – X'09'	
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01'	Relative resolution
Tile parameters:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read (see General Note) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'82'** G4 MMR-Modified Modified READ (see General Note)
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01'	1 bit/IDE

Function Set 40

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs
End Tile	ID (1)	X'8D'	
	LENGTH (1)	X'00'	
Final parameters in Function Set 40:			
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	
General note: ABIC (Bilevel Q-Coder), IBM MMR-Modified Modified Read, and G4 MMR-Modified Modified READ are applicable only to images whose IDE size is 1 bit per band, otherwise exception condition EC-9611 is raised.			

IOCA Function Set 42 (IOCA FS42)

Function Set 42 is a superset of Function Set 40 and a subset of Function Set 45 and Function Set 48. It describes tiled images with one bit per spot. Images can be either bilevel (color space YCbCr or YCrCb, IDESZ=1) or color (color space CMYK, IDESZ=4). This function set is carried by the MO:DCA and IPDS controlling environments. The permissible parameter groupings in FS42 are defined as follows:

Table 13. Function Set 42 Structure

	X'70'	Begin Segment parameter		
	X'91'	Begin Image Content parameter		
	X'FEBB'	Tile TOC parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[		Tiles	(S)	]
	X'93'	End Image Content parameter		
	X'71'	End Segment parameter		

Table 14. Tile Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'B6'	Tile Size parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[	X'B7'	Tile Set Color parameter		]
[		Image Data or Band Image Data	(S)	]
	X'8D'	End Tile parameter		

Notes:

1. Note that the parameters in Table 13 and Table 14 must come in the specified order. Even though the general IOCA architecture allows different ordering for some of the parameters, the FS42 specification is more restrictive. If the parameters are given in a different order, an out-of-sequence exception is raised.
2. In the context of FS42, the Image Size parameter, Image Subsampling parameter, and External Algorithm Specification parameter cause the EC-0001 exception (Invalid parameter) to occur. If the first parameter after the Begin Image Content parameter is not the Tile TOC parameter, the image is not a tiled image and any of the tile-specific parameters (Tile TOC parameter, Begin Tile parameter, etc.) cause EC-0001 to occur.
3. If the IDE Size is not set to 1 bit or the color space is not YCbCr or YCrCb for a tile, and the Tile Set Color parameter is specified, exception EC-B711 occurs.
4. If the Solid Fill Rectangle compression algorithm is specified for a tile and Image Data or Band Image Data is encountered, exception EC-0001 occurs.
5. The Image Encoding parameter, IDE Size parameter, Band Image parameter, and IDE Structure parameter are shown as optional and can possibly be specified in two places. The specification within a tile takes precedence over a specification outside of the tile.
6. If the IDE Size parameter is not present, the default IDE size is one bit per pel (bilevel image).

7. If the Image Encoding parameter is not present, the default compression algorithm is X'03' (No Compression), the recording algorithm defaults to X'01' (RIDIC), and the bit order defaults to zero.
8. If a tile contains the IDE Structure parameter specifying the CMYK color space, then the IDE Size parameter, Band Image parameter, and Band Image Data must also be present.
9. If the IDE Structure parameter specifying the CMYK color space is given outside of the tiles, then the IDE Size parameter and the Band Image parameter must be given either outside of the tiles or within every tile that does not contain another IDE Structure parameter specifying that the tile is bilevel.
10. CMYK tiles must carry the image data in Band Image Data. Bilevel tiles must carry the data in Image Data, unless the Solid Fill Rectangle compression algorithm is specified.
11. If a tile has Solid Fill Rectangle specified as the compression algorithm, the tile is painted using the color specified in the Tile Set Color parameter for that tile. If the Tile Set Color parameter has not been specified, the color given using the Set Bilevel Image Color field in the Image Data Descriptor is used. If the Set Bilevel Image Color field is missing, the device default is used.

The self-defining fields and values acceptable for FS42 are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters in Function Set 42:			
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Tile TOC	ID (2)	X'FEBB'	
	LENGTH (2)	X'0002' – X'7FFF'	
	RESERVED (2)	X'0000'	Reserved; should be set to zero
	Either zero repeating groups, or one per tile in the following format:		
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01'	Relative resolution
	COMPR (1)		Compression algorithm
	DATAPOS (8)		File offset to the beginning of the tile

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'20', X'82'	**X'01'** IBM MMR-Modified Modified Read (see General Note) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) (see General Note) **X'20'** Solid Fill Rectangle **X'82'** G4 MMR-Modified Modified READ (see General Note)
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01', X'04'	**X'01'** 1 bit/IDE **X'04'** 4 bits/IDE
Initial parameters in a tile:			
Begin Tile	ID (1)	X'8C'	
	LENGTH (1)	X'00'	
Tile Position	ID (1)	X'B5'	
	LENGTH (1)	X'08'	
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
Tile Size	ID (1)	X'B6'	
	LENGTH (1)	X'08' – X'09'	
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01'	Relative resolution

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Tile parameters used when IDESZ=1:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'20', X'82'	**X'01'** IBM MMR-Modified Modified Read (see General Note) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'20'** Solid Fill Rectangle **X'82'** G4 MMR-Modified Modified READ (see General Note)
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01'	1 bit/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'01'	1 bit/IDE
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Tile Set Color	ID (1)	X'B7'	
	LENGTH (1)	X'0B', X'0C'	
	CSPACE (1)	X'04', X'08'	**X'04'** CMYK **X'08'** CIELab
	RESERVED (3)	X'000000'	Should be zero
	SIZE1–SIZE3 (1)	X'08'	Bits/IDE for components 1-3
	SIZE4 (1)	X'00', X'08'	Bits/IDE for component 4
	CVAL1–CVAL4 (1)	X'00' – X'FF'	Color values

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Tile parameters used when IDESZ=4:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read (see General Note) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'82'** G4 MMR-Modified Modified READ (see General Note)
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00', X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'04'	4 bits/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'05'	
	BCOUNT (1)	X'04'	Four bands: CMYK
	BITCNT (1)	X'01'	1 bit/IDE for C band
	BITCNT (1)	X'01'	1 bit/IDE for M band
	BITCNT (1)	X'01'	1 bit/IDE for Y band
	BITCNT (1)	X'01'	1 bit/IDE for K band
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'09'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'04'	CMYK
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE (C component)
	SIZE2 (1)	X'01'	1 bit/IDE (M component)
	SIZE3 (1)	X'01'	1 bit/IDE (Y component)
	SIZE4 (1)	X'01'	1 bit/IDE (K component)

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Final parameters in a tile:			
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs
or:			
Band Image Data (BCOUNT=4 only)	**Four bands, in order by BANDNUM, in the following format:**		
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0004' – X'FFFF'	
	BANDNUM (1)	X'01' – X'04'	**X'01'** Band contains the C component of the IDEs **X'02'** Band contains the M component of the IDEs **X'03'** Band contains the Y component of the IDEs **X'04'** Band contains the K component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	
End Tile	ID (1)	X'8D'	
	LENGTH (1)	X'00'	
Final parameters in Function Set 42:			
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	
General note: ABIC (Bilevel Q-Coder), IBM MMR-Modified Modified Read, G4 MMR-Modified Modified READ, and Solid Fill Rectangle are applicable only to images whose IDE size is 1 bit per band, otherwise exception condition EC-9611 is raised.			

IOCA Function Set 45 (IOCA FS45)

Function Set 45 is a superset of Function Set 40 and Function Set 42 and a subset of Function Set 48. It describes bilevel or color tiled images. This function set is carried by the MO:DCA and IPDS controlling environments. The permissible parameter groupings in FS45 are now defined as follows:

Table 15. Function Set 45 Structure

	X'70'	Begin Segment parameter		
		Image Content	(S)	
	X'71'	End Segment parameter		

Table 16. Image Content Structure

	X'91'	Begin Image Content parameter		
	X'FEBB'	Tile TOC parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[		Data and Referencing Tiles	(S)	]
	X'93'	End Image Content parameter		

Table 17. Data Tile Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'B6'	Tile Size parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[	X'9F'	External Algorithm Specification parameter (ignored)		]
[	X'B7'	Tile Set Color parameter		]
[		Transparency Mask		]
[		Image Data or Band Image Data	(S)	]
	X'8D'	End Tile parameter		

Table 18. Referencing Tile Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'FEB8'	Include Tile parameter		
[		Transparency Mask		]
	X'8D'	End Tile parameter		

Table 19. IOCA Tile Resource Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'B6'	Tile Size parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[	X'9F'	External Algorithm Specification parameter (ignored)		]
[	X'B7'	Tile Set Color parameter		]
[		Transparency Mask		]
[		Image Data or Band Image Data	(S)	]
	X'8D'	End Tile parameter		

Table 20. Transparency Mask Structure

	X'8E'	Begin Transparency Mask parameter	
	X'94'	Image Size parameter	
[	X'95'	Image Encoding parameter	]
	X'FE92'	Image Data	
	X'8F'	End Transparency Mask parameter	

Notes:

1. Note that the parameters in Table 15 on page 112, Table 16 on page 112, Table 17 on page 112, Table 18 on page 112, Table 19, and Table 20 must come in the specified order. Even though the general IOCA architecture allows different ordering for some of the parameters, the FS45 specification is more restrictive. If the parameters are given in a different order, an out-of-sequence exception is raised.
2. The Image Encoding parameter, IDE Size parameter, Band Image parameter, and IDE Structure parameter are shown as optional and can possibly be specified in two places. Note that tile data may require that some of these parameters be specified.
3. If the IDE Size parameter is not present neither in the tile nor in the image content, the default IDE size is one bit per pel (bilevel image).
4. If the Image Encoding parameter is not present, the default compression algorithm is X'03' (No Compression), the recording algorithm defaults to X'01' (RIDIC), and the bit order defaults to zero.
5. If a tile contains the IDE Structure parameter specifying the CMYK color space, then the IDE Size parameter, Band Image parameter, and Band Image Data must also be present.
6. If the IDE Structure parameter specifying the CMYK color space is given outside of the tiles, then the IDE Size parameter and the Band Image parameter must be given either outside of the tiles or within every tile that does not contain another IDE Structure parameter specifying a different color space.
7. Receivers implementing FS45 must support at least 128 image contents in a single image segment. Otherwise, if a receiver encounters too many image contents to process, it should act as if it encountered too many image objects on a page.
8. Resource tiles included via the Include Tile parameter must not contain the Include Tile parameter or the exception EC-B811 exists.

Function Set 45

The self-defining fields and parameter values that are acceptable in Function Set 45 are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters in Function Set 45:			
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Tile TOC	ID (2)	X'FEBB'	
	LENGTH (2)	X'0002' – X'7FFF'	
	RESERVED (2)	X'0000'	Reserved; should be set to zero
	Either zero repeating groups, or one per tile in the following format:		
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01' – X'02'	Relative resolution (see Note 1)
	COMPR (1)		Compression algorithm
	DATAPOS (8)		File offset to the beginning of the tile
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'0D', X'20', X'82' – X'83'	**X'01'** IBM MMR-Modified Modified Read (see Note 2) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) (see Note 2) **X'0D'** TIFF LZW **X'20'** Solid Fill Rectangle **X'82'** G4 MMR-Modified Modified READ (see Note 2) **X'83'** JPEG algorithms (see Note 3)
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01', X'04', X'20'	**X'01'** 1 bit/IDE **X'04'** 4 bits/IDE **X'20'** 32 bits/IDE

Notes on the initial parameters:

1. In the Tile TOC parameter, the relative resolution (RELRES) of 2 is supported only for JPEG-compressed data. Using RELRES of 1 for JPEG-compressed data and RELRES of 2 for non-JPEG data results in exception EC-B610 being raised. Note that this restriction on the relative resolution holds only for this function set, not for the IOCA architecture in general.

   Implementation Note: Some FS45 receivers support a RELRES of 1 for JPEG-compressed data, and do not raise exception EC-B610 for such data. Also note that in FS48, a RELRES of 1 for JPEG-compressed data is allowed.
2. ABIC (Bilevel Q-Coder), IBM MMR-Modified Modified Read, G4 MMR-Modified Modified READ, and Solid Fill Rectangle are applicable only to images whose IDE size is 1 bit/band, otherwise exception condition EC-9611 is raised.
3. The JPEG algorithms are applicable only to images whose IDE size is 32 bits/IDE; otherwise exception condition EC-9611 is raised.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters in a data tile:			
Begin Tile	ID (1)	X'8C'	
	LENGTH (1)	X'00'	
Tile Position	ID (1)	X'B5'	
	LENGTH (1)	X'08'	
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
Tile Size	ID (1)	X'B6'	
	LENGTH (1)	X'08' – X'09'	
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01' – X'02'	Relative resolution (see Note on the data-tile initial parameters)
Note on the data-tile initial parameters: In the Tile Size parameter, the relative resolution (RELRES) of 2 is supported only for JPEG-compressed data. Using RELRES of 1 for JPEG-compressed data and RELRES of 2 for non-JPEG data results in exception EC-B610 being raised. Note that this restriction on the relative resolution holds only for this function set, not for the IOCA architecture in general. **Implementation Note:** Some FS45 receivers support a RELRES of 1 for JPEG-compressed data, and do not raise exception EC-B610 for such data. Also note that in FS48, a RELRES of 1 for JPEG-compressed data is allowed.			
Initial parameters in a referencing tile:			
Begin Tile	ID (1)	X'8C'	
	LENGTH (1)	X'00'	
Tile Position	ID (1)	X'B5'	
	LENGTH (1)	X'08'	
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
Include Tile	ID (2)	X'FEB8'	
	LENGTH (2)	X'0004'	
	TIRID (4)	X'00000000' – X'FFFFFFFF'	Resource Tile local identifier

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=1:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'20', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'20'** Solid Fill Rectangle **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01'	1 bit/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'01'	1bit/IDE
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'00000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE

Function Set 45

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Tile Set Color	ID (1)	X'B7'	
	LENGTH (1)	X'0B', X'0C'	
	CSPACE (1)	X'04', X'08'	**X'04'** CMYK **X'08'** CIELab
	RESERVED (3)	X'000000'	Should be zero
	SIZE1–SIZE3 (1)	X'08'	Bits/IDE for components 1-3
	SIZE4 (1)	X'00', X'08'	Bits/IDE for component 4
	CVAL1–CVAL4 (1)	X'00' – X'FF'	Color values

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=4:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'04'	4 bits/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'05'	
	BCOUNT (1)	X'04'	Four bands: CMYK
	BITCNT (1)	X'01'	1 bit/IDE for C band
	BITCNT (1)	X'01'	1 bit/IDE for M band
	BITCNT (1)	X'01'	1 bit/IDE for Y band
	BITCNT (1)	X'01'	1 bit/IDE for K band
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'09'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'04'	CMYK
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE (C component)
	SIZE2 (1)	X'01'	1 bit/IDE (M component)
	SIZE3 (1)	X'01'	1 bit/IDE (Y component)
	SIZE4 (1)	X'01'	1 bit/IDE (K component)

Function Set 45

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=32:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'03', X'0D', X'83'	**X'03'** No Compression **X'0D'** TIFF LZW **X'83'** JPEG algorithms
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'20'	32 bits/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'05'	
	BCOUNT (1)	X'04'	4 bands: CMYK
	BITCNT (1)	X'08'	8 bits/IDE for C band
	BITCNT (1)	X'08'	8 bits/IDE for M band
	BITCNT (1)	X'08'	8 bits/IDE for Y band
	BITCNT (1)	X'08'	8 bits/IDE for K band
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'09'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'04'	CMYK
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'08'	8 bits/IDE (C component)
	SIZE2 (1)	X'08'	8 bits/IDE (M component)
	SIZE3 (1)	X'08'	8 bits/IDE (Y component)
	SIZE4 (1)	X'08'	8 bits/IDE (K component)

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Final parameters in a tile:			
Begin Transparency Mask	ID (1)	X'8E'	
	LENGTH (1)	X'00'	
Image Size	ID (1)	X'94'	
	LENGTH (1)	X'09'	
	UNITBASE (1)	X'00' – X'01'	**X'00'** 10 inches **X'01'** 10 centimeters
	HRESOL (2)	X'0001' – X'7FFF'	
	VRESOL (2)	X'0001' – X'7FFF'	
	HSIZE (2)	X'0001' – X'7FFF'	
	VSIZE (2)	X'0001' – X'7FFF'	
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00', X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs (bilevel only)
End Transparency Mask	ID (1)	X'8F'	
	LENGTH (1)	X'00'	

Function Set 45

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs
or:			
Band Image Data (BCOUNT=4 only)	**Four bands, in order by BANDNUM, in the following format:**		
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0003' – X'FFFF'	(see Note on the tile-final parameters)
	BANDNUM (1)	X'01' – X'04'	**X'01'** Band contains the C component of the IDEs **X'02'** Band contains the M component of the IDEs **X'03'** Band contains the Y component of the IDEs **X'04'** Band contains the K component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	
End Tile	ID (1)	X'8D'	
	LENGTH (1)	X'00'	
Note on the tile-final parameters: Band Image Data parameters with length of X'0003' do not have a data field. The receiver generates zeros for the corresponding band.			
Final parameters in Function Set 45:			
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	

IOCA Function Set 48 (IOCA FS48)

Function Set 48 is a superset of Function Set 40, Function Set 42, and Function Set 45. It describes bilevel or color tiled images. This function set is carried by the MO:DCA and IPDS controlling environments. The permissible parameter groupings in FS48 are defined as follows:

Table 21. Function Set 48 Structure

	X'70'	Begin Segment parameter		
		Image Content	(S)	
	X'71'	End Segment parameter		

Table 22. Image Content Structure

	X'91'	Begin Image Content parameter		
	X'FEBB'	Tile TOC parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[		Data and Referencing Tiles	(S)	]
	X'93'	End Image Content parameter		

Table 23. Data Tile Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'B6'	Tile Size parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[	X'9F'	External Algorithm Specification parameter (ignored)		]
[	X'B7'	Tile Set Color parameter		]
[		Transparency Mask		]
[		Image Data or Band Image Data	(S)	]
	X'8D'	End Tile parameter		

Table 24. Referencing Tile Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'FEB8'	Include Tile parameter		
[		Transparency Mask		]
	X'8D'	End Tile parameter		

Table 25. IOCA Tile Resource Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'B6'	Tile Size parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[	X'9F'	External Algorithm Specification parameter (ignored)		]
[	X'B7'	Tile Set Color parameter		]
[		Transparency Mask		]
[		Image Data or Band Image Data	(S)	]
	X'8D'	End Tile parameter		

Table 26. Transparency Mask Structure

	X'8E'	Begin Transparency Mask parameter	
	X'94'	Image Size parameter	
[	X'95'	Image Encoding parameter	]
	X'FE92'	Image Data	
	X'8F'	End Transparency Mask parameter	

Notes:

1. Note that the parameters in Table 21 on page 123, Table 22 on page 123, Table 23 on page 123, Table 24 on page 123, Table 25, and Table 26 must come in the specified order. Even though the general IOCA architecture allows different ordering for some of the parameters, the FS48 specification is more restrictive. If the parameters are given in a different order, an out-of-sequence exception is raised.
2. The Image Encoding parameter, IDE Size parameter, Band Image parameter, and IDE Structure parameter are shown as optional and can possibly be specified in two places. Note that tile data may require that some of these parameters be specified.
3. If the IDE Size parameter is not present neither in the tile nor in the image content, the default IDE size is one bit per pel (bilevel image).
4. If the Image Encoding parameter is not present, the default compression algorithm is X'03' (No Compression), the recording algorithm defaults to X'01' (RIDIC), and the bit order defaults to zero.
5. If a tile contains the IDE Structure parameter specifying the CMYK or nColor color space, then the IDE Size parameter, Band Image parameter, and Band Image Data must also be present.
6. If the IDE Structure parameter specifying the CMYK or nColor color space is given outside of the tiles, then the IDE Size parameter and the Band Image parameter must be given either outside of the tiles or within every tile that does not contain another IDE Structure parameter specifying a different color space.
7. Receivers implementing FS48 must support at least 128 image contents in a single image segment. Otherwise, if a receiver encounters too many image contents to process, it should act as if it encountered too many image objects on a page.
8. Resource tiles included via the Include Tile parameter must not contain the Include Tile parameter or the exception EC-B811 exists.

The self-defining fields and parameter values that are acceptable in Function Set 48 are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters in Function Set 48:			
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Tile TOC	ID (2)	X'FEBB'	
	LENGTH (2)	X'0002' – X'7FFF'	
	RESERVED (2)	X'0000'	Reserved; should be set to zero
	Either zero repeating groups, or one per tile in the following format:		
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01' – X'02'	Relative resolution (see Note 1)
	COMPR (1)		Compression algorithm
	DATAPOS (8)		File offset to the beginning of the tile
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'0D', X'0E', X'20', X'82' – X'83'	**X'01'** IBM MMR-Modified Modified Read (see Note 2) **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) (see Note 2) **X'0D'** TIFF LZW **X'0E'** TIFF LZW with Differencing Predictor **X'20'** Solid Fill Rectangle **X'82'** G4 MMR-Modified Modified READ (see Note 2) **X'83'** JPEG algorithms (see Note 3)
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01', X'04', X'10', X'18', X'20', X'28', X'30', X'38', X'40', X'48', X'50', X'58', X'60', X'68', X'70', X'78'	**X'01'** 1 bit/IDE **X'04'** 4 bits/IDE **X'10'** 16 bits/IDE **X'18'** 24 bits/IDE **X'20'** 32 bits/IDE **X'28'** 40 bits/IDE **X'30'** 48 bits/IDE **X'38'** 56 bits/IDE **X'40'** 64 bits/IDE **X'48'** 72 bits/IDE **X'50'** 80 bits/IDE **X'58'** 88 bits/IDE **X'60'** 96 bits/IDE **X'68'** 104 bits/IDE **X'70'** 112 bits/IDE **X'78'** 120 bits/IDE

Notes on the initial parameters:

1. In the Tile TOC parameter, the relative resolution (RELRES) of 2 is supported only for JPEG-compressed data. Using a RELRES of 2 for non-JPEG data results in exception EC-B610 being raised. Note that this restriction on the relative resolution holds only for this function set, not for the IOCA architecture in general. For JPEG-compressed data, either value of RELRES is supported.
2. ABIC (Bilevel Q-Coder), IBM MMR-Modified Modified Read, G4 MMR-Modified Modified READ, and Solid Fill Rectangle are applicable only to images whose IDE size is 1 bit/band, otherwise exception condition EC-9611 is raised.
3. The JPEG algorithms are applicable only to CMYK images whose IDE size is 32 bits/IDE or to nColor images; otherwise exception condition EC-9611 is raised.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters in a data tile:			
Begin Tile	ID (1)	X'8C'	
	LENGTH (1)	X'00'	
Tile Position	ID (1)	X'B5'	
	LENGTH (1)	X'08'	
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
Tile Size	ID (1)	X'B6'	
	LENGTH (1)	X'08' – X'09'	
	THSIZE (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile size
	TVSIZE (4)	X'00000000' – X'7FFFFFFF'	Vertical tile size
	RELRES (1)	X'01' – X'02'	Relative resolution (see Note on the data-tile initial parameters)
Note on the data-tile initial parameters: In the Tile Size parameter, the relative resolution (RELRES) of 2 is supported only for JPEG-compressed data. Using a RELRES of 2 for non-JPEG data results in exception EC-B610 being raised. Note that this restriction on the relative resolution holds only for this function set, not for the IOCA architecture in general. For JPEG-compressed data, either value of RELRES is supported.			
Initial parameters in a referencing tile:			
Begin Tile	ID (1)	X'8C'	
	LENGTH (1)	X'00'	
Tile Position	ID (1)	X'B5'	
	LENGTH (1)	X'08'	
	XOFFSET (4)	X'00000000' – X'7FFFFFFF'	Horizontal tile origin
	YOFFSET (4)	X'00000000' – X'7FFFFFFF'	Vertical tile origin
Include Tile	ID (2)	X'FEB8'	
	LENGTH (2)	X'0004'	
	TIRID (4)	X'00000000' – X'FFFFFFFF'	Resource Tile local identifier

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=1:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'20', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'20'** Solid Fill Rectangle **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01'	1 bit/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'02'	
	BCOUNT (1)	X'01'	One band
	BITCNT (1)	X'01'	1bit/IDE
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'06' – X'08'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'00000'	Should be zero
	FORMAT (1)	X'02', X'12'	**X'02'** YCrCb **X'12'** YCbCr
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE
	SIZE2 (1)	X'00'	0 bits/IDE
	SIZE3 (1)	X'00'	0 bits/IDE

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Tile Set Color	ID (1)	X'B7'	
	LENGTH (1)	X'0B', X'0C'	
	CSPACE (1)	X'04', X'08'	**X'04'** CMYK **X'08'** CIELab
	RESERVED (3)	X'000000'	Should be zero
	SIZE1–SIZE3 (1)	X'08'	Bits/IDE for components 1-3
	SIZE4 (1)	X'00', X'08'	Bits/IDE for component 4
	CVAL1–CVAL4 (1)	X'00' – X'FF'	Color values

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=4:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'04'	4 bits/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'05'	
	BCOUNT (1)	X'04'	Four bands: CMYK
	BITCNT (1)	X'01'	1 bit/IDE for C band
	BITCNT (1)	X'01'	1 bit/IDE for M band
	BITCNT (1)	X'01'	1 bit/IDE for Y band
	BITCNT (1)	X'01'	1 bit/IDE for K band
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'09'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'04'	CMYK
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'01'	1 bit/IDE (C component)
	SIZE2 (1)	X'01'	1 bit/IDE (M component)
	SIZE3 (1)	X'01'	1 bit/IDE (Y component)
	SIZE4 (1)	X'01'	1 bit/IDE (K component)

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when IDESZ=32 and FORMAT=CMYK:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'03', X'0D', X'0E', X'83'	**X'03'** No Compression **X'0D'** TIFF LZW **X'0E'** TIFF LZW with Differencing Predictor **X'83'** JPEG
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'20'	32 bits/IDE
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'05'	
	BCOUNT (1)	X'04'	4 bands: CMYK
	BITCNT (1)	X'08'	8 bits/IDE for C band
	BITCNT (1)	X'08'	8 bits/IDE for M band
	BITCNT (1)	X'08'	8 bits/IDE for Y band
	BITCNT (1)	X'08'	8 bits/IDE for K band
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'09'	
	FLAGS (1)		
	ASFLAG	B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'04'	CMYK
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'08'	8 bits/IDE (C component)
	SIZE2 (1)	X'08'	8 bits/IDE (M component)
	SIZE3 (1)	X'08'	8 bits/IDE (Y component)
	SIZE4 (1)	X'08'	8 bits/IDE (K component)

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Parameters used when FORMAT=nColor:			
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'03', X'0D', X'0E', X'83'	**X'03'** No Compression **X'0D'** TIFF LZW **X'0E'** TIFF LZW with Differencing Predictor **X'83'** JPEG
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00' – X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
IDE Size	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'10', X'18', X'20', X'28', X'30', X'38', X'40', X'48', X'50', X'58', X'60', X'68', X'70', X'78'	**X'10'** 16 bits/IDE (*n*=2) **X'18'** 24 bits/IDE (*n*=3) **X'20'** 32 bits/IDE (*n*=4) **X'28'** 40 bits/IDE (*n*=5) **X'30'** 48 bits/IDE (*n*=6) **X'38'** 56 bits/IDE (*n*=7) **X'40'** 64 bits/IDE (*n*=8) **X'48'** 72 bits/IDE (*n*=9) **X'50'** 80 bits/IDE (*n*=10) **X'58'** 88 bits/IDE (*n*=11) **X'60'** 96 bits/IDE (*n*=12) **X'68'** 104 bits/IDE (*n*=13) **X'70'** 112 bits/IDE (*n*=14) **X'78'** 120 bits/IDE (*n*=15)
Band Image	ID (1)	X'98'	
	LENGTH (1)	X'03' – X'10'	
	BCOUNT (1)	X'02' – X'0F'	2–15 bands
	BITCNT (1)	X'08'	8 bits/IDE for 1st band
	BITCNT (1)	X'08'	8 bits/IDE for 2nd band
	...	...	...
	BITCNT (1)	X'08'	8 bits/IDE for *n*th band

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
IDE Structure	ID (1)	X'9B'	
	LENGTH (1)	X'07' – X'14'	
	FLAGS (1)		
	ASFLAG	Ignored, should be B'0'	Additive
	GRAYCODE	B'0'	No gray coding
	RESERVED	B'000000'	Should be zero
	FORMAT (1)	X'8*n*'	nColor (X'2' ≤ *n* ≤ X'F')
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'08'	8 bits/IDE (1st component)
	SIZE2 (1)	X'08'	8 bits/IDE (2nd component)
	...	...	...
	SIZE*n* (1)	X'08'	8 bits/IDE (*n*th component)

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Final parameters in a tile:			
Begin Transparency Mask	ID (1)	X'8E'	
	LENGTH (1)	X'00'	
Image Size	ID (1)	X'94'	
	LENGTH (1)	X'09'	
	UNITBASE (1)	X'00' – X'01'	**X'00'** 10 inches **X'01'** 10 centimeters
	HRESOL (2)	X'0001' – X'7FFF'	
	VRESOL (2)	X'0001' – X'7FFF'	
	HSIZE (2)	X'0001' – X'7FFF'	
	VSIZE (2)	X'0001' – X'7FFF'	
Image Encoding	ID (1)	X'95'	
	LENGTH (1)	X'02' – X'03'	
	COMPRID (1)	X'01', X'03', X'08', X'82'	**X'01'** IBM MMR-Modified Modified Read **X'03'** No Compression **X'08'** ABIC (Bilevel Q-Coder) **X'82'** G4 MMR-Modified Modified READ
	RECID (1)	X'01', X'04'	**X'01'** RIDIC **X'04'** Unpadded RIDIC
	BITORDR (1)	X'00', X'01'	**X'00'** Bit order within each image data byte is from left to right **X'01'** Bit order within each image data byte is from right to left
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs (bilevel only)
End Transparency Mask	ID (1)	X'8F'	
	LENGTH (1)	X'00'	

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Image Data (IDESZ=1 only)	**Unbanded image data, in the following format:**		
	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs (bilevel only)
or:			
Band Image Data (FORMAT=CMYK only)	**Four bands, in order by BANDNUM, in the following format:**		
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0003' – X'FFFF'	(see Note on the tile-final parameters)
	BANDNUM (1)	X'01' – X'04'	**X'01'** Band contains the C component of the IDEs **X'02'** Band contains the M component of the IDEs **X'03'** Band contains the Y component of the IDEs **X'04'** Band contains the K component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	
or:			
Band Image Data (FORMAT=nColor only)	***n* bands, in order by BANDNUM, in the following format:**		
	ID (2)	X'FE9C'	
	LENGTH (2)	X'0003' – X'FFFF'	(see Note on the tile-final parameters)
	BANDNUM (1)	X'01' – X'0F'	**X'0*n*'** Band contains the *n*th color component of the IDEs
	RESERVED (2)	X'0000'	Should be zero
	DATA	Any	
End Tile	ID (1)	X'8D'	
	LENGTH (1)	X'00'	
Note on the tile-final parameters: Band Image Data parameters with a length of X'0003' do not have a data field. The receiver generates zeros for the corresponding band.			
Final parameters in Function Set 48:			
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	

Appendix A. Compression and Recording Algorithms

This appendix describes in more detail the image compression and recording algorithms currently supported by the IOCA Image Encoding parameter.

This chapter consists of the image compression and recording algorithms that are presently defined in “Image Encoding” on page 34. This appendix is not meant to be a complete description or specification of each algorithm, but is meant to be a short and concise outline of the characteristics of each image compression algorithm.

Compression Algorithms

The following compression algorithms are described in this document. The number to the left of each algorithm is the value that the compression algorithm represents for the COMPRID parameter of the Image Encoding parameter.

Value	Algorithm
X'01'	IBM MMR-Modified Modified Read
X'03'	No compression
X'06'	RL4 (Run Length 4)
X'08'	ABIC (Bilevel Q-Coder)
X'09'	TIFF algorithm 2
X'0A'	Concatenated ABIC
X'0B'	Color compression used by OS/2 Image Support
X'0C'	TIFF PackBits
X'0D'	TIFF LZW
X'0E'	TIFF LZW with Differencing Predictor
X'20'	Solid Fill Rectangle
X'80'	G3 MH-Modified Huffman (ITU-TSS T.4 Group 3 one-dimensional coding standard for facsimile)
X'81'	G3 MH-Modified READ (ITU-TSS T.4 Group 3 two-dimensional coding option for facsimile)
X'82'	G4 MMR-Modified Modified READ (ITU-TSS T.6 Group 4 two-dimensional coding standard for facsimile)
X'83'	JPEG algorithms (see the External Algorithm Specification parameter for details)
X'84'	JBIG2
X'FE'	User-defined algorithms (see the External Algorithm Specification parameter for details)
Other values	All other values are reserved

All of these compression algorithms are *lossless*—they result in no loss of data—except for some JPEG algorithms, which are *lossy*.

Modified ITU-TSS Modified READ Algorithm (IBM MMR-Modified Modified Read)

This compression algorithm was developed by IBM by modifying the ITU-TSS Modified READ (Relative Element Algorithm Designate) algorithm. It allows both one- and two-dimensional correlations among changing color points in image data:

- In one-dimensional (1D) mode, color transitions in the image are coded by a run-length that denotes how long the color continues in the horizontal direction.
- In two-dimensional (2D) mode, the image is coded by how far each IDE is positioned from different color IDEs on the same line or the previous line.

The IBM MMR-Modified Modified Read algorithm differs from the ITU-TSS Modified READ algorithm in the following aspects:

- Infinite K value (only the first scan line is in 1D mode)
- No EOLs, except when switching from 1D to 2D and as part of the EOP
- No time-fill bit

The IBM MMR-Modified Modified Read algorithm also differs from a related algorithm, the ITU-TSS Modified Modified READ algorithm, in that the IBM MMR-Modified Modified Read uses one-dimensional coding for the first image line and two-dimensional coding for the remaining lines, while the ITU-TSS Modified Modified READ algorithm uses two-dimensional coding only.

With the Modified ITU-TSS Modified READ algorithm, only one EOP appears at the end of Image Content.

Notes:

1. IBM MMR-Modified Modified Read allows the IOCA Process Model to determine the number of image points in the horizontal and vertical directions. HSIZE and VSIZE can therefore be zero in the Image Size parameter.
2. If the HSIZE or VSIZE parameter of the Image Size parameter is nonzero, it may be less than the actual number of horizontal or vertical image points encoded in the image data due to padding bits or padding scan lines.

For more details about the ITU-TSS Modified READ algorithm, refer to *Standardization of Group 3 Facsimile Apparatus for Document Transmission*, ITU-TSS Recommendation T.4.

For more details about the ITU-TSS Modified Modified READ algorithm, refer to *Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus*, ITU-TSS Recommendation T.6.

For more details about the IBM MMR-Modified Modified Read compression algorithm, refer to “Binary-image-manipulation Algorithms in the Image View Facility” in *IBM Journal of Research and Development*, Volume 31, Number 1 (January 1987).

No Compression

This method sends raw image data, in binary form, without any reduction.

Note: The value No Compression does *not* allow the IOCA Process Model to determine the number of horizontal image points from the image data. However, VSIZE can be zero in the Image Size parameter.

Run Length 4 (RL4) Compression Algorithm

The Run Length 4 (RL4) algorithm is a binary, one-dimensional, run-length coding method of compression. It is based on code words using four bits. The code words used are common to both white runs and black runs. Table 27 lists the code words.

Table 27. RL4 Code Words

Run Length	Code Word	Code Length
0	B'1111 1110'	8 bits
1–8	B'0'*xxx*	4 bits
9–72	B'10'*xx xxxx*	8 bits
73–584	B'110'*x xxxx xxxx*	12 bits
585–4680	B'1110' *xxxx xxxx xxxx*	16 bits
4681–32767	B'1111 0'*xxx xxxx xxxx xxxx*	20 bits
EOL	B'1111 1111 (1111)'	8 or 12 bits

Two EOL (End-Of-Line) codes are provided to make an encoded string of each scan line start at a byte boundary. Either of these codes is used, depending on whether or not the last run-length code of the previous scan line ends at a byte boundary. Each scan line is represented in the following format:

Figure 18. Scan Line Format

Line Number	Length	W-run	B-run	W-run	. . .	B-run	EOL

← Length (In Bytes) →

Both line number and length are represented as two-byte integers, making it possible to skip lines efficiently or to access a specific line directly for display and editing purposes. Providing line numbers also allows completely white lines to be skipped when recording the compressed data.

Regarding the run encoding, the first run of each line must be white; if a line begins with a black image data element, a white run of length zero must be put in the encoded string. If a line ends with a sequence of white image data elements (which is often the case), the last white run need not be encoded, because it can be calculated from the horizontal size of the Image Content and the total length of the preceding runs.

Note: RL4 does *not* allow the IOCA Process Model to determine the number of horizontal image points from the image data. However, VSIZE can be zero in the Image Size parameter.

ABIC (Bilevel Q-Coder) Compression Algorithm

This algorithm uses an arithmetic coding technique to produce *lossless data compression*, which is an invertible mapping between any data file and a compact representation of the same information.

Note: ABIC does *not* allow the IOCA Process Model to determine the number of horizontal or vertical image points from the image data. Hence both HSIZE and VSIZE cannot be zero in the Image Size parameter.

For more details, refer to R. Arps, T. Truong, D. Lu, R. Pasco, and T. Friedman, “A multipurpose VLSI chip for adaptive data compression of bilevel images”, in *IBM Journal of Research and Development*, Volume 32, No. 6 (November 1988).

TIFF algorithm 2 Compression Algorithm

Tag Image File Format (TIFF) data compression scheme 2 is a method of compression that enables image data to be compressed one-dimensionally and is based upon the ITU-TSS T.4 G3 facsimile one-dimensional coding scheme (G3 MH-Modified Huffman).

The TIFF data compression scheme 2 differs from the ITU-TSS T.4 G3 facsimile one-dimensional coding scheme (G3 MH-Modified Huffman) in the following respects:

- New rows always begin on the next available byte boundary.
- No End-of-line (EOL) code words are used.
- No fill bits are used, except for the ignored bits at the end of the last byte of a row.
- No Return to control (RTC) is used.

Note: TIFF 2 does *not* allow the IOCA Process Model to determine the number of horizontal or vertical image points from the image data. Hence both HSIZE and VSIZE cannot be zero in the Image Size parameter.

For more details about the ITU-TSS Group 3 algorithms, refer to *Standardization of Group 3 Facsimile Apparatus for Document Transmission*, ITU-TSS Recommendation T.4.

Concatenated ABIC Compression Algorithm

This algorithm is a form of compression that utilizes the ABIC compression algorithm.

For image data with an IDE size of *n* bits, a processor begins the compression process by retrieving the first bit of the first IDE, and continuing until the first bit of every IDE has been retrieved, in the order in which the IDEs were recorded. The processor then retrieves the second bit of the first IDE, and so on until all the second bits have been retrieved. This sequential process is continued until the *n*th bit of every IDE has been retrieved.

The raster data obtained from this process is compressed using the ABIC algorithm to form a single string of ABIC compressed image data. This compression may occur during the retrieval process, or after all the raster data has been retrieved. No break in the code indicating an End-of-Line, End-of-Page, or a flag may appear in the compressed data. Thus, the length of each line, the size of the image, and the number of bits per IDE cannot be determined from the concatenated ABIC compressed image data.

Note: Concatenated ABIC does *not* allow the IOCA Process Model to determine the number of horizontal or vertical image points from the image data. Hence both HSIZE and VSIZE cannot be zero in the Image Size parameter.

For more details about the concatenated ABIC algorithm, refer to Arps et al., “A multipurpose VLSI chip for adaptive data compression of bilevel images”.

OS/2 Image Support Compression Algorithm

The color compression algorithm supported by the OS/2 Image Support program, part number 49F4608, is based on an earlier revision (R5.0) of the JPEG draft specification. It is similar to the JPEG baseline system described in “JPEG Compression Algorithms” on page 142.

The OS/2 Image Support program supports data streams in RGB pixel interleaf format only: that is, the color pixels input to the encoder and the decoder output must be of the form *RGBRGB...RGB*.

Note: The OS/2 Image Support implementation of the JPEG compression algorithm does *not* allow the IOCA Process Model to determine the number of horizontal or vertical image points from the image data. Hence both HSIZE and VSIZE cannot be zero in the Image Size parameter.

For more details, refer to William B. Pennebaker and Joan L. Mitchell, “Standardization of Color Image Data Compression”, Part I. “Sequential Coding”, in *Proceedings Electronic Imaging '89 East* (October 2–5, 1989): 109–112.

TIFF PackBits Compression Algorithm

The TIFF PackBits algorithm came from the Apple Macintosh system and is applicable to bilevel images only. Each line is packed independently of any other lines.

Note: TIFF PackBits does *not* allow the IOCA Process Model to determine the number of horizontal or vertical image points from the image data. Hence both HSIZE and VSIZE cannot be zero in the Image Size parameter.

For more details about the algorithm, refer to *TIFF*, Revision 6.0, Final (Aldus Corp.: June 3, 1992).

TIFF LZW Compression Algorithm

The LZW (Lempel-Ziv and Welch) algorithm is applicable to bilevel, and continuous tone or palette grayscale and color images. The algorithm works best if the input uncompressed data is organized into strips of about 8K bytes, with each strip being compressed independently.

The algorithm is based on a translation table, or string table, that maps strings of input characters into codes. The TIFF implementation uses variable-length codes, with a maximum code length of twelve bits. This string table is different for every strip.

Notes:

1. TIFF LZW does *not* allow the IOCA Process Model to determine the number of horizontal or vertical image points from the image data. Hence both HSIZE and VSIZE cannot be zero in the Image Size parameter.
2. LZW encoders sometimes terminate the data early. If the LZW decoder does not produce the expected number of bytes, no exception should be raised and the receiver should fill the remaining data with binary zeros.

For more details about the algorithm, refer to:

- *TIFF*, Revision 6.0, Final (Aldus Corp.: June 3, 1992).
- Terry A. Welch, “A Technique for High Performance Data Compression”, in *IEEE Computer*, vol. 17, no. 6 (June 1984).

TIFF LZW with Differencing Predictor Compression Algorithm

The TIFF LZW with Differencing Predictor compression algorithm is an extension of the TIFF LZW compression algorithm, compressing values that are the differences between pixels rather than the pixel values themselves. All information in the TIFF LZW Compression Algorithm section just above is applicable to this compression algorithm as well. The Differencing Predictor extension is described in Section 14 of the TIFF 6.0 Specification.

For continuous tone images, using the Differencing Predictor usually improves compression—often significantly—compared to base TIFF LZW. However, using this extension on other types of images, where it generally does not improve compression ratios compared to TIFF LZW, is not recommended.

In general, the nature of this algorithm lends itself to better compression when compressing a single color component, as is found in banded data.

Solid Fill Rectangle Compression Algorithm

The Solid Fill Rectangle compression algorithm is applicable to tiled images only. It indicates that the tile contains no image data (Image Data or Band Image Data). Instead, the tile may contain the Tile Set Color parameter and all the image points contained within the tile are painted by the color specified in the Tile Set Color parameter. If the Tile Set Color parameter is missing, the color is specified via the Set Bilevel Image Color parameter. If Set Bilevel Image Color is missing, the device default color is used. The Solid Fill Rectangle compression algorithm is applicable only in bilevel color spaces (IDESZ=1), since Tile Set Color specifies the color space internally and requires that the tile color space specified via the optional IDE Structure and IDE Size parameters be bilevel (that is, as either YCbCr or YCrCb and with the IDE Size as 1).

Since the tile compressed using the Solid Fill Rectangle algorithm is generated by the receiver based on the tile dimensions, the THSIZE and TVSIZE fields in the Tile Size parameters must both be nonzero.

ITU-TSS T.4 Group 3 Coding Standard (G3 MH-Modified Huffman) for Facsimile

The ITU-TSS T.4 Group 3 Coding Standard (G3 MH-Modified Huffman) is a compression method for facsimile, standardized by the ITU-TSS (formerly CCITT). It enables one-dimensional compression.

Note: G3 MH-Modified Huffman does *not* allow the IOCA Process Model to determine the number of image points in the horizontal direction. However, VSIZE can be zero in the Image Size parameter.

For more details, refer to *Standardization of Group 3 Facsimile Apparatus for Document Transmission*, ITU-TSS Recommendation T.4.

ITU-TSS T.4 Group 3 Coding Option (G3 MH-Modified READ) for Facsimile

The ITU-TSS T.4 Group 3 Coding Option (G3 MH-Modified READ) is a compression method for facsimile, standardized by the ITU-TSS. It enables two-dimensional compression.

Note: G3 MH-Modified READ does *not* allow the IOCA Process Model to determine the number of image points in the horizontal direction. However, VSIZE can be zero in the Image Size parameter.

For more details, refer to *Standardization of Group 3 Facsimile Apparatus for Document Transmission*, ITU-TSS Recommendation T.4.

ITU-TSS T.6 Group 4 Coding Standard (G4 MMR-Modified Modified READ) for Facsimile

The ITU-TSS T.6 Group 4 Coding Standard (G4 MMR-Modified Modified READ) is a compression method for facsimile, standardized by the ITU-TSS. It enables two-dimensional compression.

Note: G4 MMR-Modified Modified READ does *not* allow the IOCA Process Model to determine the number of image points in the horizontal direction. However, VSIZE can be zero in the Image Size parameter.

For more details, refer to *Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus*, ITU-TSS Recommendation T.6.

JPEG Compression Algorithms

The JPEG (Joint Photographic Experts Group) technical specification details a series of algorithms that can be applied to arbitrary source image resolutions, many color models, multiple image components, various sampling formats, and continuous-tone renditions of text. The algorithms are not applicable to bilevel images.

Some of these algorithms are lossy.

Note: JPEG stores the actual image size in its header, thus allowing the IOCA Process Model to determine the number of horizontal and vertical image points from the image data. HSIZE and VSIZE can therefore be zero in the Image Size parameter.

For more details, refer to the following publications:
- ISO/IEC International Standard 10918-1
- ITU-TSS Recommendation T.81

JBIG2 (Joint Bi-level Image Experts Group) Compression Algorithm

JBIG2 embodies a set of techniques for compressing bilevel images. It is generally an asymmetric algorithm in the sense that the compression is more computationally demanding than decompression. The data can be encoded either losslessly, so that the decompressed output is an exact copy of the original, or via lossy algorithms, where the decompressed image is “close” to the original.

JBIG2 is organized as a toolkit of many different algorithms. Different subsets of the standard are used for different images and applications. The standard codifies these subsets as “profiles”, much in the same way IOCA uses function sets. JBIG2 receivers commonly implement one or more profiles, and not the whole standard.

JBIG2 compression is defined by the ITU-T Recommendation T.88, ISO/EC 14492:2000 and ITU-T Recommendation T.89 Amendment 1.

Note: JBIG2 stores the actual image size in the compressed datastream, thus allowing the IOCA Process Model to determine the number of horizontal and vertical image points from the image data. HSIZE and VSIZE can therefore be zero in the Image Size parameter.

For more details, refer to the following publications:
- International Telecommunication Union, Recommendation T.88, “Information Technology - Coded representation of picture and audio information - Lossy/lossless coding of bi-level images”
- International Telecommunication Union, Recommendation T.89 Amendment 1, “Application Profiles for Recommendation T.88 - Lossy/lossless coding of bi-level images (JBIG2) for fascimile”

User-defined Compression Algorithm

Code point X'FE' denotes that the compression algorithm is supplied by the user, and that the algorithmic description can be found in the External Algorithm Specification parameter. Users should contact the IOCA architecture group to obtain a unique identifier for their exclusive use.

Compression Algorithms and Explicit Image Dimensions

IOCA generators should set HSIZE and VSIZE to the actual width and height of the image, regardless of the compression algorithm used. Leaving either HSIZE or VSIZE as zero might cause some IOCA receivers to abort prematurely. Table 28 lists all the image compression methods recognized by IOCA. The **HSIZE** and **VSIZE** columns are intended for the use of IOCA receivers. Some compression algorithms, such as the IBM MMR-Modified Modified Read, are able to determine the uncompressed image horizontal/vertical size from the input compressed image data, without referring to the HSIZE/VSIZE of the Image Size parameter, that is, HSIZE/VSIZE can be zero in the Image Size parameter.[2]

Table 28. Image Compression Algorithms Supported by IOCA

Compression	**COMPRID**	**HSIZE**	**VSIZE**
IBM MMR-Modified Modified READ	X'01'	Can be zero in the Image Size parameter	Can be zero in the Image Size parameter
No compression	X'03'	Must be nonzero in the Image Size parameter	Can be zero in the Image Size parameter
Run-Length 4	X'06'	Must be nonzero in the Image Size parameter	Can be zero in the Image Size parameter
ABIC (Bilevel Q-Coder)	X'08'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
TIFF algorithm 2	X'09'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
Concatenated ABIC	X'0A'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
OS/2 Image Support[2]	X'0B'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
TIFF PackBits	X'0C'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
TIFF LZW	X'0D'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
TIFF LZW with Differencing Predictor	X'0E'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
Solid Fill Rectangle	X'20'	Must be nonzero in the Image Size parameter	Must be nonzero in the Image Size parameter
G3 MH-Modified Huffman	X'80'	Must be nonzero in the Image Size parameter	Can be zero in the Image Size parameter
G3 MR Modified READ	X'81'	Must be nonzero in the Image Size parameter	Can be zero in the Image Size parameter
G3 MMR Modified READ	X'82'	Must be nonzero in the Image Size parameter	Can be zero in the Image Size parameter
JPEG algorithms	X'83'	Can be zero in the Image Size parameter	Can be zero in the Image Size parameter
JBIG2	X'84'	Can be zero in the Image Size parameter	Can be zero in the Image Size parameter

2. The OS/2 Image Support compression algorithm is based on an earlier version (V5.0) of the JPEG specification. Although JPEG encodes the actual image width and height in the compressed data header the current OS/2 Image Support implementation of the compression algorithm requires both the HSIZE and VSIZE parameters of the Image Size parameter to contain the actual image size.

Compression Algorithms for Different Image Types

Each compression algorithm is generally valid for only some of the possible image data types. In some cases, even though the use of a particular algorithm is valid, the compression performance can be poor. For a selection of compression algorithms commonly used in practice, Table 29 defines which compression algorithms can be used for each data type.

Table 29. Commonly-used Compression Algorithms for Each Data Type

IDE Size	Color Space	Algorithms
1 bit/IDE	YCrCb (X'02') YCbCr (X'12')	ABIC (X'08') G4 MMR (X'82')
4 bits/IDE	CMYK (X'04')	Concatenated ABIC (X'0A') G4 MMR (X'82')
8 bits/IDE	YCrCb (X'02') YCbCr (X'12')	TIFF LZW (X'0D') JPEG (X'83')
24 bits/IDE	RGB (X'01') YCrCb (X'02') YCbCr (X'12')	TIFF LZW (X'0D') JPEG (X'83')
32 bits/IDE	CMYK (X'04')	TIFF LZW (X'0D') JPEG (X'83')
n×8 bits/IDE (X'2' ≤ *n* ≤ X'F')	nColor (X'8*n*')	TIFF LZW (X'0D') TIFF LZW with Differencing Predictor (X'0E') JPEG (X'83')

Notes:

1. The color space is the FORMAT field in the IDE Structure parameter.
2. The compression algorithm is the COMPRID field in the Image Encoding parameter.
3. “No Compression” (X'03') is valid for all image data types.
4. A mismatch between the image data and compression algorithm causes exception EC-9511 to be raised.

The choice of the compression algorithm can have a major impact on both the printer performance and the print quality. Poor compression ratios can result in large datasets that cannot be downloaded to the printer quickly enough. The time required for decompression generally increases with the size of the compressed image and can also be a problem. The print quality is affected by using a lossy algorithm, such as JPEG, on unsuitable data. For more information on matching the compression algorithm to the type of image data, see Appendix F, “Notes for IOCA Generators”, on page 171.

Recording Algorithms

Recording algorithms describe the format of the image data when it is first created. They describe such characteristics as the direction that the IDEs are recorded, and any boundary or formatting constraint that is placed on the image data. The compression takes place on the recorded image.

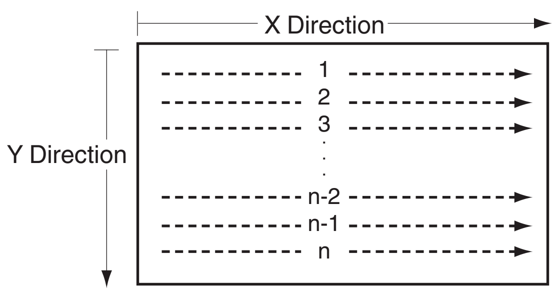
The recording algorithms that can be specified by the RECID parameter of the Image Encoding parameter are:

Value	Algorithm
X'01'	RIDIC (Recording Image Data Inline Coding)
X'03'	Bottom-to-Top
X'04'	Unpadded RIDIC
X'FE'	See the External Algorithm Specification parameter for details
Other values	All other values are reserved

RIDIC Recording Algorithm

The Recorded Image Data Inline Coding (RIDIC) recording algorithm formats the image data as a sequence of binary elements that are generated by the unidirectional raster scan operation. The scanning is from left to right (X direction) and from top to bottom (Y direction), as shown in Figure 19. There are no interlaced fields between the parallel scan lines.

Figure 19. RIDIC Recording Algorithm



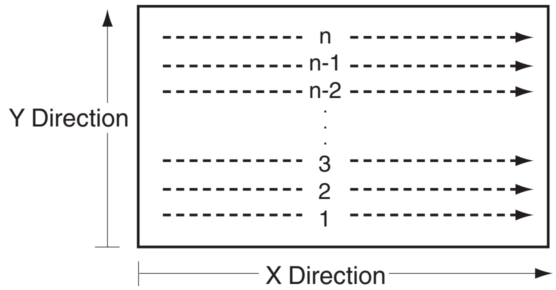
Each raster scan line is in multiples of eight bits. If the width of the image is not a multiple of eight, the scan line must be padded with zeros.

If the Image Size parameter specifies a non-multiple of 8 bits, the resulting compressed image must be compressed at the next multiple of 8 bits and must be decompressed at the next multiple of 8 bits. Once decompressed, only the number of bits specified in the Image Size parameter are to be used for each scan line.

Bottom-to-Top Recording Algorithm

The Bottom-to-Top recording algorithm formats the image data as a sequence of binary elements that are generated by the unidirectional raster scan operation. The scanning is from left to right (X direction) and from bottom to top (Y direction), as shown in Figure 20. There are no interlaced fields between the parallel scan lines.

Figure 20. Bottom-to-Top Recording Algorithm



Each raster scan line is in multiples of 32 bits. If the width of the image is not a multiple of 32, the scan line must be padded with zeros.

Unpadded RIDIC Recording Algorithm

The Unpadded RIDIC algorithm is identical to the RIDIC recording algorithm except that raster scan lines can be any length; no padding is necessary.

Appendix B. Bilevel, Grayscale, and Color Images

This appendix describes the functions of the Image Data parameters that represent bilevel, grayscale, and color images.

Related Image Data Parameters

The Image Data parameters that represent bilevel, grayscale, and color images are:
- IDE Size parameter
- IDE Structure parameter

The IDE Size Parameter specifies the total number of bits per IDE, including all the color planes. If the IDE Size Parameter is absent, the IDE size defaults to 1 bit per IDE. This implies a bilevel image. If the IDE Structure Parameter is absent, the image is assumed to be bilevel if the IDE size is 1 and grayscale otherwise.

If the image is bilevel, the foreground color can be set to an arbitrary color using the Set Bilevel Image Color and Set Extended Bilevel Image Color structured fields in the Image Data Descriptor in MO:DCA. If an image tile is bilevel, the foreground color can also be set using the Tile Set Color parameter. If no color has been specified, the device default is used.

Bilevel Images

The IDE size must be 1 (IDESZ=1) in the IDE Size parameter, or the IDE Size Parameter may be omitted. If the IDE Structure parameter is omitted, the default color space is YCbCr (the Y component is used to express the IDE value) and the GRAYCODE defaults to B'0' (off). The IDE value of B'1' is treated as a significant (toned) pel, while the IDE value of B'0' is treated as an insignificant (untoned) pel.

If the IDE Structure Parameter is present, the color space must be either X'02' (YCrCb) or X'12' (YCbCr) and the GRAYCODE flag must be B'0' (off). The ASFLAG is ignored and the IDE value of B'1' is treated as a significant (toned) pel, while the IDE value of B'0' is treated as an insignificant (untoned) pel.

The foreground color of the significant (toned) pels can be set via the Set Bilevel Image Color and Set Extended Bilevel Image Color structured fields in the Image Data Descriptor in MO:DCA, or the Tile Set Color parameter for bilevel tiles. It is recommended that implementations set the foreground color only for images that conform to the definition of bilevel images earlier in this section; an example of an image that does not conform is a multiple-banded image that contains data in only one, bilevel band.

Note: ASFLAG is ignored for bilevel images to maintain backward compatibility with the current usage, since the FS11, FS14, FS40, FS42, FS45, and FS48 function sets require ASFLAG of B'0' (additive) for bilevel images. For the Y color space, this would imply B'1' being white (untoned) pel, while the IDE value of B'0' is defined to be a toned pel. This is the opposite of how the images are processed.

Grayscale Images

Grayscale images have a value of the IDESZ field in the IDE Size parameter greater than 1. The IDE Structure parameter may be omitted, in which case the default color space is YCbCr (the Y component is used to express the IDE value), the GRAYCODE flag is B'0' (off), and the ASFLAG is B'0' (additive). SIZE1 is assumed to be equal to IDESZ in the IDE Size Parameter. The IDE value of zero is interpreted as black, while the IDE value of 2^{IDESZ}-1 is interpreted as white.

If the IDE Structure Parameter is present, the color space must be either X'02' (YCrCb) or X'12' (YCbCr). The ASFLAG value determines whether a higher IDE value is mapped to a brighter or a darker level.

Color Images

The RGB, YCbCr, and YCrCb color spaces increase in intensity as the R,G,B, and Y increase. If the ASFLAG in the IDE Structure parameter is B'0' (additive), the maximum values represent white and zero values represent black.

In the CMYK color space, an ASFLAG in the IDE Structure Parameter of B'0' means that the zero IDE is white and an IDE with maximum C, M, Y, and K values is black.

In terms of color science, the RGB, YCbCr, and YCrCb color spaces are *additive* color spaces, while the CMYK color space is a *subtractive* color space. This means that colors in the RGB, YCbCr, and YCrCb spaces get lighter as the values increase, while the colors in the CMYK space get darker as the values increase. In both cases, the ASFLAG in the IDE Structure parameter should be set to B'0' (additive) to get the expected behavior. Setting ASFLAG to B'1' (subtractive) yields reverse video.

In the nColor color space, since the characteristics of the color values are not known, it is also not known what the “expected behavior” is. Furthermore, a color management resource (CMR) from the controlling environment is required to process the *n* color components, and the selection of the CMR to use is not affected by whether the IDEs are additive or subtractive. The CMR thus must be matched to the data in terms of whether the maximum value IDE for a color component corresponds with the component being at maximum intensity or minimum intensity. Therefore, for the nColor color space, ASFLAG is ignored.

In practice, the YCbCr color space is most often used to carry RGB data for efficient JPEG compression, since it separates chrominance and luminance. Most image processing applications will describe such JPEG images as RGB. IOCA receivers should consider treating the data in interleaved JPEG-compressed RGB images as if they were YCbCr.

Banded CMYK images may have the C, M, and Y bands marked as zero in the Band Image Data parameter by setting the LENGTH field to X'03'. The resulting image should be treated as a monochrome image by the receiver.

Refer to the resource appendix in the *Mixed Object Document Content Architecture (MO:DCA) Reference* for a description of the RGB model and the Y component of the YCrCb and YCbCr models.

Color Management

Color management is handled by the controlling environment. The controlling environment will take into account the input and output color space characteristics, usually specified via the Color Management Resources, as well as processing instructions specified through the color workflow.

Appendix C. IOCA Tile Resource

This appendix describes the IOCA Tile Resource. This resource is designed to allow images or image parts that are used multiple times in the same datastream to be downloaded to the receiver only once.

A tile resource is an IOCA tile, subject to the following rules and conditions:

- A tile resource can contain any parameter otherwise allowed within tiles, except the Include Tile parameter. If a tile resource does contain the Include Tile parameter, exception EC-B811 (Inconsistent Include Tile) exists when the tile is included.
- The content of the Tile Position parameter in the tile resource is ignored. The receiver uses the Tile Position parameter specified in the calling tile instead.
- If both the tile resource and the calling tile contain Transparency Masks, they are combined using the logical AND operation. A point in the included tile is in the foreground if it is in foreground in both transparency masks. Otherwise, it belongs to the background.
- If only one of the two possible transparency masks is specified, it is used without changes.
- At inclusion time, the tile is treated just as if it were specified locally: the Tile Position parameter in the tile resource is discarded and the transparency mask from the calling tile, if any, is combined with any transparency mask in the included tile. Finally, the included tile, minus the Tile Position and with the possibly changed or added transparency mask is treated as if it appeared instead of the Include Tile parameter in the calling tile.
- Any defaults are applied as if the tile were specified locally.

Table 30 shows the structure of the tile resource.

Table 30. IOCA Tile Resource Structure

	X'8C'	Begin Tile parameter		
	X'B5'	Tile Position parameter		
	X'B6'	Tile Size parameter		
[	X'95'	Image Encoding parameter		]
[	X'96'	IDE Size parameter		]
[	X'98'	Band Image parameter		]
[	X'9B'	IDE Structure parameter		]
[	X'9F'	External Algorithm Specification parameter (ignored)		]
[	X'B7'	Tile Set Color parameter		]
[		Transparency Mask		]
[		Image Data or Band Image Data	(S)	]
	X'8D'	End Tile parameter		

Appendix D. MO:DCA Environment

This appendix describes how Image Objects may be included within a MO:DCA document for the purpose of interchanging the Image Objects between a generating node and one or more receiving nodes. Refer to the *Mixed Object Document Content Architecture (MO:DCA) Reference* for a full description of the MO:DCA data stream.

IOCA Image Object in MO:DCA Data Stream

To guarantee interchange, a MO:DCA document carrying an Image Object must include all information related to the object. The MO:DCA document must therefore contain not only the definition of the Image Object, but must also provide linkage to the resources the object references.

MO:DCA structured fields are discussed in this appendix only as they relate to IOCA Image Objects.

Compliance with MO:DCA Interchange Sets

When Image Objects are interchanged for the purpose of sending the objects to a display, printer, or any other output device, visual fidelity should be maintained as far as possible. In an attempt to satisfy this objective, IOCA defines the following for the MO:DCA environment:

- A set of rules that must be followed by all generators when constructing Image Objects
- A set of image 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 Image Objects must only generate objects that contain image structured fields and values defined in that interchange set. Including structured fields or values not in the interchange set can result in exception conditions being raised by the receiving processor, and exception actions being taken. However, a generator must not rely on a receiver's taking these actions.

In order to conform to a particular MO:DCA interchange set, products that receive Image Objects and convert them using a processor for output to some device are required to support all the image functions defined in that interchange set.

Image Structured Fields

This section describes the Image Data Descriptor (IDD) and Image Picture Data (IPD) structured fields. Each structured field consists of a MO:DCA introducer, followed by one or more image control parameters.

An IOCA Image Segment is carried by one or more IPD structured fields.

Image Data Descriptor (IDD)

The IDD structured field carries the parameters that define the size and resolution of the Image Presentation Space (IPS), and the control parameters required to interpret the Image Segment.

Structured Field Introducer				
SF Length	X'D3A6FB'	Flags	Sequence Number	Data

Offset	Type	Name	Range	Meaning	M/O
0	CODE	UNITBASE	X'00' – X'01'	Unit base: **X'00'** 10 inches **X'01'** 10 centimeters All other values are reserved.	M
1–2	UBIN	XRESOL	X'0001' – X'7FFF'	Horizontal resolution in image points per unit base	M
3–4	UBIN	YRESOL	X'0001' – X'7FFF'	Vertical resolution in image points per unit base	M
5–6	UBIN	XSIZE	X'0001' – X'7FFF'	Horizontal size of the Image Presentation Space in image points	M
7–8	UBIN	YSIZE	X'0001' – X'7FFF'	Vertical size of the Image Presentation Space in image points	M
9–*n*		SDF		Zero or more self-defining fields	O

The following examples illustrate the relationship between the resolution and size parameters of the IDD and the Image Size parameter.

Example 1: Consider an image with an Image Size parameter that specifies its horizontal and vertical sizes to be two inches, and its horizontal and vertical resolutions to be 300 image points per inch. If one wants the image, when written to the IPS, to remain at two inches in both dimensions, the mandatory parameters of the IDD would have the following values:

UNITBASE = X'00'
XRESOL = X'0BB8'
YRESOL = X'0BB8'
XSIZE = X'0258'
YSIZE = X'0258'

Example 2: If one wants the same image to appear at twice the size as the actual image when written to the IPS—that is, the image in the IPS has a horizontal and vertical size of four inches—the mandatory parameters of the IDD would have the following values:

UNITBASE = X'00'

XRESOL = X'05DC'
YRESOL = X'05DC'
XSIZE = X'0258'
YSIZE = X'0258'

This can be done more easily using the scale-to-fit mapping option in the MO:DCA data stream.

Example 3: Conversely, if one wants the same image to appear at half the size as the actual image when written to the IPS—that is, the image in the IPS has a horizontal and vertical size of one inch—the mandatory parameters of the IDD would have the following values:
UNITBASE = X'00'
XRESOL = X'1770'
YRESOL = X'1770'
XSIZE = X'0258'
YSIZE = X'0258'

As can be seen in the previous examples, the horizontal and vertical resolutions of the image as specified in the Image Size parameter are ignored when writing to the IPS. Resolutions specified in the IDD are used instead of the resolutions specified in the Image Size parameter. In the case of an image with undefined resolution, as described in "Image Size" on page 32, each image point in the IOCA Image Content is mapped to one image point in the IPS. The combination of the horizontal and vertical sizes of the Image Size parameter and the horizontal and vertical resolutions of the IDD determines the actual presentation size of the image.

The image is always written to the IPS starting from the IPS origin. For example, the two-inch square image mentioned in Example 1 appears on the top half of the IPS if the mandatory parameters of the IDD contain the following values:
UNITBASE = X'00'
XRESOL = X'0BB8'
YRESOL = X'0BB8'
XSIZE = X'0258'
YSIZE = X'04B0'

If the image cannot fit entirely within the IPS, the IOCA exception condition EC-A902 is raised.

Set Bilevel Image Color

This optional self-defining field specifies a named color value of the significant image points of Bilevel Images.

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'F6'	Set Bilevel Image Color	M
1	UBIN	LENGTH	X'04'	Length of the parameters to follow	M
2	CODE	AREA	X'00'	Applicability area: **X'00'** Foreground All other values are reserved.	M
3			X'00'	Reserved; should be zero	M
4	CODE	NAMECOLR	X'0000' – X'0010', X'FF00' – X'FF08', X'FFFF'	Named colors: **X'0000'** Presentation process default **X'0001'** Blue **X'0002'** Red **X'0003'** Magenta or pink **X'0004'** Green **X'0005'** Cyan or turquoise **X'0006'** Yellow **X'0007'** White **X'0008'** Black **X'0009'** Dark blue **X'000A'** Orange **X'000B'** Purple **X'000C'** Dark green **X'000D'** Dark turquoise **X'000E'** Mustard **X'000F'** Gray **X'0010'** Brown **X'FF00'** Presentation process default **X'FF01'** Blue **X'FF02'** Red **X'FF03'** Magenta or pink **X'FF04'** Green **X'FF05'** Cyan or turquoise **X'FF06'** Yellow **X'FF07'** Presentation process default **X'FF08'** Color of the medium **X'FFFF'** Presentation process default All other values are reserved.	M

If an invalid or unsupported value is encountered in the self-defining field, the entire self-defining field is ignored. If multiple Set Bilevel Image Color self-defining fields appear within the same IDD, the last one encountered is used and all the others are ignored. The IOCA Process Model should notify the controlling environment when it encounters any of the above exception conditions.

Notes:

1. The medium is typically the physical paper in a printer environment, and the monitor screen in the display environment.
2. The presentation process is typically the program that performs the final imaging step on the medium.
3. This self-defining field is ignored if it is present and the image is not bilevel.

4. Color specified by X'0007', rendered on presentation devices that do not support white, is device dependent. For example, some printers simulate white with the color of the medium, which results in white when a white medium is used.

Set Extended Bilevel Image Color

This optional self-defining field specifies a color value and defines the color space and encoding for that value. This SDF is applicable only to significant image points of Bilevel Images.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'F4'	Set Extended Bilevel Image Color	M
1	UBIN	LENGTH	X'0C' – X'0E'	Length of the parameters to follow	M
2				Reserved; must be zero	M
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	M
4–7				Reserved; must be zero	M
8	UBIN	ColSize1	X'01' – X'08', X'10'	Number of bits in component 1; see color space definitions	M
9	UBIN	ColSize2	X'00' – X'08'	Number of bits in component 2; see color space definitions	M
10	UBIN	ColSize3	X'00' – X'08'	Number of bits in component 3; see color space definitions	M
11	UBIN	ColSize4	X'00' – X'08'	Number of bits in component 4; see color space definitions	M
12–*n*		Color		Color specification; see “Set Extended Bilevel Image Color Semantics” on page 158 for details	M

Set Extended Bilevel Image Color Semantics

ColSpce Is a code that defines the color space and the encoding for the color specification.

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 range for the R,G,B components is 0 to 1, which is mapped to the binary value range 0 to ($2^{ColSizeN}$ - 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* D_{6500} or simply *D65*. The R,G,B components are assumed to be gamma-corrected (non-linear) 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 ($2^{ColSize\mathit{N}}$ - 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-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.

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' to X'00FF' are reserved for existing highlight color applications and devices. That is, indexed color values in the range X'0000' to 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' to 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^{ColSize1} - 1). The range for the a and b components is -127 to +127, which is mapped to the binary range -($2^{\mathrm{ColSize}N\text{-}1}$ - 1) to +($2^{\mathrm{ColSize}N\text{-}1}$ - 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.

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.

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.

Color Specifies the color value in the defined format and encoding. Note that the number of bytes specified for this parameter depends on the color space. For example, when using 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 self-defining field length is valid.

To illustrate the syntax for the color value specified in the Color field, the following table shows examples of various ways that a light-green color can be specified. Note that the light-green color value is approximated in each of the color spaces.

ColSpce	ColSize1	ColSize2	ColSize3	ColSize4	Color
RGB	X'08'	X'08'	X'08'	N/A	X'00B761'
RGB	X'05'	X'05'	X'05'	N/A	X'00160B'
CMYK	X'08'	X'08'	X'08'	X'08'	X'FF489E00'
CMYK	X'01'	X'02'	X'04'	X'08'	X'01010900'
Highlight (see note)	X'10'	X'08'	X'00'	N/A	X'000264'
CIELAB	X'08'	X'08'	X'08'	N/A	X'A8CC21'
Standard OCA	X'10'	N/A	N/A	N/A	X'0004'
Note: This example assumes that the light-green color is loaded in the printer as highlight color X'0002'.					

Architecture Note: 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.

Notes:

1. This self-defining field is ignored if it is present and the image is not bilevel.
2. This field can coexist with the Set Bilevel Image Color self-defining field.
3. If multiple instances of this field and the Set Bilevel Image Color field are present, the last instance of a supported field is used, while the others are ignored.

If an invalid or unsupported value is encountered in the self-defining field, the entire self-defining field is ignored. The IOCA Process Model should notify the controlling environment if this exception condition appears, or if multiple instances of this field and/or Set Bilevel Image Color field are present.

IOCA Function Set Identification

This optional self-defining field is carried by the IDD described in "Image Data Descriptor (IDD)" on page 154. It specifies the IOCA function set carried by the IPD.

IOCA function sets are defined in "Function Sets" on page 83.

Offset	Type	Name	Range	Meaning	M/O
0	CODE	SDFID	X'F7'	IOCA Function Set Identification	M
1	UBIN	LENGTH	X'02'	Length of the parameters to follow	M
2	CODE	CATEGORY	X'01'	Function Set category: **X'01'** Function Set identifier All other values are reserved.	M
3	CODE	FCNSET	X'0A' – X'0B', X'0E', X'28', X'2A', X'2D', X'30'	Function Set identifier: **X'0A'** Function Set 10 **X'0B'** Function Set 11 **X'0E'** Function Set 14 **X'28'** Function Set 40 **X'2A'** Function Set 42 **X'2D'** Function Set 45 **X'30'** Function Set 48 All other values are reserved.	M

Image Picture Data (IPD)

An IOCA Image Segment is carried by one or more IPD structured fields.

Structured Field Introducer				
SF Length	X'D3EEFB'	Flags	Sequence Number	Image Segment

See Chapter 5, "IOCA Image Segment", on page 23 for the syntax and description of Image Segments.

Notes:

1. An IOCA Image Segment can be split into multiple IPD structured fields. There are no restrictions on how the image segment is split between multiple IPD structured fields. Data beyond the End Segment self-defining field is ignored by receivers.
2. Each image point in IOCA Image Content is mapped to one image point in the Image Presentation Space.

Appendix E. IPDS Environment

The Intelligent Printer Data Stream (IPDS) provides the printer subsystem environment for Image Objects. This appendix describes:

- The context of Image Objects in the IPDS environment
- IPDS commands specific to images
- Some special considerations when printing an image

For further information about the IPDS architecture, refer to *Intelligent Printer Data Stream Reference*.

IOCA Image Objects in an IPDS Architecture

The IPDS architecture provides various commands to control advanced-function printers. It supports all-points-addressable printing functions that allow text and individual image, graphics, and bar code objects to be positioned and presented at any point on the printed page.

Image Objects are described to IPDS printers in terms of Image Segments as defined by IOCA. They are presented in rectangular output areas called *object areas*. These object areas may be positioned at any addressable point on a page, in an overlay or a page segment definition, and may be defined in any orientation relative to the X-axis of the reference system. The size, position, and orientation of the image object area is defined to the printer by parameters that are specified in the Write Image Control 2 command.

The data within the Image Presentation Space (IPS) can be mapped to the image object area in several different ways, as specified by the Mapping Control Option parameter of the Write Image Control 2 command. These options are as follows:

Scale to Fit
: Map the center of the IPS to the center of the object area, and uniformly scale to fit, without changing the aspect ratio of the image. All image data within the IPS is presented when this option is specified.

Scale to Fill
: Map the center of the IPS to the center of the object area. Scale independently in the X and Y directions so that the object area is filled. The aspect ratio of the image may change. All image data within the IPS is presented when this option is specified.

Center and Trim
: Map the center of the IPS to the center of the object area without scaling. Excess image data, if any, is trimmed at object area boundaries.

Position and Trim
: Map the upper left corner of the IPS to the object area without scaling, using the specified offset from the image object area origin. Excess image data, if any, is trimmed at object area boundaries.

Replicate and Trim
: Map the upper-left corner of the IPS to the object area without scaling, then replicate in both the X and Y directions until the object area is filled. Excess image data, if any, is trimmed at object area boundaries.

Image Point-to-Pel
: Map the upper-left corner of the IPS to the origin of the object area. Each image point is mapped to a single output pel: that is, no resolution correction is done. Excess image data, if any, is trimmed at object area boundaries.

Image Point-to-Pel with Double Dot
: Same as Image Point-to-Pel, except that each image point is mapped to four pels in the object area by doubling the image point in both dimensions. No resolution correction is done. Excess image data, if any, is trimmed at object area boundaries.

If the Image Output Control parameters are omitted, the default is Position and Trim.

Note: If the IOCA object is included in a MO:DCA object and the Map Image Object structured field is not present, the MO:DCA default of Scale to Fit applies and the resulting IPDS contains an explicit Scale to Fit Mapping Control Option. For this reason, the IPDS default is very unlikely to be relevant for most applications.

Resolution correction occurs in the Scale to Fit, Scale to Fill, Center and Trim, Position and Trim, and Replicate and Trim mapping options whenever the resolution of the image points in the IPS, in one or both dimensions, is different from the pel resolution of the printer.

Manipulation of Image Objects can be performed in an IO-Image object state that is entered from any one of three IPDS printer states:
- Page state
- Overlay state
- Page segment state

When the image functions are carried out in the overlay or page segment state, the image data sent to the printer is saved as part of the overlay or page segment definition. It is later included on pages by the Load Copy Control, Include Overlay, or Include Page Segment command.

IPDS IO-Image Command Set

The IPDS architecture provides the IO-Image Command Set to convey image information to printers. This Command Set consists of:

- The Write Image Control 2 command, to define where and how to present an Image Object
- The Write Image 2 command, which contains an Image Segment

Write Image Control 2

The Write Image Control 2 (WIC2) command is identified by command code X'D63E', and is sent to the printer before the Write Image 2 command. It tells the printer that the Write Image 2 commands that follow are directed to an image object area on the current page, overlay, or page segment.

This command defines the size, placement, and orientation of the image object area. It also establishes the parameters required to interpret the Image Segment.

The Write Image Control 2 data is made up of the following three consecutive self-defining fields:

Image Area Position (IAP)
This mandatory self-defining field defines the position and orientation of the image object area relative to a reference coordinate system.

Image Output Control (IOC)
This optional self-defining field specifies the size of the image object area and mapping option for mapping the Image Presentation Space to the image object area.

If it is omitted, the Position and Trim mapping option applies where the offsets are zero, and the image object area size is the same as the Image Presentation Space size as defined in the IDD self-defining field.

Image Data Descriptor (IDD)
This mandatory self-defining field specifies the parameters that define the Image Presentation Space size, and control parameters required to interpret the Image Segment.

Refer to the *Intelligent Printer Data Stream Reference* for a complete description of the above self-defining fields.

Write Image 2

The Write Image 2 (WI2) command is identified by command code X'D64E'. One or more Write Image 2 commands carry one IOCA Image Segment to the printer.

All Image Segments are executed in Immediate mode. That is, they are not retained or stored as named segments, but processed immediately when the printer receives them.

There are no quantity restrictions on data sent to the printer in a single Write Image 2 command, except for the 32K-length limit of the command. An Image Segment, delimited by the Begin Segment and End Segment self-defining fields, may span two or more consecutive Write Image 2 commands.

The IO-Image Command Set allows for Image Segments that conform to Function Sets, described in Chapter 7, “Compliance”, on page 83.

Exception Handling

A data-stream exception occurs when the printer detects an invalid or unsupported command, control, or parameter value in the data stream received from the controlling environment. The IPDS architecture assigns a unique exception code to each exception condition.

The IPDS architecture defines exception conditions and actions that may be detected in IOCA Image Segments carried in the IPDS data stream. They are compatible with IOCA-defined exception conditions and actions.

The IPDS Exception Identifier consists of the two-byte EC identifier defined by IOCA, prefixed by an IPDS exception class value of X'05'. The exception class value is used to distinguish between the two-byte EC identifiers assigned by IOCA, and other two-byte EC identifiers assigned to presentation text (PTOCA), graphics (GOCA), and bar code (BCOCA) objects.

Unsupported IOCA Function in an IPDS Environment

Not all IOCA printers support the full range of IOCA function; these printers will return an appropriate NACK if unsupported IOCA self-defining fields or values are included in an image. For example, if an IOCA FS11, FS14, FS40, FS42, FS45, or FS48 image is sent to an IPDS printer that only supports IOCA FS10, the printer will encounter a data stream error and will return one or more exception conditions such as EC-0001 (invalid or unsupported self-defining field code) or EC-9510 (unsupported compression algorithm).

Additional Related Commands

The following commands are used for query and resource management functions. Only an overview of these commands is presented here. 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 supported command sets. The information requested is returned in the Special Data Area of the Acknowledge Reply. The command sets and the 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 coded-font resolution, and color support.

Special Notes

This section describes special considerations for the IPDS environment.

Image Segment in IO-Image Command Set

For untiled image contents, the image size is specified in the Image Size parameter that is a mandatory parameter within an untiled Image Content. An exception condition occurs if the parameter either is not found, appears more than once, appears before the Begin Image Content, or appears after the first Image Data self-defining field. In this situation, the IOCA standard exception action and IPDS Alternate Exception Action (AEA) is to process the rest of the Image Segment.

Since the Image Size parameter is mandatory in each untiled Image Content, its contents (except for values in Unit Base, Horizontal, and Vertical Resolutions) must be checked for validation. Exceptions occur under the following conditions:

- The Image Size parameter specifies an unknown horizontal image size (HSIZE=0), and an image compression algorithm other than IBM MMR-Modified Modified Read, JPEG, or JBIG2 is selected in the Image Encoding parameter. The IOCA exception action and the IPDS AEA is to skip to the end of the Image Segment.
- The size detected from the image data is different from that specified in the Image Size parameter. The IOCA exception action and the IPDS AEA is to use the size of the image detected from the image data.

  When the image size extends beyond the XSIZE or YSIZE of the Image Presentation Space, an exception condition occurs. The IOCA exception action and the IPDS AEA is to write only portions of the image that are within the Image Presentation Space, and discard all portions that extend outside it. The portions that are not written onto are filled with zeros.

Each image point in IOCA Image Content is mapped to one image point in the Image Presentation Space. The relationship between the resolution and size parameters of the IDD and the Image Size parameter are further described in "Image Data Descriptor (IDD)" on page 154.

Interpretation of IDE Value

Bilevel images are represented by an IDE size of one. Each IDE can represent two values, B'1' or B'0'. In the IPDS architecture, an IDE value of B'1' represents a significant bit that is an image point representing a toned pel in the printer, while B'0' represents an insignificant bit that is an image point representing an untoned pel in the printer.

Image Presentation Space Mapping

The image to be printed is represented as an array of image points in the Image Presentation Space after execution of the Image Segment. The size of the Image Presentation Space and the resolution of the image points within it are defined in the IPDS WIC2 IDD self-defining field.

The size of the Image object area is defined in the IPDS WIC2 IOC self-defining field.

Printing the image data requires the printer to map the *logical* image existing in the Image Presentation Space to a *physical* image in the image object area on the page. The mapping options specified in the IPDS WIC2 IOC self-defining field define how the image will be located with respect to the object area, and whether scaling is needed.

Resolution correction occurs in the Scale to Fit, Scale to Fill, Center and Trim, Position and Trim, and Replicate and Trim mapping options whenever the resolution of the image points in the IPS, in one or both dimensions, is different from the pel resolution of the printer.

Appendix F. Notes for IOCA Generators

IOCA is designed to support printing of images at high speed. However, it is relatively easy to construct syntactically valid images that have extremely poor performance. This is particularly true in printing color, since high speed color printing is a very demanding task.

This appendix reviews some of the most important concerns that should be addressed to ensure that the images print with both high performance and good quality.

General Considerations

When printing images, the concern is the complexity of an *average* page. The printer control unit in fast continuous forms printers generally processes pages in advance of printing. Thus, a sequence of several “easy” pages, followed by a “hard” page might print at rated speed, since the average page complexity is still acceptable.

A letter-sized page at 600 dpi contains roughly 33 million pixels. Image operations on images of such a size, even when they are black and white bilevel (one bit per pel), tend to be prohibitively expensive. If at all possible, the images should be generated at the right size and orientation.

Generators should bear in mind that in MO:DCA datastreams the default mapping option is Scale to Fit. A Map Image Object specifying Position and Trim should be specified explicitly. If the Image Presentation Space dimensions do not quite match the image object dimensions specified in the Object Area Descriptor, the default mapping forces the printer to scale the image. Even if the scaling is unnoticeable (for example, there is a one scanline difference between the object and image lengths), it extracts a significant performance penalty. In contrast, trimming or padding of bilevel images can usually be performed at rated speed.

Unlike scaling, rotation of images can sometimes be performed at high speed. For color images, this subject is discussed below. For black and white bilevel images, some printers can perform rotation in the runend domain. In the runend domain, only the transitions between black and white runs in the image are recorded. For images containing text or line art, there are few runs per scanline and the runend domain algorithms perform very efficiently. Halftone images, on the other hand, are far less suitable for such an approach.

Given the complexity of the rotation issue, it is much better to generate images at the proper orientation. Note that in continuous forms printers, the default orientation for one-up printing is landscape. To achieve high image performance in this context, the images should be prerotated 90 degrees and should have the rotation in the Object Area Position set to 270 degrees. In most printers, assuming that one-up prints at 90-degree landscape orientation, this avoids rotation in the printer control unit.

Printing halftones poses several distinct challenges:

- *Compressed image size*. High frequency halftones tend to compress very poorly. For example, 212lpi halftones used in some of the color printers cause the G4 MMR compression to actually expand data. If the halftoned area is not large, or if the image is light, this is not a particular concern. If the halftoned images are causing performance difficulties, lower frequency screens of 106lpi or below should be used.
- *Device dependency*. Halftoned images are device dependent. The halftone screens are built for a particular type of the print engine. Moreover, each print engine behaves differently and behavior changes unpredictably with time, based on many environmental and internal factors. For the best quality, the halftones should be calibrated frequently. If quality output is desired, halftone images should not be archived. The generators should rather archive the original color or grayscale and generate the halftoned IOCA when the print device characteristics are known. Black and white text and linework are not device-specific and can be archived safely.

Copyright © AFP Consortium 2010, 2022

- *Scaling impact*. Scaling halftoned images by non-integer factors results in artifacts and unacceptable output quality. The generators should ensure that the image is generated with the same resolution used by the printer. The only exception is if the printer resolution is an even multiple of the image resolution. For example, printing a 300dpi image on a 600dpi printer produces a good quality image, albeit at 300dpi. Printing a 240dpi image on a 600dpi printer results in visible artifacts and poor quality, because 600 is not evenly divisible by 240.

Most bilevel IOCA images are generated using the RIDIC recording algorithm and the G4 compression algorithm. Generators should keep in mind that RIDIC requires the image scanlines to be padded to a multiple of eight before compression. Note that TIFF images, which are often used as a source for generating IOCA, also support G4, but do not require that the scanlines be padded. Rewrapping G4 TIFF images with widths that are not multiples of 8 in IOCA is a major source of errors.

If a TIFF image has a width that is not a multiple of 8, generators should decompress the image, pad each scanline to a multiple of 8 and then recompress. Alternatively, generators should use the Unpadded RIDIC recording algorithm, which does not require that the scanlines be padded. Be warned, however, that not all printers support the Unpadded RIDIC recording algorithm.

Function Set 42 Considerations

Function Set 42 should be used only for those color printers that do not support the full color Function Set 45. Images using one bit per spot (per pel per color component) have worse quality than images using 8 bits per spot. The greater bit depth also allows a range of more sophisticated compression schemes, so the full color images also require less data per unit of image area than images using one bit per spot.

Since the images in Function Set 42 have one bit per spot, the colors are obtained by halftoning. This includes any color used in the image except fully saturated cyan, magenta, yellow, black, or white.

Color printers supporting FS42 will likely use high frequency screens. Function Set 42 also supports the ABIC compression algorithm, which does compress the halftoned data. ABIC, however, tends to be very expensive to decompress. In some cases, it might be preferable to send FS42 images uncompressed.

Tiles compressed using the Solid Fill Rectangle algorithm are not affected by scaling, regardless of the color specified in the related Tile Set Color or Set Bilevel Image Color.

Images that contain just black or other fully saturated color text and line work can also be scaled in the printer without excessive loss of quality, though performance still suffers.

Function Set 42 images containing CMYK tiles cannot be transformed to print on a bilevel (black and white) printer with reasonable performance and quality. These images are halftoned, which involves an information loss. To obtain a bilevel image, the CMYK bilevel image must first be analyzed and transformed back into 8-bits/band CMYK. The 8-bit data can then be used to compute the 8-bit luminance (grayscale), that in turn has to be halftoned for the bilevel output device. The process is very compute-intensive and, given the information loss at several stages, likely to lead to poor-quality output. If the application anticipates having to present the image on different devices, the full color image, either 8 bit CMYK or, even better, a device-independent format like CIELab, should be archived. Applications are strongly discouraged from trying to recover device-independent color from the 1-bit/band CMYK. Since each output CMYK device has different characteristics, even printing a CMYK image halftoned for one device on a different device might lead to poor quality.

Function Set 45 and 48 Considerations

To achieve good performance and quality with the full color images, it is crucial that the images are compressed using a compression algorithm that is best matched to the type of the image:

- IBM MMR-Modified Modified READ algorithm is obsolete. Using G4 MMR compression almost always results in better compression.
- MMR algorithms are well-suited for compressing text and line art. If the image contains halftones, the compression ratios degrade as the screens get finer. At roughly 150 lines per inch, the G4 algorithm generally compresses the data. At 212 lines per inch (high end color printers tend to use frequencies of 212lpi and above), the MMR algorithms cause the image to actually expand, possibly by a factor of two or more. For such images, using no compression is currently the best choice.
- The ABIC compression algorithm compresses even high frequency halftones. ABIC is a complex algorithm and decompressors can be slow, depending on the printer. In some cases, an image compressed with ABIC takes longer to download and decompress than the same image uncompressed, even though the uncompressed image has more than twice the amount of data. The performance of the ABIC decompressor in the printer should be tested before the decision is made to use ABIC.
- The JPEG algorithm is well-suited for compressing continuous tone images such as photographs. Using JPEG on text, line art, pie charts, and similar images results in artifacts and unacceptable image quality. Such images should be compressed using the TIFF LZW algorithm.
- The TIFF LZW algorithm is an excellent general-purpose lossless algorithm. It is particularly well suited to compressing large areas of uniform color. While the output very rarely expands (unlike MMR-type algorithms on halftones), it generally achieves only 10% compression on continuous tone images. For such images, JPEG should be used.

Using a valid compression algorithm that is poorly matched to the data does not cause any exception to be raised, but negatively affects either the printer performance or output quality or both.

Given the large datasets needed to print full color images, it is even more crucial that the images be generated at the right size, resolution, and orientation.

Appendix G. Retired Architecture

Architecture listed in this appendix has been retired in the sense that generators can stop issuing the self-defining fields. The receivers must not generate the EC-0001 exception on receiving them, but are allowed to ignore them. Receivers that support the retired self defining fields can continue to parse these fields and generate exceptions if the fields are specified out of sequence, or their contents are invalid.

Each section in this Appendix that does not cover a self-defining field has the receiver impact described in the introduction.

Image LUT-ID

This optional self-defining field identifies the LUT-ID (LUT) that should be used to interpret the image data. Each IDE value is an index into this LUT.

Syntax

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ID	X'97'	Image LUT-ID parameter	M
1	UBIN	LENGTH	X'01'	Length of the parameters to follow	M
2	CODE	LUTID	X'00' – X'FF'	LUT-ID identifier	M

If the Image LUT-ID parameter is not present, the default value for LUTID is zero for the standard LUT-ID.

Exception Conditions

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

EC-0003 **Invalid length**

Condition: The LENGTH value is not in the valid range.

EC-970F **Invalid sequence**

Condition: The Image LUT-ID parameter appeared out of sequence or more than once.

EC-9710 **Invalid or unsupported Image Data parameter value**

Condition: The Image LUT-ID parameter contains an invalid or unsupported value.

Image Structured Fields in MO:DCA-L Data Stream

MO:DCA-L has been retired and removed from the MO:DCA reference into a new book: *MO:DCA-L: The OS/2 Presentation Manager Metafile (.met) Format*. IOCA constructs that support MO:DCA-L have been retired.

MO:DCA-L was not used in printing. Encountering data specific for MO:DCA-L and not allowed in MO:DCA should generate an exception.

This section shows the syntax of IDD and IPD in the MO:DCA-L interchange set. An IOCA Image Segment is carried by one or more IPD structured fields.

IDD in MO:DCA-L Data Stream

Structured Field Introducer				
SF Length	X'D3A6FB'	Flags	Sequence Number	Data

Offset	Type	Name	Range	Meaning	M/O
0	CODE	UNITBASE	X'00' – X'01'	Unit base: **X'00'** 10 inches **X'01'** 10 centimeters All other values are reserved.	M
1–2	UBIN	XRESOL	X'0001' – X'7FFF'	Horizontal resolution in image points per unit base	M
3–4	UBIN	YRESOL	X'0001' – X'7FFF'	Vertical resolution in image points per unit base	M
5–6	UBIN	XSIZE	X'0001' – X'7FFF'	Horizontal size of the Image Presentation Space in image points	M
7–8	UBIN	YSIZE	X'0001' – X'7FFF'	Vertical size of the Image Presentation Space in image points	M

IPD in MO:DCA-L Data Stream

Structured Field Introducer				
SF Length	X'D3EEFB'	Flags	Sequence Number	IOCA Function Set 20

See “IOCA Function Set 20 (IOCA FS20)” on page 177 for details.

Note: An IOCA FS20 Image Segment can be split into multiple IPD structured fields. Data beyond the End Segment self-defining field is ignored by receivers.

IOCA Function Set 20 (IOCA FS20)

Function Set 20 was not used in printing. If a data stream specifies Function Set 20, products can either ignore it or generate an exception.

Function Set 20 describes bilevel, grayscale, and color images. This function set is carried by the MO:DCA-L controlling environment. The permissible parameter groupings in FS20 are defined as follows:

Table 31. Function Set 20 Structure

		X'70'	Begin Segment parameter		
		X'91'	Begin Image Content parameter		
+		X'94'	Image Size parameter		
+	[	X'95'	Image Encoding parameter		]
+	[	X'96'	IDE Size parameter		]
+	[	X'97'	Retired (Image LUT-ID parameter)		]
+	[	X'9B'	IDE Structure parameter		]
		X'FE92'	Image Data	(S)	
		X'93'	End Image Content parameter		
		X'71'	End Segment parameter		

Its acceptable self-defining fields and parameter values are shown in the following table.

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
Initial parameters:			
Begin Segment	ID (1)	X'70'	
	LENGTH (1)	X'00'	
Begin Image Content	ID (1)	X'91'	
	LENGTH (1)	X'01'	
	OBJTYPE (1)	X'FF'	IOCA
Image Size parameter	ID (1)	X'94'	
	LENGTH (1)	X'09'	
	UNITBASE (1)	X'00' – X'01'	
	HRESOL (2)	X'0000' – X'7FFF'	
	VRESOL (2)	X'0000' – X'7FFF'	
	HSIZE (2)	X'0000' – X'7FFF'	
	VSIZE (2)	X'0000' – X'7FFF'	
Image Encoding parameter	ID (1)	X'95'	
	LENGTH (1)	X'02'	
	COMPRID (1)	X'01', X'03', X'82'	**X'01'** IBM MMR-Modified Modified Read (see Note 1) **X'03'** No Compression **X'82'** G4 MMR-Modified Modified READ (see Note 1)

IOCA Self-defining Field	Parameter (Bytes)	Acceptable Value	Comments
	RECID (1)	X'01', X'03'	**X'01'** RIDIC **X'03'** Bottom-to-Top (see Note 2)
IDE Size parameter	ID (1)	X'96'	
	LENGTH (1)	X'01'	
	IDESZ (1)	X'01', X'04', X'08', X'18'	**X'01'** 1 bit/IDE **X'04'** 4 bits/IDE **X'08'** 8 bits/IDE **X'18'** 24 bits/IDE
Notes on the initial parameters: 1. IBM MMR-Modified Modified Read and G4 MMR-Modified Modified READ are applicable only to images whose IDE size is 1 bit/IDE; otherwise exception condition EC-9611 is raised. 2. Bottom-to-Top is used only in conjunction with No Compression; otherwise exception condition EC-9510 is raised.			
Parameters used when IDESZ=1:			
Retired	RESERVED (3)	X'970100', X'970101'	Retired Image LUT-ID parameter
Parameters used when IDESZ=4 or IDESZ=8:			
Retired	RESERVED (3)	X'970101'	Retired Image LUT-ID parameter
Parameters used when IDESZ=24:			
Retired	RESERVED (3)	X'970100'	Retired Image LUT-ID parameter
IDE Structure parameter	ID (1)	X'9B'	
	LENGTH (1)	X'08'	
	FLAGS (1)	X'00'	Additive and No gray coding
	FORMAT (1)	X'01'	RGB
	RESERVED (3)	X'000000'	Should be zero
	SIZE1 (1)	X'08'	8 bits of the IDE for the R component
	SIZE2 (1)	X'08'	8 bits of the IDE for the G component
	SIZE3 (1)	X'08'	8 bits of the IDE for the B component
Final parameters:			
Image Data	ID (2)	X'FE92'	
	LENGTH (2)	X'0001' – X'FFFF'	
	DATA	Any	IDEs
End Image Content	ID (1)	X'93'	
	LENGTH (1)	X'00'	
End Segment	ID (1)	X'71'	
	LENGTH (1)	X'00'	

Exception Condition EC-0002

Exception condition EC-0002 was an optional exception that, based on the definition of reserved fields, should never be returned by a receiver. Its definition was as follows:

EC-0002 Reserved bits or bytes are not zeros

Condition: Reserved bits or bytes are not zeros in the Image Data parameter within the Image Segment.

Note: This exception condition is optional.

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

AFPC and AFP Consortium are trademarks of the AFP Consortium.

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, in 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

These terms are trademarks or registered trademarks of the International Business Machines Corporation in the United States, in other countries, or both:

IBM
MVS

These terms are trademarks or registered trademarks of Adobe Systems Incorporated in the United States, in other countries, or both:

Acrobat
Adobe
Photoshop
Postscript

These terms are trademarks or registered trademarks of Hewlett-Packard Company in the United States, in other countries, or both:

Hewlett-Packard
PCL

PANTONE is a registered trademark of Pantone LLC.

Other company, product, or service names might be trademarks or service marks of others.

Glossary

This glossary contains terms that apply to the Advanced Function Presentation (AFP) Architecture and also terms that apply to other related presentation architectures.

Note: Only changes having to do with newly-added IOCA 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.

abstract profile. An ICC profile that represents abstract transforms and does not represent any device model. Color transformations using abstract profiles are performed from PCS to PCS. Abstract profiles cannot be embedded in images.

Abstract Syntax Notation One (ASN.1). A notation for defining data structures and data types. The notation is defined in international standard ISO/IEC 8824(E). See also object identifier.

ACK. See Positive Acknowledge Reply.

Acknowledge Reply. A printer-to-host reply that returns printer information or reports exceptions. An Acknowledge Reply can be positive or negative. See also Positive Acknowledge Reply and Negative Acknowledge Reply.

Acknowledgment Request. A request from the host for information from the printer. An example of an Acknowledgment Request is the use of the acknowledgment-required flag by a host system to request an Acknowledge Reply from an attached printer.

acknowledgment-required flag (ARQ). A flag that requests a printer to return an Acknowledge Reply. The acknowledgment-required flag is bit zero of an IPDS command's flag byte.

active coded font. The coded font that is currently being used by a product to process text.

additive primary colors. Red, green, and blue light, transmitted in video monitors and televisions. When used in various degrees of intensity and variation, they create all other colors of light; when superimposed equally, they create white. Contrast with subtractive primary colors.

addressable position. A position in a presentation space or on a physical medium that can be identified by a coordinate from the coordinate system of the presentation space or physical medium. See also picture element. Synonymous with position.

Advanced Function Presentation (AFP). An open architecture for the management of presentable information that is developed by the AFP Consortium (AFPC). AFP comprises a number of data stream and data object architectures:

- Mixed Object Document Content Architecture (MO:DCA); formerly referred to as AFPDS
- Intelligent Printer Data Stream (IPDS)
- AFP Line Data Architecture
- Bar Code Object Content Architecture (BCOCA)
- Color Management Object Content Architecture (CMOCA)
- Font Object Content Architecture (FOCA)
- Graphics Object Content Architecture for AFP (AFP GOCA)
- Image Object Content Architecture (IOCA)
- Metadata Object Content Architecture (MOCA)
- Presentation Text Object Content Architecture (PTOCA)

AEA. See alternate exception action.

AFM file. A file containing the metric information required for positioning the characters of a font. The metric information contained in this file was extracted from a PFB file, in an ASCII file format defined by Adobe® Systems Inc., and used for character positioning and page formatting.

AFP. See Advanced Function Presentation.

AFP archive. See AFP/A.

AFP Consortium (AFPC). A formal open standards body that develops and maintains AFP architecture. Information about the consortium can be found at `www.afpconsortium.org`.

AFP data stream. A presentation data stream that is processed in AFP environments. The MO:DCA architecture defines the strategic AFP interchange data stream. The IPDS architecture defines the strategic AFP printer data stream.

AFPDS. A term formerly used to identify the composed-page MO:DCA-based data stream interchanged in AFP environments. See also MO:DCA and AFP data stream.

AFP environment. Wherever the AFP architecture is used in any way; by an AFP vendor, an AFP customer, or any combination thereof.

AFP GOCA. A subset of the GOCA architecture, originally defined by IBM, specifically designed for AFP environments. See Graphics Object Content Architecture (GOCA).

AFP Line Data Architecture. An AFP architecture that controls formatting of line data using a Page Definition (PageDef).

AFP/A. A constrained version of the general MO:DCA architecture aimed at interoperability for AFP documents in an archiving system. Refer to the ISO 18565:2015 "Document management – AFP/Archive" standard for a complete definition of AFP/A.

AIAG. See Automotive Industry Action Group.

AIM. See Automatic Identification Manufacturers, Inc.

all points addressable (APA). The capability to address, reference, and position data elements at any addressable position in a presentation space or on a physical medium. Contrast with character cell addressable, in which the presentation space is divided into a fixed number of character-size rectangles in which characters can appear. Only the cells are addressable. An example of all points addressable is the positioning of text, graphics, and images at any addressable point on the physical medium. See also picture element.

alternate exception action (AEA). In the IPDS architecture, a defined action that a printer can take when a clearly defined, but unsupported, request is received. Control over alternate exception actions is specified by an Execute Order Anystate Exception-Handling Control command.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups. ANSI establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. It is the United States constituent body of the International Organization for Standardization (ISO).

anamorphic scaling. Scaling an object differently in the vertical and horizontal directions. See also scaling, horizontal font size, and vertical font size.

annotation. (1) A process by which additional data or attributes, such as highlighting, are associated with a page or a position on a page. Application of this data or attributes to the page is typically under the control of the user. Common functions such as applying adhesive removable notes to paper documents or using a transparent highlighter are emulated electronically by the annotation process. (2) A comment or explanation associated with the contents of a document component. An example of an annotation is a string of text that represents a comment on an image object on a page.

annotation link. In MO:DCA, a link type that specifies the linkage from a source document component to a target document component that contains an annotation.

annotation object. In MO:DCA, an object that contains an annotation. Objects that are targets of annotation links are annotation objects.

ANSI. See American National Standards Institute.

APA. See all points addressable.

append. In MO:DCA, an addition to or continuation of the contents of a document component. An example of an append is a string of text that is an addition to an existing string of text on a page.

append link. In MO:DCA, a link type that specifies the linkage from the end of a source document component to a target document component that contains an append.

append object. In MO:DCA, an object that contains an append. Objects that are targets of append links are append objects.

application. (1) The use to which an information system is put. (2) A collection of software components used to perform specific types of work on a computer.

application program. A program written for or by a user that applies to the user's work.

arc. A continuous portion of the curved line of a circle or ellipse. See also full arc.

architected. Identifies data that is defined and controlled by an architecture. Contrast with unarchitected.

archive interchange set. A constrained version of the general MO:DCA architecture aimed at interoperability for

AFP documents in an archiving system. For archive systems, the key requirement is to make each page stand alone by eliminating the use of resolution-dependent fonts and images, device-default fonts, and external resources. See AFP/A.

arc parameters. Variables that specify the curvature of an arc.

area. In GOCA, a set of closed figures that can be filled with a pattern or a color.

area filling. A method used to fill an area with a pattern or a color.

ARQ. See acknowledgment-required flag.

array. A structure that contains an ordered group of data elements. All elements in an array have the same data type.

article. The physical item that a bar code identifies.

ascender. The parts of certain lowercase letters, such as *b*, *d*, or *f*, that at zero-degree character rotation rise above the top edge of other lowercase letters such as *a*, *c*, and *e*. Contrast with descender.

ascender height. The character shape's most positive character coordinate system Y-axis value.

ASCII. Acronym for American Standard Code for Information Interchange. A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters.

ASN.1. See Abstract Syntax Notation One.

A space. The distance from the character reference point to the least positive character coordinate system X-axis value of the character shape. A-space can be positive, zero, or negative. See also B space and C space.

aspect ratio. (1) The ratio of the horizontal size of a picture to the vertical size of the picture. (2) In a bar code symbol, the ratio of bar height to symbol length.

asynchronous exception. Any exception other than those used to report a synchronous data-stream defect (action code X'01' or X'1F'), function no longer achievable (action code X'06'), or synchronous resource-storage problem (action code X'0C'). Asynchronous exceptions occur after the received page station. An example of an asynchronous exception is a paper jam. See also data-stream exception. Contrast with synchronous exception.

attribute. A property or characteristic of one or more constructs. See also character attribute, color attribute, current drawing attributes, default drawing attributes, line attributes, marker attributes, and pattern attributes.

audit CMR. A color management resource that reflects processing that has been done on an object.

Automatic Identification Manufacturers, Inc. (AIM). A trade organization consisting of manufacturers, suppliers, and users of bar codes.

Automotive Industry Action Group (AIAG). The coalition of automobile manufacturers and suppliers working to standardize electronic communications within the auto industry.

B

+B. Positive baseline direction.

B. See baseline direction.

background. (1) The part of a presentation space that is not occupied with object data. Contrast with foreground. (2) In GOCA, that portion of a graphics primitive that is mixed into the presentation space under the control of the current values of the background mix and background color attributes. (3) In GOCA, that portion of a character cell that does not represent a character. (4) In bar codes, the spaces, quiet zones, and area surrounding a printed bar code symbol.

background color. The color of a background. Contrast with foreground color.

background mix. (1) An attribute that determines how the color of the background of a graphics primitive is combined with the existing color of the graphics presentation space. (2) An attribute that determines how the points in overlapping presentation space backgrounds are combined. Contrast with foreground mix.

band. An arbitrary layer of an image. An image can consist of one or more bands of data.

bar. In bar codes, the darker element of a printed bar code symbol. See also element. Contrast with space.

bar code. An array of elements, such as bars, spaces, and two-dimensional modules that together represent data elements or characters in a particular symbology. The elements are arranged in a predetermined pattern following unambiguous rules defined by the symbology. See also bar code symbol.

Bar Code command set. In the IPDS architecture, a collection of commands used to present bar code symbols in a page, page segment, or overlay.

bar code density. The number of characters per inch (cpi) in a bar code symbology. In most cases, the range is three to ten cpi. See also character density, density, and information density.

bar code object area. The rectangular area on a logical page into which a bar code presentation space is mapped.

Bar Code Object Content Architecture (BCOCA). An architected collection of constructs used to interchange and present bar code data.

bar code presentation space. A two-dimensional conceptual space in which bar code symbols are generated.

bar code symbol. A combination of characters including start and stop characters, quiet zones, data characters, and check characters required by a particular symbology, that form a complete, scannable entity. See also bar code.

bar code symbology. A bar code language. Bar code symbologies are defined and controlled by various industry groups and standards organizations. Bar code symbologies are described in public domain bar code specification documents. Synonymous with symbology. See also Canadian Grocery Product Code (CGPC), European Article Numbering (EAN), Japanese Article Numbering (JAN), and Universal Product Code (UPC).

bar height. In bar codes, the bar dimension perpendicular to the bar width. Synonymous with bar length and height.

bar length. In bar codes, the bar dimension perpendicular to the bar width. Synonymous with bar height and height.

bar width. In bar codes, the thickness of a bar measured from the edge closest to the symbol start character to the trailing edge of the same bar.

bar width reduction. In bar codes, the reduction of the nominal bar width dimension on film masters or printing plates to compensate for systematic errors in some printing processes.

base-and-towers concept. A conceptual illustration of an architecture that shows the architecture as a base with optional towers. The base and the towers represent different degrees of function achieved by the architecture.

baseline. A conceptual line with respect to which successive characters are aligned. See also character baseline. Synonymous with printing baseline and sequential baseline.

baseline coordinate. One of a pair of values that identify the position of an addressable position with respect to the origin of a specified I,B coordinate system. This value is specified as a distance in addressable positions from the I axis of an I,B coordinate system. Synonymous with B coordinate.

baseline direction (B). The direction in which successive lines of text appear on a logical page. Synonymous with baseline progression and B direction.

baseline extent. A rectangular space oriented around the character baseline and having one dimension parallel to the character baseline. The space is measured along the Y axis of the character coordinate system. For bounded character boxes, the baseline extent at any rotation is its character coordinate system Y-axis extent. Baseline extent varies with character rotation. See also maximum baseline extent.

baseline increment. The distance between successive baselines.

baseline offset. The perpendicular distance from the character baseline to the character box edge that is parallel to the baseline and has the more positive character coordinate system Y-axis value. For characters entirely within the negative Y-axis region, the baseline offset can be zero or negative. An example is a subscript character. Baseline offset can vary with character rotation.

baseline presentation origin (B_o). The point on the B axis where the value of the baseline coordinate is zero.

baseline progression (B). The direction in which successive lines of text appear on a logical page. Synonymous with baseline direction and B direction.

base LND. The first Line Descriptor (LND) used to process an input line-data record. See also reuse LND.

base support level. Within the base-and-towers concept, the smallest portion of architected function that is allowed to be implemented. This is represented by a base with no towers. Synonymous with mandatory support level.

B axis. The axis of the I,B coordinate system that extends in the baseline or B direction. The B axis does not have to be parallel to the Y_p axis of its bounding X_p,Y_p coordinate space.

B_c. See current baseline presentation coordinate.

b_c. See current baseline print coordinate.

BCOCA. See Bar Code Object Content Architecture.

B coordinate. One of a pair of values that identify the position of an addressable position with respect to the origin of a specified I,B coordinate system. This value is specified as a distance in addressable positions from the I axis of an I,B coordinate system. Synonymous with baseline coordinate.

B direction (B). The direction in which successive lines of text appear on a logical page. Synonymous with baseline direction and baseline progression.

Bearer Bars. Bars that surround an Interleaved 2-of-5 bar code to prevent misreads and short scans that might occur when a skewed scanning beam enters or exits the bar code symbol through its top or bottom edge. When plates are used in the printing process, Bearer Bars help equalize the pressure exerted by the printing plate over the entire surface of the symbol to improve print quality. There are two styles: 1) four bars that completely surround the bar/space pattern and 2) two bars that are placed at the top and the bottom of the bar/space pattern.

Begin Segment Introducer (BSI). An IPDS graphics self-defining field that precedes all of the drawing orders in a graphics segment.

between-the-pels. The concept of pel positioning that establishes the location of a pel's reference point at the edge of the pel nearest to the preceding pel rather than through the center of the pel.

B extent. The extent in the B-axis direction of an I,B coordinate system. The B extent must be parallel to one of the axes of the coordinate system that contains the I,B coordinate system. The B extent is parallel to the Y_p extent when the B axis is parallel to the Y_p axis or to the X_p extent when the B axis is parallel to the X_p axis.

b_i. See initial baseline print coordinate.

big endian. A format for storage or transmission of binary data in which the most significant bit (or byte) is placed first. Contrast with little endian.

bilevel. Having two levels; for example, every point in a bilevel image has the value 1 or 0, representing a colored image point or empty space. Contrast with multilevel.

bilevel custom pattern. In GOCA, a custom pattern that is uncolored at definition time, then has a single color assigned to it when it is used to fill an area. Contrast with full-color custom pattern.

bilevel device. A device that is used in a manner that permits it to process two-level color data. Contrast with multilevel device.

BITS. A data type for architecture syntax, indicating one or more bytes to be interpreted as bit string information.

blend. A mixing rule in which the intersection of part of a new presentation space P_{new} with part of an existing presentation space $P_{existing}$ changes to a new color attribute that represents a color-mixing of the color attributes of P_{new} with the color attributes of $P_{existing}$. For example, if P_{new} has foreground color-attribute blue and $P_{existing}$ has foreground color-attribute yellow, the area where the two foregrounds intersect changes to a color attribute of green. See also mixing rule. Contrast with overpaint and underpaint.

B_o. See baseline presentation origin.

body. (1) On a printed page, the area between the top and bottom margins that can contain data. (2) In a book, the portion between the front matter and the back matter.

boldface. A heavy-faced type weight. Printing in a heavy-faced type weight.

boundary alignment. A method used to align image data elements by adding padding bits to each image data element.

bounded character box. A conceptual rectangular box, with two sides parallel to the character baseline, that circumscribes a character and is just large enough to contain the character, that is, just touching the shape on all four sides.

brightness. Attribute of a visual sensation according to which an area appears to exhibit more or less light.

BSI. See Begin Segment Introducer.

B space. The distance between the character coordinate system X-axis values of the two extremities of a character shape. See also A space and C space.

buffered pages. Pages and copies of pages that have been received but not yet reflected in committed page counters and copy counters.

BYTE. A data type for architecture syntax consisting of 8 bits and indicating that each byte has no predefined interpretation. Therefore, in CMOCA, each byte is interpreted as defined in the tag explanation.

C

calibration. To adjust the correct value of a reading by comparison to a standard.

Canadian Grocery Product Code (CGPC). The bar code symbology used to code grocery items in Canada.

cap-M height. The average height of the uppercase characters in a font. This value is specified by the designer of a font and is usually the height of the uppercase M.

Cartesian coordinate system. In a plane, an image coordinate system that has positive values for the X and Y axis in the top-right quadrant. The origin is the upper left-hand corner of the bottom-right quadrant. A pair of (x,y) values corresponds to one image point. Each image point is described by an image data element.

CCSID. See Coded Character Set Identifier.

CGCSGID. See Coded Graphic Character Set Global Identifier.

CGPC. See Canadian Grocery Product Code.

CHAR. A data type for architecture syntax, indicating one or more bytes to be interpreted as character information.

character. (1) A member of a set of elements used for the organization, control, or representation of data. A character can be either a graphic character or a control character. See also graphic character and control character. (2) In bar codes, a single group of bar code elements that represent an individual number, letter, punctuation mark, or other symbol.

character angle. The angle that is between the baseline of a character string and the horizontal axis of a presentation space or physical medium.

character attribute. A characteristic that controls the appearance of a character or character string.

character baseline. A conceptual reference line that is coincident with the X axis of the character coordinate system.

character box. A conceptual rectangular box with two sides parallel to the character baseline. A character's shape is formed within a character box by a presentation process, and the character box is then positioned in a presentation space or on a physical medium. The character box can be rotated before it is positioned.

character-box reference edges. The four edges of a character box.

character cell addressable. Allowing characters to be addressed, referenced, and positioned only in a fixed number of character-size rectangles into which a presentation space is divided. Contrast with all points addressable.

character cell size. The size of a rectangle in a drawing space used to scale font symbols into the drawing space.

character code. An element of a code page or a cell in a code table to which a character can be assigned. The element is associated with a binary value. The assignment of a character to an element of a code page determines the binary value that will be used to represent each occurrence of the character in a character string.

character coordinate system. An orthogonal coordinate system that defines font and character measurement distances. The origin is the character reference point. The X axis coincides with the character baseline.

character density. The number of characters per inch (cpi) in a bar code symbology. In most cases, the range is three to ten cpi. See also bar code density, density, and information density.

character direction. In GOCA, an attribute controlling the direction in which a character string grows relative to the inline direction. Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Synonymous with direction.

character escapement point. The point where the next character reference point is usually positioned. See also character increment and presentation position.

character identifier. The unique name for a graphic character.

character increment. The distance from a character reference point to a character escapement point. For each character, the increment is the sum of a character's A space, B space, and C space. A character's character increment is the distance the inline coordinate is incremented when that character is placed in a presentation space or on a physical medium. Character increment is a property of each graphic character in a font and of the font's character rotation.

character increment adjustment. In a scaled font, an adjustment to character increment values. The adjustment value is derived from the kerning track values for the font used to present the characters.

character metrics. Measurement information that defines individual character values such as height, width, and space. Character metrics can be expressed in specific fixed units, such as pels, or in relative units that are independent of both the resolution and the size of the font. Often included as part of the more general term font metrics. See also character set metrics and font metrics.

character origin. The point within the graphic pattern of a character that is to be aligned with the presentation position. See also character reference point.

character pattern. The scan pattern for a graphic character of a particular size, style, and weight.

character-pattern descriptor. Information that the printer needs to separate font raster patterns. Each character pattern descriptor is eight bytes long and specifies both the character box size and an offset value; the offset value permits the printer to find the beginning of the character raster pattern within the character raster pattern data for the complete coded font.

character positioning. A method used to determine where a character is to appear in a presentation space or on a physical medium.

character precision. The acceptable amount of variation in the appearance of a character on a physical medium from a specified ideal appearance, including no acceptable variation. Examples of appearance characteristics that can vary for a character are character shape and character position.

character reference point. The origin of a character coordinate system. The X axis is the character baseline. See also character origin.

character rotation. The alignment of a character with respect to its character baseline, measured in degrees in a clockwise direction. Examples are 0°, 90°, 180°, and 270°. Zero-degree character rotation exists when a character is in its customary alignment with the baseline. Character rotation and font inline sequence are related in that character rotation is a clockwise rotation; font inline sequence is a counter-clockwise rotation. Contrast with rotation.

character set. A finite set of different graphic characters or control characters that is complete for a given purpose. For example, the character set in ISO Standard 646, *7-Bit Coded Character Set for Information Processing Interchange*.

character set attribute. An attribute used to specify a coded font.

character set metrics. The measurements used in a font. Examples are height, width, and character increment for each character of the font. See also character metrics and font metrics.

character shape. The visual representation of a graphic character.

character shape presentation. A method used to form a character shape on a physical medium at an addressable position.

character shear. The angle of slant of a character cell that is not perpendicular to a baseline. Synonymous with shear.

character string. A sequence of characters.

check character. In bar codes, a character included within a bar code message whose value is used to perform a mathematical check to ensure the accuracy of that message. Synonymous with check digit.

check digit. In bar codes, a character included within a bar code message whose value is used to perform a mathematical check to ensure the accuracy of that message. Synonymous with check character.

CID file. A file containing the font information required for presenting the characters of a font. The shape information (glyph procedures) contained in this file is in a binary encoded format defined by Adobe Systems Inc., optimized for large character set fonts (for example, Japanese ideographic fonts having several thousand characters).

CIE. See Commission Internationale d'Éclairage.

CIELAB color space. Internationally accepted color space model used as a standard to define color within the graphic arts industry, as well as other industries. L*, a*, and b* are plotted at right angles to one another. Equal distances in the space represent approximately equal color difference.

CIEXYZ color space. The fundamental CIE-based color space that allows colors to be expressed as a mixture of the three tristimulus values X, Y, and Z.

CJK fonts. Fonts that contain a set of unified ideographic characters used in the written Chinese, Japanese, and Korean languages. The character encoding is the same for each language, but there might be glyph variants between languages.

clear area. A clear space that contains no machine-readable marks preceding the start character of a bar code symbol or following the stop character. Synonymous with quiet zone. Contrast with intercharacter gap and space.

clipping. Eliminating those parts of a picture that are outside of a clipping boundary such as a viewing window or presentation space. See also viewing window. Synonymous with trimming.

cluster-dot screening. A halftone method that uses multiple pixels that vary from small to large dots as the color gets darker. It is characterized by a polka-dot look.

CMAP file. A file containing the mapping of code points to the character index values used in a CID file. The code points conform to a particular character coding system that is used to identify the characters in a document data stream. The character index values are assigned in a CID file for identification of the glyph procedure used to define the character shape. The mapping information in this file is in an ASCII file format defined by Adobe Systems Inc.

CMOCA. See Color Management Object Content Architecture.

CMR. See color management resource.

CMY. Cyan, magenta, and yellow, the subtractive primary colors.

CMYK color space. (1) The color model used in four-color printing. Cyan, magenta, and yellow, the subtractive primary colors, are used with black to effectively create a multitude of other colors. (2) The primary colors used together in printing to effectively create a multitude of other colors: cyan, magenta, yellow, and black. Based on the subtractive color theory; the primary colors used in four-color printing processes.

Codabar. A bar code symbology characterized by a discrete, self-checking, numeric code with each character represented by a standalone group of four bars and the three spaces between them.

CODE. A data type for architecture syntax that indicates an architected constant to be interpreted as defined by the architecture.

Code 39. A bar code symbology characterized by a variable-length, bidirectional, discrete, self-checking, alphanumeric code. Three of the nine elements are wide and six are narrow. It is the standard for LOGMARS (the Department of Defense) and the AIAG.

Code 128. A bar code symbology characterized by a variable-length, alphanumeric code with 128 characters.

Coded Character Set Identifier (CCSID). A 16-bit number identifying a specific set consisting of an encoding scheme identifier, character set identifiers, code page identifiers, and other relevant information that uniquely identifies the coded graphic character representation used.

coded font. (1) A resource containing elements of a code page and a font character set, used for presenting text, graphics character strings, and bar code HRI. See also code page and font character set. (2) In FOCA, a resource containing the resource names of a valid pair of font character set and code page resources. The graphic character set of the font character set must match the graphic character set of the code page for the coded font resource pair to be valid. (3) In the IPDS architecture, a raster font resource containing code points that are directly paired to font metrics and the raster representation of character shapes, for a specific graphic character set. (4) In the IPDS architecture, a font resource containing descriptive information, a code page, font metrics, and a digital-technology representation of character shapes for a specific graphic character set.

coded font local identifier. A binary identifier that is mapped by the controlling environment to a named resource to identify a coded font. See also local identifier.

coded graphic character. A graphic character that has been assigned one or more code points within a code page.

coded graphic character set. A set of graphic characters with their assigned code points.

Coded Graphic Character Set Global Identifier (CGCSGID). A four-byte binary or a ten-digit decimal identifier consisting of the concatenation of a GCSGID and a CPGID. The CGCSGID identifies the code point assignments in the code page for a specific graphic character set, from among all the graphic characters that are assigned in the code page.

code page. (1) A resource object containing descriptive information, graphic character identifiers, and code points corresponding to a coded graphic character set. Graphic characters can be added over time; therefore, to specifically identify a code page, both a GCSGID and a CPGID should be used. See also coded graphic character set. (2) A set of assignments, each of which assigns a code point to a character. Each code page has a unique name or identifier. Within a given code page, a code point is assigned to one character. More than one character set can be assigned code points from the same code page. See also code point and section.

Code Page Global Identifier (CPGID). A unique code page identifier that can be expressed as either a two-byte binary or a five-digit decimal value.

code point. A unique bit pattern that can serve as an element of a code page or a site in a code table, to which a character can be assigned. The element is associated with a binary value. The assignment of a character to an element of a code page determines the binary value that will be used to represent each occurrence of the character in a character string. Code points are one or more bytes long. See also code table and section.

code table. A table showing the character allocated to each code point in a code. See also code page and code point.

color. A visual attribute of things that results from the light they emit, transmit, or reflect.

colorants. Colors (pigments, dyes, inks) used by a device, primarily a printer, to reproduce colors.

color attribute. An attribute that affects the color values provided in a graphics primitive, a text control sequence, or an IPDS command. Examples of color attributes are foreground color and background color.

color calibration. The process of altering the behavior of an input or output device to make it conform to an established state, specified by a manufacturer, user, industry specification, or standard.

color component. A dimension of a color value expressed as a numeric value. For example, a color value might consist of one, two, three, four, or eight components, also referred to as channels.

color conversion. The process of converting colors from one color space to another.

color image. Images whose image data elements are represented by multiple bits or whose image data element values are mapped to color values. Constructs that map image-data-element values to color values are look-up tables and image-data-element structure parameters. Examples of color values are screen color values for displays and color toner values for printers.

colorimetric intent. A gamut mapping method that is intended to preserve the relationships between in-gamut colors at the expense of out-of-gamut colors.

colorimetry. The science of measuring color and color appearance. Classical colorimetry deals primarily with color matches rather than with color appearance as such. The main focus of colorimetry has been the development of methods for predicting perceptual matches on the basis of physical measurements.

color management. The technology to calibrate the color of input devices (such as scanners or digital cameras), display devices, and output devices (such as printers or offset presses).

Color Management Object Content Architecture (CMOCA). An architected collection of constructs used for the interchange and presentation of the color management information required to render a print file, document, group of pages or sheets, page, overlay, or data object with color fidelity.

color management resource. An object that provides color management in presentation environments.

color management system. A set of software designed to increase the accuracy and consistency of color between color devices like a scanner, display, and printer.

color model. The method by which a color is specified. For example, the RGB color space specifies color in terms of three intensities for red (R), green (G), and blue (B). Also referred to as color space.

color of medium. The color of a presentation space before any data is added to it. Synonymous with reset color.

color palette. A system of designated colors that are used in conjunction with each other to achieve visual consistency.

Color Rendering Dictionary. A PostScript language construct for converting colors from the CIEXYZ color space to the device color space. It is analogous to the "from PCS" part of an ICC printer profile with one rendering intent; that is, the part used when the profile is a destination profile.

color space. The method by which a color is specified. For example, the RGB color space specifies color in terms of three intensities for red (R), green (G), and blue (B). Also referred to as color model.

ColorSpace conversion profile. An ICC profile that provides the relevant information to perform a color space transformation between the non-device color spaces and the Profile Connection Space. It does not represent any device model. ColorSpace conversion profiles can be embedded in images.

color table. A collection of color element sets. The table can also specify the method used to combine the intensity levels of each element in an element set to produce a specific color. Examples of methods used to combine intensity levels are the additive method and the subtractive method. See also color model.

column. A subarray consisting of all elements that have an identical position within the low dimension of a regular two-dimensional array.

command. (1) In the IPDS architecture, a structured field sent from a host to a printer. (2) In GOCA, a data-stream construct used to communicate from the controlling environment to the drawing process. The command introducer is environment dependent. (3) A request for system action.

command set. A collection of IPDS commands.

command-set vector. Information that identifies an IPDS command set and data level supported by a printer. Command-set vectors are returned with an Acknowledge Reply to an IPDS Sense Type and Model command.

Commission Internationale d'Éclairage (CIE). An association of international color scientists who produced the standards that are used as the basis of the description of color.

complex text layout. The typesetting of writing systems that require complex transformations between text input and text display for proper rendering on the screen or the printed page.

compression algorithm. An algorithm used to compress image data. Compression of image data can decrease the volume of data required to represent an image.

construct. An architected set of data such as a structured field or a triplet.

continuous code. A bar code symbology characterized by designating all spaces within the symbol as parts of characters, for example, Interleaved 2 of 5. There is no intercharacter gap in a continuous code. Contrast with discrete code.

continuous-form media. Connected sheets. An example of connected sheets is sheets of paper connected by a perforated tear strip. Contrast with cut-sheet media.

control character. (1) A character that denotes the start, modification, or end of a control function. A control character can be recorded for use in a subsequent action, and it can have a graphic representation. See also character. (2) A control function the coded representation of which consists of a single code point.

control instruction. A data construct transmitted from the controlling environment and interpreted by the environment interface to control the operation of the graphics processor.

controlled white space. White space caused by execution of a control sequence. See also white space.

controlling environment. The environment in which an object is embedded, for example, the IPDS and MO:DCA data streams.

control sequence. A sequence of bytes that specifies a control function. A control sequence consists of a control sequence introducer and zero or more parameters.

control sequence chaining. A method used to identify a sequential string of control sequences so they can be processed efficiently.

control sequence class. An assigned coded character that identifies a control sequence's syntax and how that syntax is to be interpreted. An example of a control sequence class is X'D3', that identifies presentation text object control sequences.

control sequence function type. The coded character occupying the fourth byte of an unchained control sequence introducer. This code defines the function whose semantics can be prescribed by succeeding control sequence parameters.

control sequence introducer. The information at the beginning of a control sequence. An unchained control sequence introducer consists of a control sequence prefix, a class, a length, and a function type. A chained control sequence introducer consists of a length and a function type.

control sequence length. The number of bytes used to encode a control sequence excluding the control sequence prefix and class.

control sequence prefix. The escape character used to identify a control sequence. The control sequence prefix is the first byte of a control sequence. An example of a control sequence prefix is X'2B'.

coordinates. A pair of values that specify a position in a coordinate space. See also absolute coordinate and relative coordinate.

coordinate system. A Cartesian coordinate system. An example is the image coordinate system that uses the fourth quadrant with positive values for the Y axis. The origin is the upper left-hand corner of the fourth quadrant. A pair of (x,y) values corresponds to one image point. Each image point is described by an image data element. See also character coordinate system.

copy control. A method used to specify the number of copies for a presentation space and the modifications to be made to each copy.

copy counter. Bytes in an Acknowledge Reply that identify the number of copies of a page that have passed a particular point in the logical paper path.

copy group. A set of copy subgroups that specify all copies of a sheet. In the IPDS architecture, a copy group is specified by a Load Copy Control command. In MO:DCA, a copy group is specified within a Medium Map. See also copy subgroup.

copy modification. The process of adding, deleting, or replacing data on selected copies of a presentation space.

copy set. A collection of pages intended to be printed multiple times. For example, when multiple copies of a book or booklet is printed, each copy of the book or booklet is a copy set. This term was originally used with copy machines to identify collections of copies that are delivered as sets or stapled as sets. The term was also used when printing multiple copies of an MVS™ data set.

copy subgroup. A part of a copy group that specifies a number of identical copies of a sheet and all modifications to those copies. Modifications include the media source, the media destination, medium overlays to be presented on the sheet, text suppressions, the number of pages on the sheet, and either simplex or duplex presentation. In the IPDS architecture, copy subgroups are specified by Load Copy Control command entries. In MO:DCA, copy subgroups are specified by repeating groups in the Medium Copy Count structured field in a Medium Map. See also copy group.

correlation. A method used in the IPDS architecture to match exceptions with commands.

correlation ID. A two-byte value that specifies an identifier of an IPDS command. The correlation ID is optional and is present only if bit one of the command's flag byte is B'1'.

CPGID. See Code Page Global Identifier.

cpi. Characters per inch.

C space. The distance from the most positive character coordinate system X-axis value of a character shape to the character escapement point. C-space can be positive, zero, or negative. See also A space and B space.

current baseline coordinate. The baseline presentation position at the present time. The baseline presentation position is the summation of the increments of all baseline controls since the baseline was established in the presentation space. The baseline presentation position is established in a presentation space either as part of the initialization procedures for processing an object or by an Absolute Move Baseline control sequence. Synonymous with current baseline presentation coordinate.

current baseline presentation coordinate (B_c). The baseline presentation position at the present time. The baseline presentation position is the summation of the increments of all baseline controls since the baseline was established in the presentation space. The baseline presentation position is established in a presentation space either as part of the initialization procedures for processing an object or by an Absolute Move Baseline control sequence. Synonymous with current baseline coordinate.

current baseline print coordinate (b_c). In the IPDS architecture, the baseline coordinate corresponding to the current print position on a logical page. The current baseline print coordinate is a coordinate in an I,B coordinate system. See also I,B coordinate system.

current drawing attributes. The set of attributes used at the present time to direct a drawing process. Contrast with default drawing attributes.

current drawing controls. The set of drawing controls used at the present time to direct a drawing process. Contrast with default drawing controls.

current inline coordinate. The inline presentation position at the present time. This inline presentation position is the summation of the increments of all inline controls since the inline coordinate was established in the presentation space. An inline presentation position is established in a presentation space either as part of the initialization procedures for processing an object or by an Absolute Move Inline control sequence. Synonymous with current inline presentation coordinate.

current inline presentation coordinate (I_c). The inline presentation position at the present time. This inline presentation position is the summation of the increments of all inline controls since the inline coordinate was established in the presentation space. An inline presentation position is established in a presentation space either as part of the initialization procedures for processing an object or by an Absolute Move Inline control sequence. Synonymous with current inline coordinate.

current inline print coordinate (i_c). In the IPDS architecture, the inline coordinate corresponding to the current print position on a logical page. The current inline print coordinate is a coordinate in an I,B coordinate system. See also I,B coordinate system.

current logical page. The logical page presentation space that is currently being used to process the data within a page object or an overlay object.

current position. The position identified by the current presentation space coordinates. For example, the coordinate position reached after the execution of a drawing order. See also current baseline presentation coordinate and current inline presentation coordinate. Contrast with given position.

custom line type value. A user-defined line type, defined by a series of pairs of a dash/dot length followed by a move length. Contrast with standard line type value.

custom pattern. In GOCA, a user-defined pattern, defined by the picture drawn by a series of drawing orders between a Begin Custom Pattern drawing order and an End Custom Pattern drawing order. Custom patterns can be either bilevel custom patterns or full-color custom patterns. Contrast with patterns in the default pattern set.

custom pattern mode. In GOCA, a mode that is entered when a Begin Custom Pattern drawing order is executed and exited when an End Custom Pattern drawing order is executed. While in this mode, drawing is done in a separate, temporary graphics presentation space rather than in the graphics presentation space of the current GOCA object.

cut-sheet media. Unconnected sheets. Contrast with continuous-form media.

D

data block. A deprecated term for object area.

data element. A unit of data that is considered indivisible.

data frame. A rectangular division of computer output on microfilm.

Data Map. A print control object in a Page Definition (PageDef) that establishes the page environment and specifies the mapping of line data to the page. Synonymous with Page Format.

data mask. A sequence of bits that can be used to identify boundary alignment bits in image data.

data object. In the IPDS architecture, a presentation-form object that is either specified within a page or overlay or is activated as a resource and later included in a page or overlay via the IDO command. Examples include: PDF single-page objects, Encapsulated PostScript objects, and IO Images. See also resource and data object resource.

data-object font. (1) In the IPDS architecture, a complete-font resource that is a combination of font components at a particular size, character rotation, and encoding. A data-object font can be used in a manner analogous to a coded font. The following useful combinations can be activated into a data-object font:

- A TrueType/OpenType font, an optional code page, and optional linked TrueType/OpenType objects; activated at a particular size, character rotation, and encoding
- A TrueType/OpenType collection, either an index value or a full font name to identify the desired font within the collection, an optional code page, and optional linked TrueType/OpenType objects; activated at a particular size, character rotation, and encoding

See also data-object-font component. (2) In the MO:DCA architecture, a complete non-FOCA font resource object that is analogous to a coded font. Examples of data-object fonts are TrueType fonts and OpenType fonts.

data-object-font component. In the IPDS architecture, a font resource that is either printer resident or is downloaded using object container commands. Data-object-font components are used as components of a data-object font. Examples of data-object-font components include TrueType/OpenType fonts and TrueType/OpenType collections. See also data-object font.

data object resource. In the IPDS architecture, an object-container resource or IO-Image resource that is either printer resident or downloaded. Data object resources can be:

- Used to prepare for the presentation of a data object; such as with a color management resource (CMR) or Resident Color Profile Resource
- Included in a page or overlay via the Include Data Object command; examples include: PDF single-page objects, Encapsulated PostScript objects, and IO Images

- Invoked from within a data object; examples include: PDF Resource objects and Non-OCA Resource objects

See also data object and resource.

data stream. A continuous stream of data that has a defined format. An example of a defined format is a structured field.

data-stream exception. In the IPDS architecture, a condition that exists when the printer detects an invalid or unsupported command, order, control, or parameter value from the host. Data-stream exceptions are those whose action code is X'01', X'19', or X'1F'. See also asynchronous exception and synchronous exception.

DBCS. See double-byte character set.

decoder. In bar codes, the component of a bar code reading system that receives the signals from the scanner, performs the algorithm to interpret the signals into meaningful data, and provides the interface to other devices. See also reader and scanner.

decryption. The process of taking encrypted data and converting it back into data that a human or a computer can read and understand. See also encryption

default. A value, attribute, or option that is assumed when none has been specified and one is needed to continue processing. See also default drawing attributes and default drawing controls.

default drawing attributes. The set of drawing attributes adopted at the beginning of a drawing process and usually at the beginning of each root segment that is processed. See also root segment. Contrast with current drawing attributes.

default drawing controls. The set of drawing controls adopted at the start of a drawing process and usually at the start of each root segment that is processed. See also root segment. Contrast with current drawing controls.

default indicator. A field whose bits are all B'1' indicating that a hierarchical default value is to be used. The value can be specified by an external parameter. See also external parameter.

default pattern set. In GOCA, a set of predefined patterns, like solid, dots, or horizontal lines. Contrast with custom pattern.

density. The number of characters per inch (cpi) in a bar code symbology. In most cases, the range is three to ten cpi. See also bar code density, character density, and information density.

deprecated. An architected construct is marked as "deprecated" to indicate that it should no longer be used because it has been superseded by a newer construct. Use or support of a deprecated construct is permitted but no longer recommended. Constructs are deprecated rather than immediately removed to provide backward compatibility.

descender. The part of the character that extends into the character coordinate system negative Y-axis region. Examples of letters with descenders at zero-degree character rotation are *g*, *j*, *p*, *q*, *y*, and *Q*. Contrast with ascender.

descender depth. The character shape's most negative character coordinate system Y-axis value.

design metrics. A set of quantitative values, recommended by a font designer, to describe the characters in a font.

design size. The size of the unit Em for a font. All relative font measurement values are expressed as a proportion of the design size. For example, the width of the letter *I* can be specified as one-fourth of the design size.

device attribute . A property or characteristic of a device.

Device-Control command set. In the IPDS architecture, a collection of commands used to set up a page, communicate device controls, and manage printer acknowledgment protocol.

device dependent. Dependent upon one or more device characteristics. An example of device dependency is a font whose characteristics are specified in terms of addressable positions of specific devices. See also system-level font resource.

device independent. Not dependent upon device characteristics.

device-independent color space. A CIE-based color space that allows color to be expressed in a device-independent way. It ensures colors to be predictably and accurately matched among various color devices.

device level font resource. A device-specific font object from which a presentation device can obtain the font information required to present character images.

device profile. A structure that provides a means of defining the color characteristics of a given device in a particular state.

device resolution. The number of pels that can be printed in an inch, both horizontally and vertically. This is the resolution that the printer uses when printing. Some printers can be configured to print with a variety of resolutions that can be selected by the operator. The device resolution can be different in the two directions (for example, a resolution of 360 by 720).

device-version code page. In the IPDS architecture, a device version of a code page contains all of the characters that were registered for the CPGID at the time the printer was developed; since then, more characters might have been added to the registry for that CPGID. A device-version code page is identified by a CPGID. See also code page.

digital halftoning. A method used to simulate gray levels on a bilevel device.

digital image. An image whose image data was sampled at regular intervals to produce a digital representation of the image. The digital representation is usually restricted to a specified set of values.

dimension. The attribute of size given to arrays and tables.

direction. In GOCA, an attribute that controls the direction in which a character string grows relative to the inline direction. Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top. Synonymous with character direction.

discrete code. A bar code symbology characterized by placing spaces that are not a part of the code between characters, that is, intercharacter gaps.

dispersed-dot halftone. Any halftone algorithm that turns on binary pixels individually without grouping them into clusters. The “smallest available” dots are scattered in a pseudorandom manner to print varying densities. Commonly contrasted with cluster-dot screening.

dither. An intentional form of noise added to an image to randomize quantization error. Dithering an image can prevent unwanted patterns from appearing within the image.

DOCS. See drawing order coordinate space.

document. (1) A machine-readable collection of one or more objects that represents a composition, a work, or a collection of data. (2) A publication or other written material.

document component. An architected part of a document data stream. Examples of document components are documents, pages, page groups, indexes, resource groups, objects, and process elements.

document-component hierarchy. In MO:DCA, an ordering of the document in terms of its lower-level components. The components are ordered by decreasing level as follows:

- Print file (highest level)
- Document
- Page group
- Page
- Data object (lowest level)

document content architecture. A family of architectures that define the syntax and semantics of the document component. See also document component and structured field.

document editing. A method used to create or modify a document.

document element. A self-identifying, variable-length, bounded record, that can have a content portion that provides control information, data, or both. An application or device does not have to understand control information or data to parse a data stream when all the records in the data stream are document elements. See also structured field.

document fidelity. The degree to which a document presentation preserves the creator's intent.

document formatting. A method used to determine where information is positioned in presentation spaces or on physical media.

document presentation. A method used to produce a visible copy of formatted information on physical media.

dot gain. The phenomenon that occurs when ink is transferred from the plate to the blanket of the press and finally to the paper on which it is being printed. A dot for a halftone or a screen gets larger because of the mechanical process of transferring ink.

dots per inch. (1) The number of dots that will fit in an inch. (2) A unit of measure for output resolution. (3) Dots per inch (dpi) is also used to measure the quality of input when using a scanner. In this case, dpi becomes a square function measuring the dots both vertically as well as horizontally. Consequently, when an image is scanned in at 300 dpi, there are 90,000 dots or bits of electronic data (300 x 300) in every square inch.

double-byte character set (DBCS). A character set that can contain up to 65536 characters.

double-byte coded font. A coded font in which the code points are two bytes long.

downloaded resource. In the IPDS architecture, a resource in a printer that is installed and removed under control of a host presentation services program. A downloaded resource is referenced by a host-assigned name that is valid for the duration of the session between the presentation services program and the printer. Contrast with resident resource.

dpi. See dots per inch.

drag. To use a pointing device to move an object. For example, clicking on a window border, and dragging it to make the window larger.

draw functions. Functions that can be done during the drawing of a picture. Examples of draw functions are displaying a picture, boundary computation, and erasing a graphics presentation space.

drawing control. A control that determines how a picture is drawn. Examples of drawing controls are arc parameters, transforms, and the viewing window.

drawing defaults. In GOCA, the set of attributes adopted at the start of each segment that is processed. These attributes are set either from standard defaults defined by the controlling environment or from the Set Current Defaults instruction that is contained in the Graphics Data Descriptor. Synonymous with default drawing attributes. Contrast with current drawing attributes.

drawing order. In GOCA, a graphics construct that the controlling environment builds to instruct a drawing processor about what to draw and how to draw it. The order can specify, for example, that a graphics primitive be drawn, a change to drawing attributes or drawing controls be effected, or a segment be called. One or more graphics primitives can be used to draw a picture. Drawing orders can be included in a structured field. See also order.

drawing order coordinate space (DOCS). A two-dimensional conceptual space in which graphics primitives are drawn, using drawing orders, to create pictures.

drawing process control. In GOCA, a control used by the graphics processor that determines how a picture is drawn. Examples of drawing process controls are arc parameters.

drawing processor. A graphics processor component that executes segments to draw a picture in a presentation space. See also segment, graphics presentation space, and image presentation space.

drawing units. Units of measurement used within a graphics presentation space to specify absolute and relative positions.

draw rule. A method used to construct a line, called a rule, between two specified presentation positions. The line that is constructed is either parallel to the inline I axis or baseline B axis.

duplex. A method used to print data on both sides of a sheet. Normal-duplex printing occurs when the sheet is turned over the Y_m axis. Tumble-duplex printing occurs when the sheet is turned over the X_m axis.

duplex printing. A method used to print data on both sides of a sheet. Contrast with simplex printing.

dynamic segment. A segment whose graphics primitives can be redrawn in different positions by dragging them from one position to the next across a picture without destroying the traversed parts of the picture.

E

EAN. See European Article Numbering.

EBCDIC. See Extended Binary-Coded Decimal Interchange Code.

Efficient XML Interchange (EXI). A format that allows XML documents to be encoded as binary data, rather than as plain text.

element. (1) A bar or space in a bar code character or a bar code symbol. (2) A structured field in a document content architecture data stream. (3) In GOCA, a portion of a segment consisting of either a single order or a group of orders enclosed in an element bracket, in other words, between a *begin* element and an *end* element. (4) A basic member of a mathematical or logical class or set.

Em. In printing, a unit of linear measure referring to the baseline-to-baseline distance of a font, in the absence of any external leading.

embedded ICC profile. ICC profiles that are embedded within graphic documents and images. An embedded ICC profile allows users to transparently move color data between different computers, networks and even operating systems without having to worry if the necessary profiles are present on the destination systems.

Em square. A square layout space used for designing each of the characters of a font.

encoding scheme. A set of specific definitions that describe the philosophy used to represent character data. The number of bits, the number of bytes, the allowable ranges of bytes, the maximum number of characters, and the meanings assigned to some generic and specific bit patterns, are some examples of specifications to be found in such a definition.

Encoding Scheme Identifier (ESID). A 16-bit number assigned to uniquely identify a particular encoding scheme specification. See also encoding scheme.

encryption. A process to manipulate data to achieve data security. To read an encrypted data string, access to key information that enables decryption of the data is required. See also decryption

environment interface. The part of the graphics processor that interprets commands and instructions from the controlling environment.

EPS. Acronym for Encapsulated PostScript. A standard file format for importing and exporting PostScript language files among applications in a variety of heterogeneous environments.

error diffusion halftone. A specific halftone method in which quantization errors are diffused spatially in a quasi-random manner.

escapement direction. In FOCA, the direction from a character reference point to the character escapement point, that is, the font designer's intended direction for successive character shapes. See also character direction and inline direction.

escape sequence. (1) In the IPDS architecture, the first two bytes of a control sequence. An example of an escape sequence is X'2BD3'. (2) A string of bit combinations that is used for control in code extension procedures. The first of these bit combinations represents the control function Escape.

ESID. See Encoding Scheme Identifier.

established baseline coordinate. The current baseline presentation coordinate when no temporary baseline exists or the last current baseline presentation coordinate that existed before the first active temporary baseline was created. If temporary baselines are created, the current baseline presentation coordinate coincides with the presentation coordinate of the most recently created temporary baseline.

European Article Numbering (EAN). The bar code symbology used to code grocery items in Europe.

exception. (1) An invalid or unsupported data-stream construct. (2) In the IPDS architecture, a condition requiring host notification. (3) In the IPDS architecture, a condition that requires the host to resend data. See also data-stream exception, asynchronous exception, and synchronous exception.

exception action. Action taken when an exception is detected.

exception condition. The condition that exists when a product finds an invalid or unsupported construct.

exchange. The predictable interpretation of shared information by a family of system processes in an environment where the characteristics of each process must be known to all other processes. Contrast with interchange.

EXI. See Efficient XML Interchange.

expanded. A type width that widens all characters of a typeface.

Extended Binary-Coded Decimal Interchange Code (EBCDIC). A coded character set that consists of eight-bit coded characters.

Extensible Markup Language (XML). A set of rules for encoding documents in a format that is both human-readable and machine-readable.

Extensible Metadata Platform (XMP). An ISO standard, originally created by Adobe Systems Incorporated, for the creation, processing, and interchange of standardized and custom metadata for all kinds of resources.

external leading. The amount of white space, in addition to the internal leading, that can be added to interline spacing without degrading the aesthetic appearance of a font. This value is usually specified by a font designer. Contrast with internal leading.

external parameter. A parameter for which the current value can be provided by the controlling environment, for example, the data stream, or by the application itself. Contrast with internal parameter.

F

factoring. The movement of a parameter value from one state to a higher-level state. This permits the parameter value to apply to all of the lower-level states unless specifically overridden at the lower level.

FGID. See Font Typeface Global Identifier.

filename map file. A file containing the mapping of object names to file names for use in establishing a font file system. Object names and file names do not conform to the same naming requirements, so it is necessary to provide a mapping between them. The mapping information in this file is in an ASCII file format defined by Adobe Systems Inc.

fillet. A curved line drawn tangential to a specified set of straight lines. An example of a fillet is the concave junction formed where two lines meet.

final form data. Data that has been formatted for presentation.

first read rate. In bar codes, the ratio of the number of successful reads on the first attempt to the total number of attempts made to obtain a successful read. Synonymous with read rate.

fixed medium information. Information that can be applied to a sheet by a printer or printer-attached device that is independent of data provided through the data stream. Fixed medium information does not mix with the data provided by the data stream and is presented on a sheet either before or after the text, image, graphics, or bar code data provided within the data stream. Fixed medium information can be used to create preprinted forms, or other types of printing, such as colored logos or letterheads, that cannot be created conveniently within the data stream.

fixed metrics. Graphic character measurements in physical units such as pels, inches, or centimeters.

FNN linked. In FOCA, the FNN (Font Name map) structured field permits the mapping of a set of IBM GCGIDs to the character index values that occur in either a CMAP file or a rearranged file. Because the set of GCGIDs and the set of character index values must correspond to the same set of characters, it is necessary to identify which CMAP or rearranged file (among the many that could be located in a font file system) is associated (linked) with the FNN structured field. Note that the Font Name Map is known as the Character ID Map in IPDS.

FOCA. See Font Object Content Architecture.

font. A set of graphic 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 class, and increment. Examples of fonts are fully described fonts, symbol sets, and their internal printer representations. See also coded font and symbol set.

font baseline extent. In the IPDS architecture, the sum of the uniform or maximum baseline offset and the maximum baseline descender of all characters in the font.

font character set. A FOCA resource containing descriptive information, font metrics, and the digital representation of character shapes for a specified graphic character set.

font control record. The record sent in an IPDS Load Font Control command to specify a font ID and other font parameters that apply to the complete font.

font height (FH). (1) A characteristic value, perpendicular to the character baseline, that represents the size of all graphic characters in a font. Synonymous with vertical font size. (2) In a font character set, nominal font height is a font-designer defined value corresponding to the nominal distance between adjacent baselines when character rotation is zero degrees and no external leading is used. This distance represents the baseline-to-baseline increment that includes the font's maximum baseline extent and the designer's recommendation for internal leading. The font designer can also define a minimum and a maximum vertical font size to represent the limits of scaling. (3) In font referencing, the specified font height is the desired size of the font when the characters are presented. If this size is different from the nominal vertical font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

font index. (1) The mapping of a descriptive font name to a font member name in a font library. An example of a font member in a font library is a font resource object. Examples of attributes used to form a descriptive font name are typeface, family name, point size, style, weight class, and width class. (2) In the IPDS architecture, an LF1-type raster-font resource containing character metrics for each code point of a raster font or raster-font section for a particular font inline sequence. There can be a font index for 0 degree, 90 degree, 180 degree, and 270 degree font inline sequences. A font index can be downloaded to a printer using the Load Font Index command. An LF1-type coded font or coded-font section is the combination of one fully described font and one font index. See also fully described font.

font inline sequence. The clockwise rotation of the inline direction relative to a character pattern. Character rotation and font inline sequence are related in that character rotation is a clockwise rotation; font inline sequence is a counter-clockwise rotation.

font local identifier. A binary identifier that is mapped by the controlling environment to a named resource to identify a font. See also local identifier.

font metrics. Measurement information that defines individual character values such as height, width, and space, as well as overall font values such as averages and maximums. Font metrics can be expressed in specific fixed units, such as pels, or in relative units that are independent of both the resolution and the size of the font. See also character metrics and character set metrics.

font modification parameters. Parameters that alter the appearance of a typeface.

font object. A resource object that contains some or all of the description of a font.

Font Object Content Architecture (FOCA). An architected collection of constructs used to describe fonts and to interchange those font descriptions.

font production. A method used to create a font. This method includes designing each character image, converting the character images to a digital-technology format, defining parameter values for each character, assigning appropriate descriptive and identifying information, and creating a font resource that contains the required information in a format that can be used by a text processing system. Digital-technology formats include bit image, vector drawing orders, and outline algorithms. Parameter values include such attributes as height, width, and escapement.

font referencing. A method used to identify or characterize a font. Examples of processes that use font referencing are document editing, document formatting, and document presentation.

Font Typeface Global Identifier (FGID). A unique font identifier that can be expressed as either a two-byte binary or a five-digit decimal value. The FGID is used to identify a type style and the following characteristics: posture, weight class, and width class.

font width (FW). (1) A characteristic value, parallel to the character baseline, that represents the size of all graphic characters in a font. Synonymous with horizontal font size. (2) In a font character set, nominal font width is a font-designer defined value corresponding to the nominal character increment for a font character set. The value is generally the width of the space character and is defined differently for fonts with different spacing characteristics.

- For fixed-pitch, uniform character increment fonts: the fixed character increment, that is also the space character increment
- For PSM fonts: the width of the space character
- For typographic, proportionally spaced fonts: one-third of the vertical font size, that is also the default size of the space character.

The font designer can also define a minimum and a maximum horizontal font size to represent the limits of scaling. (3) In font referencing, the specified font width is the desired size of the font when the characters are presented. If this size is different from the nominal horizontal font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

foreground. (1) The part of a presentation space that is occupied by object data. (2) In GOCA, the portion of a graphics primitive that is mixed into the presentation space under the control of the current value of the mix and color attributes. See also pel. Contrast with background.

foreground color. A color attribute used to specify the color of the foreground of a primitive. Contrast with background color.

foreground mix. An attribute used to determine how the foreground color of data is combined with the existing color of a graphics presentation space. An example of data is a graphics primitive. Contrast with background mix.

form. A division of the physical medium; multiple forms can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a form. Envelopes are an example of a physical medium that comprises only one form. The IPDS architecture defines four types of forms: cut-sheet media, continuous-form media, envelopes, and computer output on microfilm. Each type of form has a top edge. A form has two sides, a front side and a back side. Synonymous with sheet.

format. The arrangement or layout of data on a physical medium or in a presentation space.

formatter. A process used to prepare a document for presentation.

formblend. (1) In IPDS, this mixing rule is only used when a preprinted form overlay (PFO) is merged as presentation space P_{PFO} with other presentation data (presentation space P_{data}). The intersection of P_{PFO} and P_{data} is assigned the following color attribute:

- Wherever the color attribute of P_{PFO} is either color of medium, or “white” (CMYK = X'00000000' for a printer, RGB = X'FFFFFF' for an RGB display), the intersection is assigned the color attribute of P_{data}. Likewise, wherever the color attribute of P_{data} is either color of medium, or “white” (CMYK = X'00000000' for a printer, RGB = X'FFFFFF' for an RGB display), the intersection is assigned the color attribute of P_{PFO}.
- With other overlapping color values, the intersection assumes a new color attribute that is generated in a device-specific manner to simulate how the P_{data} color attribute would mix onto a preprinted form that has the color attribute of P_{PFO}. In general, this mixing is a blending of the color attributes of P_{data} and P_{PFO} that is determined by the two color attributes and by the print media and the print technology.

See also mixing rule. (2) In MO:DCA, this mixing rule is only used when a simulated preprinted form, which is simulated as either a Medium Preprinted Form overlay (M-PFO) or a PMC Preprinted Form overlay (PMC-PFO), is merged as a new presentation space P_n, onto an existing presentation space P_e. The intersection of the foregrounds of P_n and P_e is assigned the following color attribute:

- Wherever the color attribute of P_e is either the color of medium, or the color white (CMYK = X'00000000' or RGB = X'FFFFFF'), the intersection is assigned the color attribute of P_n.
- Wherever the color attribute of P_e is not the color of medium and not the color white, the intersection assumes a new color attribute that is generated in a device-specific manner to simulate how the P_e color attribute would mix onto a preprinted form that has the color attribute of P_n. In general, this mixing is a blending of the color attributes of P_n and P_e that is determined by the two color attributes and by the print media and the print technology.

Formdef. See Form Definition.

Form Definition (Formdef). A print control object that contains an environment definition and one or more Medium Maps. Synonymous with Form map.

Form Map. A print control object that contains an environment definition and one or more Medium Maps. Synonymous with Form Definition. See also Medium Map.

full arc. A complete circle or ellipse. See also arc.

full-color custom pattern. In GOCA, a custom pattern that has its colors completely assigned during its definition, and can therefore contain any number of colors. Contrast with bilevel custom pattern.

fully described font. In the IPDS architecture, an LF1-type raster-font resource containing font metrics, descriptive information, and the raster representation of character shapes, for a specific graphic character set. A fully described font can be downloaded to a printer using the Load Font Control and Load Font commands. An LF1-type coded font or coded-font section is the combination of one fully described font and one font index. See also font index.

function set. (1) A collection of architecture constructs and associated values. Function sets can be defined across or within subsets. (2) In the MO:DCA architecture, a formal extension to a MO:DCA interchange set that provides additional capabilities beyond those provided by the interchange set.

FW. See font width.

G

gamma. A measure of contrast in photographic images. More precisely, a parameter that describes the shape of the transfer function for one or more stages in an imaging pipeline. The transfer function is given by the expression output = input ^{gamma} where both input and output are scaled to the range 0 to 1.

gamut. In color reproduction, the subset of colors that can be accurately represented in a given circumstance, such as within a given color space or by a certain output device.

GCGID. See Graphic Character Global Identifier.

GCSGID. See Graphic Character Set Global Identifier.

GCUID. See Graphic Character UCS Identifier.

generic. Relating to, or characteristic of, a whole group or class.

GID. See global identifier.

GIF. See Graphic Interchange Format.

given position. The coordinate position at which drawing is to begin. A given position is specified in a drawing order. Contrast with current position.

GLC chain. The set of glyph layout control sequences used to present a set of glyphs. It consists of a GLC control sequence followed by one or more GIR/GAR/GOR control sequence groupings, wherein the GOR is always optional. These control sequences must be chained together using PTOCA chaining rules. No other control sequences can be interspersed within the GIR/GAR/GOR groupings or between the groupings. The GLC chain may be terminated by an optional UCT control sequence that carries the code points of the glyphs rendered by the GLC chain.

Global Identifier (GID). Any of the following:

- Coded Character Set Identifier (CCSID).
- Coded Graphic Character Set Global Identifier (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 MO:DCA, an encoded graphic character string that provides a reference name for a document element.
- Object identifier (OID)
- A Uniform Resource Locator (URL), as defined in RFC 1738, Internet Engineering Task Force (IETF), December, 1994

Global Resource Identifier (GRID). An eight-byte identifier that identifies a coded font resource. A GRID contains the following fields in the order shown:

1. GCSGID of a minimum set of graphic characters required for presentation. It can be a character set that is associated with the code page, or with the font character set, or with both.
2. CPGID of the associated code page
3. FGID of the associated font character set
4. Font width in 1440ths of an inch.

glyph. (1) A member of a set of symbols that represent data. Glyphs can be letters, digits, punctuation marks, or other symbols. Synonymous with graphic character. See also character. (2) In typography, a glyph is a particular graphical representation of a grapheme, or sometimes several graphemes in combination (a composed glyph), or only a part of a grapheme. In computing as well as typography, the term character refers to a grapheme or grapheme-like unit of text, as found in natural language writing systems (scripts). A character or grapheme is a unit of text, whereas a glyph is a graphical unit. TrueType/ OpenType fonts describe glyphs as a set of paths.

glyph advance. A glyph advance is the absolute displacement of a glyph's origin on the baseline in the inline direction from a specific point. In the context of complex text rendering using GLC chains, the specific point is the current text position at the beginning of the GLC chain.

glyph ID. A glyph ID is an index to a table entry in a TrueType/OpenType font that allows an application to retrieve the glyph's shape data.

glyph offset. A glyph offset is the offset of the glyph's origin from the current baseline in the baseline direction. In the context of complex text rendering using GLC chains, the current baseline is the baseline defined at the beginning of the GLC chain.

GOCA. See Graphics Object Content Architecture.

GPS. See graphics presentation space.

gradient. In GOCA, an area fill where one color gradually changes to another. A gradient is a type of pattern.

grapheme. (1) A minimally distinctive unit of writing in the context of a particular writing system. For example, å ("a + Combining Ring Above" or "Latin Small Letter A with Ring Above") is a grapheme in the Danish writing system. (2) What an end-user thinks of as a character. (3) In typography, a grapheme is the fundamental unit in written language. Graphemes include alphabetic letters, Chinese characters, numerals, punctuation marks, and all the individual symbols of any of the world's writing systems. In a typeface each character typically corresponds to a single glyph, but there are exceptions, such as a font used for a language with a large alphabet or complex writing system, where one character may correspond to several glyphs, or several characters to one glyph.

graphic arts. Image rich, customized content that is typically used for brochures and marketing documents.

graphic character. A member of a set of symbols that represent data. Graphic characters can be letters, digits, punctuation marks, or other symbols. Synonymous with glyph. See also character.

Graphic Character Global Identifier (GCGID). An alphanumeric character string used to identify a specific graphic character. A GCGID can be from four bytes to eight bytes long.

graphic character identifier. The unique name for a graphic character in a font or in a graphic character set. See also character identifier.

Graphic Character Set Global Identifier (GCSGID). A unique graphic character set identifier that can be expressed as either a two-byte binary or a five-digit decimal value.

Graphic Character UCS Identifier (GCUID). An alphanumeric character string used to identify a specific graphic character. The GCUID naming scheme is used for additional characters and sets of characters that exist in UNICODE; each GCUID begins with the letter *U* and ends with a UNICODE code point. The Unicode Standard is fully compatible with the earlier Universal Character Set (UCS) Standard.

Graphic Interchange Format (GIF). An image format type generated specifically for computer use. Its resolution is usually very low (72 dpi, or that of your computer screen), making it undesirable for printing purposes.

Graphics command set. In the IPDS architecture, a collection of commands used to present GOCA data in a page, page segment, or overlay.

graphics data. Data containing lines, arcs, markers, and other constructs that describe a picture.

graphics model space. A two-dimensional conceptual space in which a picture is constructed. All model transforms are completed before a picture is constructed in a graphics model space. Contrast with graphics presentation space. Synonymous with model space.

graphics object. An object that contains graphics data. See also object.

graphics object area. A rectangular area on a logical page into which a graphics presentation space window is mapped.

Graphics Object Content Architecture (GOCA). An architected collection of constructs used to interchange and present graphics data. GOCA was originally defined by IBM; this architecture is no longer used in AFP. Instead, a subset of GOCA was defined for use in AFP environments, called AFP GOCA. Usually when the term "GOCA" is used in AFP documentation, it means AFP GOCA.

graphics presentation space. A two-dimensional conceptual space in which a picture is constructed. In this space graphics drawing orders are defined. The picture can then be mapped onto an output medium. All viewing transforms are completed before the picture is generated for presentation on an output medium. An example of a graphics presentation space is the abstract space containing graphics pictures defined in an IPDS Write Graphics Control command. Contrast with graphics model space.

graphics presentation space window. The portion of a graphics presentation space that can be mapped to a graphics object area on a logical page.

graphics primitive. A basic construct used by an output device to draw a picture. Examples of graphics primitives are arc, line, fillet, character string, and marker.

graphics processor. The processing capability required to interpret a GOCA object, that is, to present the picture represented by the object. It includes the environment interface, that interprets commands and instructions, and the drawing processor, that interprets the drawing orders.

graphics segment. A set of graphics drawing orders contained within a Begin Segment command. See also segment.

grayscale. A means of specifying color using only one color component in shades of gray ranging from black to white.

grayscale image. Images whose image data elements are represented by multiple bits and whose image data element values are mapped to more than one level of brightness through an image data element structure parameter or a look-up table.

GRID. See Global Resource Identifier.

guard bars. The bars at both ends and the center of an EAN, JAN, or UPC symbol, that provide reference points for scanning.

gzip. A widely-used, free software compression algorithm.

H

HAID. See Host-Assigned ID.

halftone. A method of generating, on a press or laser printer, an image that requires varying densities or shades to accurately render the image. This is achieved by representing the image as a pattern of dots of varying size. Larger dots represent darker areas, and smaller dots represent lighter areas of an image.

hard object. An object that is mapped with a Map structured field in the environment group of a Form Map, page, or overlay, that causes the server to retrieve the object and send it to the presentation device. The object is then referenced for inclusion at a later time. Contrast with soft object.

height. In bar codes, the bar dimension perpendicular to the bar width. Synonymous with bar height and bar length.

hexadecimal. A number system with a base of sixteen. The decimal digits 0 through 9 and characters A through F are used to represent hexadecimal digits. The hexadecimal digits A through F correspond to the decimal numbers 10 through 15, respectively. An example of a hexadecimal number is X'1B', that is equal to the decimal number 27.

hierarchy. A series of elements that have been graded or ranked in some useful manner.

highlight color. A spot color that is used to accentuate or contrast monochromatic areas. See also spot color.

highlighting. The emphasis of displayed or printed information. Examples are increased intensity of selected characters on a display screen and exception highlighting on an IPDS printer.

hollow font. A font design in which the graphic character shapes include only the outer edges of the strokes.

home state. An initial IPDS operating state. A printer returns to home state at the end of each page, and after downloading a font, overlay, or page segment.

horizontal bar code. A bar code pattern presenting the axis of the symbol in its length dimension parallel to the X_{bc} axis of the bar code presentation space. Synonymous with picket fence bar code.

horizontal font size. (1) A characteristic value, parallel to the character baseline, that represents the size of all graphic characters in a font. Synonymous with font width. (2) In a font character set, nominal horizontal font size is a font-designer defined value corresponding to the nominal character increment for a font character set. The value is generally the width of the space character and is defined differently for fonts with different spacing characteristics.

- For fixed-pitch, uniform character increment fonts: the fixed character increment, that is also the space character increment
- For PSM fonts: the width of the space character
- For typographic fonts and proportionally spaced fonts: one-third of the vertical font size, that is also the default size of the space character.

The font designer can also define a minimum and a maximum horizontal font size to represent the limits of scaling. (3) In font referencing, the specified horizontal font size is the desired size of the font when the characters are presented. If this size is different from the nominal horizontal font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

horizontal scale factor. (1) In outline-font referencing, the specified horizontal adjustment of the Em square. The horizontal scale factor is specified in 1440ths of an inch. When the horizontal and vertical scale factors are different, anamorphic scaling occurs. See also vertical scale factor. (2) In FOCA, the numerator of a scaling ratio, determined by dividing the horizontal scale factor by the vertical font size. If the value specified is greater or less than the specified vertical font size, the graphic characters and their corresponding metric values are stretched or compressed in the horizontal direction relative to the vertical direction by the scaling ratio indicated.

host. (1) In the IPDS architecture, a computer that drives a printer. (2) In IOCA, the host is the controlling environment.

Host-Assigned ID (HAID). A two-byte ID in the range X'0001'–X'7EFF' that is assigned to an IPDS resource by a presentation-services program in the host. This ID uniquely identifies a resource until that resource is deactivated, in which case the HAID can be reused. HAIDs are used in IPDS resource management commands.

Host-Assigned Resource ID. The combination of a Host-Assigned ID with a section identifier, or a font inline sequence, or both. The section identifier and font inline sequence values are ignored for both page segments and overlays. See also section identifier and font inline sequence.

HRI. See human-readable interpretation.

HSV color space. (1) A transformation of the RGB color space that allow colors to be described in terms more natural to an artist. The name HSV stands for hue, saturation, and value. (2) Abbreviation for hue, saturation, and value (a color model used in some graphics programs). HSV must be translated to another model for color printing or for forming screen colors.

human-readable interpretation (HRI). The printed translation of bar code characters into equivalent Latin alphabetic characters, Arabic numeral decimal digits, and common special characters normally used for printed human communication.

hypermedia. Interlinked pieces of information consisting of a variety of data types such as text, graphics data, image, audio, and video.

hypertext. Interlinked pieces of information consisting primarily of text.

I

+I. Positive inline direction.

I. See inline direction.

I axis. The axis of an I,B coordinate system that extends in the inline direction. The I axis does not have to be parallel to the X_p axis of its bounding X_p,Y_p coordinate space.

I,B coordinate system. The coordinate system used to present graphic characters. This coordinate system is used to establish the inline direction and baseline direction for the placement of successive graphic characters within a presentation space. See also X_p,Y_p coordinate system.

I_c. See current inline presentation coordinate.

i_c. See current inline print coordinate.

ICC. See International Color Consortium.

ICC-absolute colorimetric. A rendering intent in which the chromatically adapted tristimulus values of the in-gamut colors are unchanged. It is useful for spot colors and when simulating one medium on another (proofing). Note that this definition of ICC-absolute colorimetry is actually called “relative colorimetry” in CIE terminology, since the data has been normalized relative to the perfect diffuser viewed under the same illumination source as the sample.

ICC DeviceLink profile. An ICC profile that provides a mechanism in which to save and store a series of device profiles and non-device profiles in a concatenated format as long as the series begins and ends with a device profile. This is useful for workflows where a combination of device profiles and non-device profiles are used repeatedly.

ICC profile. A file in the International Color Consortium profile format, containing information about the color reproduction capabilities of a device such as a scanner, a digital camera, a monitor, or a printer. An ICC profile includes three elements: 128-byte file header, tag table, and tagged element data. The intent of this format is to provide a cross-platform device profile format. Such device profiles can be used to translate color data created on one device into another device's native color space.

ID. Identifier. See also Host-Assigned ID (HAID), correlation ID, font control record, and overlay ID.

IDE. See image data element.

I direction. (1) The direction in which successive characters appear in a line of text. (2) In GOCA, the direction specified by the character angle attribute. Synonymous with inline direction.

IDP. See image data parameter.

IEEE. Institute of Electrical and Electronics Engineers.

I extent. The X_p extent when the I axis is parallel to the X_p axis or the Y_p extent when the I axis is parallel to the Y_p axis. The definition of the I extent depends on the X_p or Y_p extent because the I,B coordinate system is contained within an X_p,Y_p coordinate system.

i_i. See initial inline print coordinate.

illuminant. Something that can serve as a source of light.

image. An electronic representation of a picture produced by means of sensing light, sound, electron radiation, or other emanations coming from the picture or reflected by the picture. An image can also be generated directly by software without reference to an existing picture.

image block. A deprecated term for image object area.

image content. Image data and its associated image data parameters.

image coordinate system. An X,Y Cartesian coordinate system using only the fourth quadrant with positive values for the Y axis. The origin of an image coordinate system is its upper left hand corner. An X,Y coordinate specifies a presentation position that corresponds to one and only one image data element in the image content.

image data. Rectangular arrays of raster information that define an image.

image data element (IDE). A basic unit of image information. An image data element expresses the intensity of a signal at a corresponding image point. An image data element can use a look-up table to introduce a level of indirection into the expression of grayscale image or color image.

image data parameter (IDP). A parameter that describes characteristics of image data.

image distortion. Deformation of an image such that the original proportions of the image are changed and the original balance and symmetry of the image are lost.

image object. An object that contains image data. See also object.

image object area. A rectangular area on a logical page into which an image presentation space is mapped.

Image Object Content Architecture (IOCA). An architected collection of constructs used to interchange and present images.

image point. A discrete X,Y coordinate in the image presentation space. See also addressable position.

image presentation space (IPS). A two-dimensional conceptual space in which an image is generated.

image segment. Image content bracketed by Begin Segment and End Segment self-defining fields. See also segment.

IM Image. A migration image object that is resolution dependent, bi level, and cannot be compressed or scaled. Contrast with IO Image.

IM-Image command set. In the IPDS architecture, a collection of commands used to present IM-Image data in a page, page segment, or overlay.

immediate mode. The mode in which segments are executed as they are received and then discarded. Contrast with store mode.

indexed color. A color image format that contains a palette of colors to define the image. Indexed color can reduce file size while maintaining visual quality.

indexed object. An object in a MO:DCA document that is referenced by an Index Element structured field in a MO:DCA index. Examples of indexed objects are pages and page groups.

information density. The number of characters per inch (cpi) in a bar code symbology. In most cases, the range is three to ten cpi. See also bar code density, character density, and density.

initial addressable position. The values assigned to I_c and B_c by the data stream at the start of object state. The standard action values are I_o and B_o.

initial baseline print coordinate (b_i). The baseline coordinate of the first print position on a logical page. See also initial inline print coordinate.

initial inline print coordinate (i_i). The inline coordinate of the first print position on a logical page. See also initial baseline print coordinate.

inline-baseline coordinate system. See I,B coordinate system.

inline coordinate. The first of a pair of values that identifies the position of an addressable position with respect to the origin of a specified I,B coordinate system. This value is specified as a distance in addressable positions from the B axis of an I,B coordinate system.

inline direction (I). (1) The direction in which successive characters appear in a line of text. (2) In GOCA, the direction specified by the character angle attribute. Synonymous with I direction.

inline margin. The inline coordinate that identifies the initial addressable position for a line of text.

inline presentation origin (I_o). The point on the I axis where the value of the inline coordinate is zero.

inline resource. A resource object carried in a resource group that precedes all documents in an AFP print file.

input profile. An ICC profile that is associated with the image and describes the characteristics of the device on which the image was created.

instruction CMR. A color management resource that identifies processing that is to be done to an object.

Intelligent Printer Data Stream (IPDS). An architected host-to-printer data stream that contains both data and controls defining how the data is to be presented.

intensity. The extreme strength, degree, or amount of ink.

interchange. The predictable interpretation of shared information in an environment where the characteristics of each process need not be known to all other processes. Contrast with exchange.

interchange set. A defined set of MO:DCA function that describes a level of interchange.

intercharacter adjustment. Additional distance applied to a character increment that increases or decreases the distance between presentation positions, effectively modifying the amount of white space between graphic characters. The amount of white space between graphic characters is changed to spread the characters of a word for emphasis, distribute excess white space on a line among the words of that line to achieve right justification, or move the characters on the line closer together as in kerning. Examples of intercharacter adjustment are intercharacter increment and intercharacter decrement.

intercharacter decrement. Intercharacter adjustment applied in the negative I direction from the current presentation position. See also intercharacter adjustment.

intercharacter gap. In bar codes, the space between two adjacent bar code characters in a discrete code, for example, the space between two characters in Code 39. Synonymous with intercharacter space. Contrast with clear area, element, and space.

intercharacter increment. Intercharacter adjustment applied in the positive I direction from the current presentation position. See also intercharacter adjustment.

intercharacter space. In bar codes, the space between two adjacent bar code characters in a discrete code, for example, the space between two characters in Code 39. Synonymous with intercharacter gap. Contrast with element and space.

interleaved bar code. A bar code symbology in which characters are paired, using bars to represent the first character and spaces to represent the second. An example is Interleaved 2 of 5.

intermediate device. In the IPDS architecture, a device that operates on the data stream and is situated between a printer and a presentation services program in the host. Examples include devices that capture and cache resources and devices that spool the data stream.

internal leading. A font design parameter referring to the space provided between lines of type to keep ascenders separated from descenders and to provide an aesthetically pleasing interline spacing. The value of this parameter usually equals the difference between the vertical font size and the font baseline extent. Contrast with external leading.

internal parameter. In PTOCA, a parameter whose current value is contained within the object. Contrast with external parameter.

International Color Consortium (ICC). A group of companies chartered to develop, use, and promote cross-platform standards so that applications and devices can exchange color data without ambiguity.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

interoperability. The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units.

introducer. In GOCA, that part of the data stream passed from a controlling environment to a communication processor that indicates whether entities are to be processed in immediate mode or store mode. See also immediate mode and store mode.

I_o. See inline presentation origin.

IOCA. See Image Object Content Architecture.

IO Image. An image object containing IOCA constructs. Contrast with IM Image.

IO-Image command set. In the IPDS architecture, a collection of commands used to present IOCA data in a page, page segment, or overlay.

IPDS. See Intelligent Printer Data Stream.

IPDS dialog. A series of IPDS commands and IPDS Acknowledge Replies. An IPDS dialog begins with the first IPDS command that an IPDS device receives and ends either when an IPDS command explicitly ends the dialog or when the carrying-protocol session ends. There can be multiple independent sessions each with an IPDS dialog. See also session.

IPS. See image presentation space.

ISO. See International Organization for Standardization.

italics. A typeface with characters that slant upward to the right. In FOCA, italics is the common name for the defined inclined typeface posture attribute or parameter.

J

JAN. See Japanese Article Numbering.

Japanese Article Numbering (JAN). The bar code symbology used to code grocery items in Japan.

JFIF. See JPEG File Interchange Format.

jog. To cause printed sheets to be stacked in an output stacker offset from previously stacked sheets. Jogging is requested by using an IPDS Execute Order Anystate Alternate Offset Stacker command.

Joint Photographic Experts Group (JPEG). The Joint Photographic Experts Group (JPEG) is a standards committee that designed an image compression format. The compression format they designed is lossy, in that it deletes information from an image that it considers unnecessary. JPEG files can range from small amounts of lossless compression to large amounts of lossy compression.

JPEG. An image compression standard. See Joint Photographic Experts Group.

JPEG File Interchange Format (JFIF). (1) JPEG File Interchange Format (JFIF) is the most common file format for JPEG images. (TIFF is another file format that can be used to store JPEG images, and JNG is a third.) JFIF is not a formal standard; it was designed by a group of companies (though it is most often associated with C-Cube Microsystems, one of whose employees published it) and became a de facto industry standard. (2) Three-component JPEG images. RGB data is assumed without gamma correction and the APP0 marker is used to specify the resolution and optionally the thumbnail.

K

Kanji. A graphic character set for symbols used in Japanese ideographic alphabets.

kerning. The design of graphic characters so that their character boxes overlap, resulting in the reduction of space between characters. This allows characters to be designed for cursive languages, ligatures, and proportionally spaced fonts. An example of kerning is the printing of adjacent graphic characters so they overlap on the left or right side.

kerning track. A straight-line graph that associates vertical font size with white space adjustment. The result of this association is used to scale fonts.

kerning track intercept. The X-intercept of a kerning track for a given vertical font size or white space adjustment value.

kerning track slope. The slope of a kerning track.

key information. Bytes used by the decryption system to decrypt data that has been encrypted.

keyword. A two-part self-defining parameter consisting of a one-byte identifier and a one-byte value.

L

ladder bar code. A bar code pattern presenting the axis of the symbol in its length dimension parallel to the Y_{bc} axis of the bar code presentation space. Synonymous with vertical bar code.

LAN. See local area network.

landscape. A presentation orientation in which the X_m axis is parallel to the long sides of a rectangular physical medium. Contrast with portrait.

language. A set of symbols, conventions, and rules that is used for conveying information. See also pragmatics, semantics, and syntax.

LCID. See Local Character Set Identifier.

leading. A printer's term for the amount of space between lines of a printed page. Leading refers to the lead slug placed between lines of type in traditional typesetting. See also internal leading and external leading.

leading edge. (1) The edge of a character box that in the inline direction precedes the graphic character. (2) The front edge of a sheet as it moves through a printer.

legibility. Characteristics of presented characters that affect how rapidly, easily, and accurately one character can be distinguished from another. The greater the speed, ease, and accuracy of perception, the more legible the presented characters. Examples of characteristics that affect legibility are character shape, spacing, and composition.

LID. See local identifier.

ligature. A single glyph representing two or more characters. Examples of characters that can be presented as ligatures are *ff* and *ffi*.

linear gradient. In GOCA, a gradient where the color change takes place along a line. Contrast with radial gradient.

line art. An image that contains only black and white with no shades of gray.

line attributes. Those attributes that pertain to straight and curved lines. Examples of line attributes are line type and line width.

line data. Unformatted text data. Line data can be formatted using a Page Definition (PageDef).

line screen frequency. The measure of distance between the rows of dots that make up a halftone screen. Lower line screens are used on rougher, low quality printing substrates (such as newsprint), while higher line screens are used for high quality print jobs on smooth art papers.

lines per inch (lpi). (1) The number of lines per inch on a halftone screen. (2) Units used when measuring line screen frequency.

line type. A line attribute that controls the appearance of a line. The line type can either be a standard line type value or a custom line type value. Contrast with line width.

line width. A line attribute that controls the appearance of a line. Examples of line width are normal and thick. Contrast with line type.

link. A logical connection from a source document component to a target document component.

little endian. A bit or byte ordering where the right-most bits or bytes (those with a higher address) are most significant. Contrast with big endian.

Loaded-Font command set. In the IPDS architecture, a collection of commands used to load font information into a printer and to deactivate font resources.

local area network (LAN). A data network located on a user's premises in which serial transmission is used for direct data communication among data stations.

Local Character Set Identifier (LCID). A local identifier used as a character, marker, or pattern set attribute.

local identifier (LID). An identifier that is mapped by the controlling environment to a named resource.

location. A site within a data stream. A location is specified in terms of an offset in the number of structured fields from the beginning of a data stream, or in the number of bytes from another location within the data stream.

logical page. A presentation space. One or more object areas can be mapped to a logical page. A logical page has specifiable characteristics, such as size, shape, orientation, and offset. The shape of a logical page is the shape of a rectangle. Orientation and offset are specified relative to a medium coordinate system.

logical unit. A unit of linear measurement expressed with a unit base and units per unit-base value. For example, in MO:DCA and IPDS architectures, the following logical units are used:

- 1 logical unit = 1/1440 inch
  (unit base = 10 inches,
  units per unit base = 14,400)
- 1 logical unit = 1/240 inch
  (unit base = 10 inches,
  units per unit base = 2400)

Synonymous with L unit.

look-up table (LUT). (1) A table used to map one or more input values to one or more output values. (2) A logical list of colors or intensities. The list has a name and can be referenced to select a color or intensity. See also color table.

lossless. A form of image transformation in which all of the data is retained. Contrast with lossy.

lossy. A form of image transformation in which some of the data is lost. Contrast with lossless.

lowercase. Pertaining to small letters as distinguished from capital letters. Examples of small letters are *a*, *b*, and *g*. Contrast with uppercase.

lpi. See lines per inch.

L unit. A unit of linear measurement expressed with a unit base and units per unit-base value. For example, in MO:DCA and IPDS architectures, the following L units are used:

- 1 L unit = 1/1440 inch
  (unit base = 10 inches,
  units per unit base = 14,400)
- 1 L unit = 1/240 inch
  (unit base = 10 inches,
  units per unit base = 2400)

Synonymous with logical unit.

LUT. See look-up table.

Luv color space. The CIE color space in which L*, u* and v* are plotted at right angles to one another. Equal distances in the space represent approximately equal color difference.

M

magnetic ink character recognition (MICR). Recognition of characters printed with ink that contains particles of a magnetic material.

mainframe interactive (MFI). Pertaining to systems in which nonprogrammable terminals are connected to a mainframe.

mandatory support level. Within the base-and-towers concept, the smallest portion of architected function that is allowed to be implemented. This is represented by a base with no towers. Synonymous with base support level.

marker. A symbol with a recognizable appearance that is used to identify a particular location. An example of a marker is a symbol that is positioned by the center point of its cell.

marker attributes. The characteristics that control the appearance of a marker. Examples of marker attributes are size and color.

marker cell. A conceptual rectangular box that can include a marker symbol and the space surrounding that symbol.

marker precision. A method used to specify the degree of influence that marker attributes have on the appearance of a marker; this method has been made obsolete.

marker set. In GOCA, an attribute used to access a coded font.

marker symbol. A symbol that is used for a marker.

maximum ascender height. The maximum of the individual character ascender heights. A value for maximum ascender height is specified for each supported character rotation. Contrast with maximum descender depth.

maximum baseline extent. In FOCA, the sum of the maximum of the individual character baseline offsets and the maximum of the individual character descender depths, for a given font.

maximum descender depth. The maximum of the individual character descender depths. A value for maximum descender depth is specified for each supported character rotation. Contrast with maximum ascender height.

meaning. A table heading for architecture syntax. The entries under this heading convey the meaning or purpose of a construct. A meaning entry can be a long name, a description, or a brief statement of function.

measurement base. A base unit of measure from which other units of measure are derived.

media. Plural of medium. See also medium.

media destination. The destination to which sheets are sent as the last step in the print process. Some printers support several media destinations to allow options such as print job distribution to one or more specific destinations, collated copies without having to resend the document to the printer multiple times, and routing output to a specific destination for security reasons. Contrast with media source.

media-relative colorimetric. This rendering intent rescales the in-gamut, chromatically-adapted tristimulus values such that the white point of the actual medium is mapped to the PCS white point (for either input or output). It is useful for colors that have already been mapped to a medium with a smaller gamut than the reference medium (and therefore need no further compression).

media source. The source from which sheets are obtained for printing. Some printers support several media sources so that media with different characteristics (such as size, color, and type) can be selected when desired. Contrast with media destination.

medium. A two-dimensional conceptual space with a base coordinate system from which all other coordinate systems are either directly or indirectly derived. A medium is mapped onto a physical medium in a presentation-system-dependent manner. Synonymous with medium presentation space. See also logical page, physical medium, and presentation space.

Medium Map. A print control object in a Form Map that defines resource mappings and controls modifications to a form, page placement on a form, and form copy generation. See also Form Map.

medium preprinted form overlay (M-PFO). In MO:DCA, a PFO that is designed to simulate a preprinted form for a sheet-side. An M-PFO is invoked with the MMC structured field and is applied last to the medium presentation space after all other data for the sheet-side has been applied.

medium presentation space. A two-dimensional conceptual space with a base coordinate system from which all other coordinate systems are either directly or indirectly derived. A medium presentation space is mapped onto a physical medium in a presentation-system-dependent manner. Synonymous with medium. See also logical page, physical medium, and presentation space.

metadata. Descriptive information that is associated with and augments other data.

Metadata command set. In the IPDS architecture, a collection of commands used to associate MOCA data with objects.

metadata object. In AFP, the resource object that carries metadata.

Metadata Object Content Architecture (MOCA). A resource object architecture to carry metadata that serves to provide context or additional information about an AFP object or other AFP data.

MFI. See mainframe interactive.

MICR. See magnetic ink character recognition.

Microfilm frame. A rectangular area on the microfilm bounded by imaginary, intersecting grid lines within which a data frame may be recorded. The grid lines are part of gauges used for checking microfilm, but they do not actually appear on the microfilm.

mil. 1/1000 inch.

mix. A method used to determine how the color of a graphics primitive is combined with the existing color of a graphics presentation space. See also foreground mix and background mix.

Mixed Object Document Content Architecture (MO:DCA). An architected, presentation-system-independent data stream for interchanging documents.

mixing. (1) Combining foreground and background of one presentation space with foreground and background of another presentation space in areas where the presentation spaces intersect. (2) Combining foreground and background of multiple intersecting object data elements in the object presentation space.

mixing rule. A method for specifying the color attributes of the resulting foreground and background in areas where two presentation spaces intersect.

M/O. A table heading for architecture syntax. The entries under this heading indicate whether the construct is mandatory (M) or optional (O).

MOCA. See Metadata Object Content Architecture.

MO:DCA. See Mixed Object Document Content Architecture.

MO:DCA GA. A MO:DCA function set that supports levels of PDF used in graphic arts printing.

MO:DCA IS/1. MO:DCA Interchange Set 1. A subset of MO:DCA that defines an interchange format for presentation documents.

MO:DCA IS/2. MO:DCA Interchange Set 2. A retired subset of MO:DCA that defines an interchange format for presentation documents.

MO:DCA IS/3. MO:DCA Interchange Set 3. A subset of MO:DCA that defines an interchange format for print files that supersedes MO:DCA IS/1.

MO:DCA-L. A MO:DCA subset that defines the OS/2 Presentation Manager (PM) metafile. This format is also known as a .met file. The definition of this MO:DCA subset is stabilized and is no longer being developed as part of the MO:DCA architecture. It is defined in the document *MO:DCA-L: The OS/2 Presentation Manager Metafile (.met) Format*, available at `www.afpconsortium.org`.

MO:DCA-P. A subset of the MO:DCA architecture that defines presentation documents. This term is now synonymous with the term MO:DCA.

model space. A two-dimensional conceptual space in which a picture is constructed. All model transforms are completed before a picture is constructed in a graphics model space. Contrast with graphics presentation space. Synonymous with graphics model space.

model transform. A transform that is applied to drawing-order coordinates. Contrast with viewing transform.

module. In a bar code symbology, the nominal width of the smallest element of a bar or space. Actual bar code symbology bars and spaces can be a single module wide or some multiple of the module width. The multiple need not be an integer.

modulo-N check. A check in which an operand is divided by a modulus to generate a remainder that is retained and later used for checking. An example of an operand is the sum of a set of digits. See also modulus.

modulus. In a modulo check, the number by which an operand is divided. An example of an operand is the sum of a set of digits. See also modulo-N check.

monochrome. A single color. Monochrome usually refers to a black-and-white image. Also referred to as line art or bitmap mode in Adobe Photoshop®. See also bilevel.

monospaced font. A font with graphic characters having a uniform character increment. The distance between reference points of adjacent graphic characters is constant in the escapement direction. The blank space between the graphic characters can vary. Synonymous with uniformly spaced font. Contrast with proportionally spaced font and typographic font.

move order. A drawing order that specifies or implies movement from the current position to a given position. See also current position and given position.

M-PFO. See medium preprinted form overlay (M-PFO).

multilevel. Having multiple levels; for example, every point in a multilevel image can have values from 0 to n, where n is greater than 1. Contrast with bilevel.

multilevel device. A device that is used in a manner that permits it to process color data of more than two levels. Contrast with bilevel device.

N

NACK. See Negative Acknowledge Reply.

name. A table heading for architecture syntax. The entries under this heading are short names that give a general indication of the contents of the construct.

named color. A color that is specified with a descriptive name. An example of a named color is “green”.

navigation. The traversing of a document based on links between contextually related document components.

navigation link. A link type that specifies the linkage from a source document component to a contextually related target document component. Navigation links can be used to support applications such as hypertext and hypermedia.

nColor color model. The color model used in IOCA images that contain color components that typically do not match any of the standard AFP color models, such as RGB and CMYK. Often, such images contain more than four color components, although the number of color components can be anywhere from two to fifteen, inclusive.

Negative Acknowledge Reply (NACK). In the IPDS architecture, a reply from a printer to a host, indicating that an exception has occurred. Contrast with Positive Acknowledge Reply.

neighborhood-operation halftone. Halftone algorithm that transfers the quantization error due to thresholding to the unhalftoned neighbors of the current pixel. Error diffusion is a neighborhood operation since it operates not only on the input pixel, but also its neighbors. Contrast with point-operation halftone.

nested resource. A resource that is invoked within another resource using either an Include command or a local ID. See also nesting resource.

nesting coordinate space. A coordinate space that contains another coordinate space. Examples of coordinate spaces are medium, overlay, page, and object area.

nesting resource. A resource that invokes nested resources. See also nested resource.

neutral white. A color attribute that gives a presentation-system-dependent default color, typically white on a screen and black on a printer. Note that neutral white and color of medium are two different colors.

non-presentation object. An object that is not a presentation object.

nonprocess runout (NPRO). An operation that moves sheets of physical media through the printer without printing on them. This operation is used to stack the last printed sheet.

no operation (NOP). A construct whose execution causes a product to proceed to the next instruction to be processed without taking any other action.

NOP. See no operation.

normal-duplex printing. Duplex printing that simulates the effect of physically turning the sheet around the Y_m axis.

NPRO. See nonprocess runout.

N-up. The partitioning of a side of a sheet into a fixed number of equal size partitions. For example, 4-up divides each side of a sheet into four equal partitions.

O

object. (1) A collection of structured fields. The first structured field provides a begin-object function, and the last structured field provides an end-object function. The object can contain one or more other structured fields whose content consists of one or more data elements of a particular data type. An object can be assigned a name, that can be used to reference the object. Examples of objects are presentation text, font, graphics, and image objects. (2) Something that a user works with to perform a task.

object area. A rectangular area in a presentation space into which a data object is mapped. The presentation space can be for a page or an overlay. Examples are a graphics object area, an image object area, and a bar code object area.

object data. A collection of related data elements bundled together. Examples of object data include graphic characters, image data elements, and drawing orders.

object identifier (OID). (1) A notation that assigns a globally unambiguous name to an object or a document component. The notation is defined in international standard ISO/IEC 8824(E). (2) A variable length (2-bytes long to 129-bytes long) binary ID that uniquely identifies an object. OIDs use the ASN.1 definite-short-form object identifier format defined in the ISO/IEC 8824:1990(E) international standard and described in the MO:DCA Registry Appendix of the *Mixed Object Document Content Architecture Reference*. An OID consists of a one-byte identifier (X'06'), followed by a one-byte length (between X'00' and X'7F'), followed by 0–127 content bytes.

obsolete. Removed from the architecture, and thus ignored by receivers.

OCR A. See Optical Character Recognition A.

OCR B. See Optical Character Recognition B.

offline. A device state in which the device is not under the direct control of a host. Contrast with online.

offset. A table heading for architecture syntax. The entries under this heading indicate the numeric displacement into a construct. The offset is measured in bytes and starts with byte zero. Individual bits can be expressed as displacements within bytes.

OID. See object identifier.

online. A device state in which the device is under the direct control of a host. Contrast with offline.

opacity. In bar codes, the optical property of a substrate material that minimizes showing through from the back side or the next sheet.

Optical Character Recognition A (OCR A). A font containing the character set in ANSI standard X3.17-1981, that contains characters that are both human readable and machine readable.

Optical Character Recognition B (OCR B). A font containing the character set in ANSI standard X3.49-1975, that contains characters that are both human readable and machine readable.

order. (1) In GOCA, a graphics construct that the controlling environment builds to instruct a drawing processor about what to draw and how to draw it. The order can specify, for example, that a graphics primitive be drawn, a change to drawing attributes or drawing controls be effected, or a segment be called. One or more graphics primitives can be used to draw a picture. Orders can be included in a structured field. Synonymous with drawing order. (2) In the IPDS architecture, a construct within an execute-order command. (3) In IOCA, a functional operation that is performed on the image content.

ordered page. In the IPDS architecture, a logical page that does not contain any page segments or overlays, and in which all text data and all image, graphics, and bar code objects are ordered. The order of the data objects is such that physical pel locations on the physical medium are accessed by the printer in a sequential left-to-right and top-to-bottom manner, where these directions are relative to the top edge of the physical medium. Once a physical pel location has been accessed by the printer, the page data does not require the printer to access that same physical pel location again.

orientation. The angular distance a presentation space or object area is rotated in a specified coordinate system, expressed in degrees and minutes. For example, the orientation of printing on a physical medium, relative to the X_m axis of the X_m,Y_m coordinate system. See also presentation space orientation and text orientation.

origin. The point in a coordinate system where the axes intersect. Examples of origins are the addressable position in an X_m,Y_m coordinate system where both coordinate values are zero and the character reference point in a character coordinate system.

orthogonal. Intersecting at right angles. An example of orthogonal is the positional relationship between the axes of a Cartesian coordinate system.

outline font. A shape technology in which the graphic character shapes are represented in digital form by a series of mathematical expressions that define the outer edges of the strokes. The resultant graphic character shapes can be either solid or hollow.

output profile. An ICC profile that describes the characteristics of the output device for which the image is destined. The profile is used to color match the image to the device's gamut.

overhead. In a bar code symbology, the fixed number of characters required for starting, stopping, and checking a bar code symbol.

overlay. (1) A resource object that contains presentation data such as, text, image, graphics, and bar code data. Overlays define their own environment and are often used as pre-defined pages or electronic forms. Overlays are classified according to how they are presented with other presentation data: a medium overlay is positioned at the origin of the medium presentation space before any pages are presented, and a page overlay is positioned at a specified point in a page's logical page. A Page Modification Control (PMC) overlay is a special type of page overlay used in MO:DCA environments. (2) The final representation of such an object on a physical medium. Contrast with page segment.

Overlay command set. In the IPDS architecture, a collection of commands used to load, deactivate, and include overlays.

overlay ID. A one-byte ID assigned by a host to an overlay. Overlay IDs are used in IPDS Begin Overlay, Deactivate Overlay, Include Overlay, and Load Copy Control commands.

overlay state. An operating state that allows overlay data to be downloaded to a product. For example, a printer enters overlay state from home state when the printer receives an IPDS Begin Overlay command.

overpaint. A mixing rule in which the intersection of part of a new presentation space P_{new} with an existing presentation space $P_{existing}$ keeps the color attribute of P_{new}. This is also referred to as “opaque” mixing. See also mixing rule. Contrast with blend and underpaint.

overscore. A line parallel to the baseline and placed above the character.

overstrike. In PTOCA, the presentation of a designated character as a string of characters in a specified text field. The intended effect is to make the resulting presentation appear as though the text field, whether filled with characters or blanks, has been marked out with the overstriking character.

overstriking. The method used to merge two or more graphic characters at the same addressable position in a presentation space or on a physical medium.

P

page. (1) A data stream object delimited by a Begin Page structured field and an End Page structured field. A page can contain presentation data such as text, image, graphics, and bar code data. (2) The final representation of a page object on a physical medium.

page counter. Bytes in an IPDS Acknowledge Reply that specify the number of pages that have passed a particular point in a logical paper path.

PageDef. See Page Definition.

Page Definition (PageDef). A print control object used to format line data into page data. A Page Definition contains one or more Data Maps and may optionally specify conditional processing of the line data. Synonymous with Page Map. See also Data Map.

Page Format. Synonymous with Data Map.

page group. A named group of sequential pages. A page group is delimited by a Begin Named Page Group structured field and an End Named Page Group structured field. A page group can contain nested page groups. All pages in the page group inherit the attributes and processing characteristics that are assigned to the page group.

Page Map. A print control object used to format line data into page data. A Page Map contains one or more Data Maps and may optionally specify conditional processing of the line data. Synonymous with Page Definition. See also Data Map.

page segment. (1) In the IPDS architecture, a resource object that can contain text, image, graphics, and bar code data. Page segments do not define their own environment, but are processed in the existing environment. (2) In MO:DCA, a resource object that can contain any mixture of bar code objects, graphics objects, and 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.

Page-Segment command set. In the IPDS architecture, a collection of commands used to load, deactivate, and include page segments.

page-segment state. An operating state that makes page-segment data available to a product. For example, a printer enters page-segment state from home state when it receives an IPDS Begin Page Segment command.

page state. In the IPDS architecture, an operating state that makes page data available to a product. For example, a printer enters page state from home state when it receives an IPDS Begin Page command.

paginated object. A data object that can be rendered on a single page or overlay. An example of a paginated object is a single image in a multi-image TIFF file.

palette. The collection of colors or shades available to a graphics system or program.

PANTONE®. The proprietary PANTONE color matching system is the most popular method of specifying extra colors—not out of the CMYK four color process—for print. PANTONE colors are numbered and mixed from a base set of colors. By specifying a specific PANTONE color, a designer knows that there is little chance of color variance on the presses.

parameter. (1) A variable that is given a constant value for a specified application. (2) A variable used in conjunction with a command to affect its result.

partition. Dividing the medium presentation space into a specified number of equal-sized areas in a manner determined by the current physical media.

partitioning. A method used to place parts of a control into two or more segments or structured fields. Partitioning can cause difficulties for a receiver if one of the segments or structured fields is not received or is received out of order.

pattern. An array of symbols used to fill an area.

pattern attributes. The characteristics that specify the appearance of a pattern.

pattern reference point. In GOCA, a position in the graphics presentation space to be used as the origin of a custom pattern; the pattern is tiled in all directions from this position.

pattern set. An attribute in GOCA used to access a symbol set or coded font.

pattern symbol. The geometric construct that is used repetitively to generate a pattern. Examples of pattern symbols are dots, squares, and triangles.

PCL®. A set of printer commands, developed by Hewlett-Packard®, that provide access to printer features.

PCS. (1) See Print Contrast Signal. (2) See Profile Connection Space.

PDF. An acronym for Acrobat® Portable Document Format. PDF files are cross platform and contain all of the image and font data. Design attributes are retained in a compressed single package.

pel. The smallest printable or displayable unit on a physical medium. In computer graphics, the smallest element of a physical medium that can be independently assigned color and intensity. Pels per inch is often used as a measurement of presentation granularity. Synonymous with picture element and pixel.

perceptual rendering intent. The exact gamut mapping of the perceptual rendering intent is vendor specific and involves compromises such as trading off preservation of contrast in order to preserve detail throughout the tonal range. It is useful for general reproduction of images, particularly pictorial or photographic-type images.

PFB file. A file containing the font information required for presenting the characters of a font. The shape information (glyph procedures) contained in this file is in a binary encoded format defined by Adobe Systems Inc., optimized for small character set fonts having one to two hundred characters (for example, English, Greek, and Cyrillic).

PFO. See preprinted form overlay (PFO).

physical file. A single operating system file intended for presentation. The format of the file, and its delineation, is defined by the operating system.

physical medium. A physical entity on which information is presented. Examples of a physical medium are a sheet of paper, a roll of paper, an envelope, and a display screen. See also medium presentation space and sheet.

physical printable area. A bounded area defined on a side of a sheet within which printing can take place. The physical printable area is an attribute of sheet size and printer capabilities, and cannot be altered by the host. The physical printable area is mapped to the medium presentation space, and is used in user printable area and valid printable area calculations. Contrast with user printable area and valid printable area.

picket fence bar code. A bar code pattern presenting the axis of the symbol in its length dimension parallel to the X_{bc} axis of the bar code presentation space. Synonymous with horizontal bar code.

picture chain. A string of segments that defines a picture. Synonymous with segment chain.

picture element. The smallest printable or displayable unit on a physical medium. In computer graphics, the smallest element of a physical medium that can be independently assigned color and intensity. Picture elements per inch is often used as a measurement of presentation granularity. Synonymous with pel and pixel.

pixel. The smallest printable or displayable unit on a physical medium. In computer graphics, the smallest element of a physical medium that can be independently assigned color and intensity. Picture elements per inch is often used as a measurement of presentation granularity. Synonymous with pel and picture element.

PMC-PFO. See PMC preprinted form overlay (PMC-PFO).

PMC preprinted form overlay (PMC-PFO). In MO:DCA, a PFO that is designed to simulate a preprinted form for a page. A PMC-PFO is invoked with the PMC structured field and is applied last to the page presentation space after all other data for the page has been applied.

PNG. See Portable Network Graphics.

point. (1) A unit of measure used mainly for measuring typographical material. There are seventy-two points to an inch. (2) In GOCA, a parameter that specifies the position within the drawing order coordinate space. See also drawing order coordinate space.

point-operation halftone. Any halftone algorithm that produces output for a given location based only on the single input pixel at that location, independent of its neighbors. Thus, it is accomplished by a simple point-wise comparison of the input image against a predetermined threshold array or mask. Contrast with neighborhood-operation halftone.

polyline. A sequence of connected lines.

Portable Network Graphics (PNG). A lossless image format.

portrait. A presentation orientation in which the X_m axis is parallel to the short sides of a rectangular physical medium. Contrast with landscape.

position. A position in a presentation space or on a physical medium that can be identified by a coordinate from the coordinate system of the presentation space or physical medium. See also picture element. Synonymous with addressable position.

Positive Acknowledge Reply (ACK). In the IPDS architecture, a reply to an IPDS command that has its acknowledgment-required flag on and in which no exception is reported. Contrast with Negative Acknowledge Reply.

PostScript. A page description programming language created by Adobe Systems Inc. that is a presentation-system-independent industry standard for outputting documents and graphics. It describes pages to any output device with a PostScript interpreter.

posture. Inclination of a letter with respect to a vertical axis. Examples of inclination are upright and inclined. An example of upright is Roman. An example of inclined is italics.

pragmatics. Information related to the usage of a construct. See also semantics and syntax.

preprinted form. A form or sheet that is not blank when it is selected as input media for presentation.

preprinted form overlay (PFO). An overlay and associated processing designed to simulate a preprinted form.

presentation data stream. A presentation data stream that is processed in AFP environments. The MO:DCA architecture describes the AFP interchange data stream. The IPDS architecture describes the AFP printer data stream.

presentation device. A device that produces character shapes, graphics pictures, images, or bar code symbols on a physical medium. Examples of a physical medium are a display screen and a sheet of paper.

presentation object. An object that describes presentation data such as text, image, and graphics, in a paginated, final-form format suitable for presentation on a page. Contrast with non-presentation object.

presentation position. An addressable position that is coincident with a character reference point. See also addressable position and character reference point.

presentation process. Synonymous with presentation system.

presentation services. In printing, a software component that communicates with a printer using a printer data stream, such as the IPDS data stream, to print pages, download and manage print resources, and handle exceptions.

presentation space. A conceptual address space with a specified coordinate system and a set of addressable positions. The coordinate system and addressable positions can coincide with those of a physical medium. Examples of presentation spaces are medium, logical page, and object area. See also graphics presentation space, image presentation space, logical page, medium presentation space, and text presentation space.

presentation space orientation. The number of degrees and minutes a presentation space is rotated in a specified coordinate system. For example, the orientation of printing on a physical medium, relative to the X_m axis of the X_m,Y_m coordinate system. See also orientation and text orientation.

presentation system. A system for presenting data. In AFP environments such a system normally contains at least a formatting application, a print server, and a printer. Synonymous with presentation process.

presentation text object. An object that contains presentation text data. See also object.

Presentation Text Object Content Architecture (PTOCA). An architected collection of constructs used to interchange and present presentation text data.

print contrast. A measurement of the ratio of the reflectivities between the bars and spaces of a bar code symbol, commonly expressed as a percent. Synonymous with Print Contrast Signal.

Print Contrast Signal (PCS). A measurement of the ratio of the reflectivities between the bars and spaces of a bar code symbol, commonly expressed as a percent. Synonymous with print contrast.

print control object. A resource object that contains layout, finishing, and resource mapping information used to present a document on physical media. Examples of print control objects are Form Maps and Medium Maps.

print direction. In FOCA, the direction in which successive characters appear in a line of text.

print file. A file that is created for the purpose of printing data. The print file is the highest level of the MO:DCA data-stream document-component hierarchy.

printing baseline. A conceptual line with respect to which successive characters are aligned. See also character baseline. Synonymous with baseline and sequential baseline.

print quality. In bar codes, the measure of compliance of a bar code symbol to the requirements of dimensional tolerance, edge roughness, spots, voids, reflectance, PCS, and quiet zones defined within a bar code symbology.

print unit. In the IPDS architecture, a group of pages bounded by XOH-DGB commands and subject to the group operation *keep group together as a print unit*. A print unit is commonly referred to as a print job.

process color. A color that is specified as a combination of the components, or primaries, of a color space. A process color is rendered by mixing the specified amounts of the primaries. An example of a process color is C=0.1, M=0.8, Y=0.2, K=0.1 in the cyan/magenta/yellow/black (CMYK) color space. Contrast with spot color.

process element. In MO:DCA, a document component that is defined by a structured field and that facilitates a form of document processing that does not affect the presentation of the document. Examples of process elements are Tag Logical Elements (TLEs) that specify document attributes and Link Logical Elements (LLEs) that specify linkages between document components.

Profile Connection Space (PCS). The reference color space defined by ICC, in which colors are encoded in order to provide an interface for connecting source and destination transforms. The PCS is based on the CIE 1931 standard colorimetric observer.

prolog. The first portion of a segment's data. Prologs are optional. They contain attribute settings and drawing controls. Synonymous with segment prolog.

propagation. A method used to retain a segment's properties through other segments that it calls.

proper subset. A set whose members are also members of a larger set.

proportion. Relationship of the width of a letter to its height.

proportionally spaced font. A font with graphic characters that have varying character increments. Proportional spacing can be used to provide the appearance of even spacing between presented characters and to eliminate excess blank space around narrow characters. An example of a narrow character is the letter i. Synonymous with typographic font. Contrast with monospaced font and uniformly spaced font.

proportional spacing. The spacing of characters in a printed line so that each character is allotted a space based on the character's width.

Proportional Spacing Machine font (PSM font). A font originating with the electric typewriter and having character increment values that are integer multiples of the narrowest character width.

PSM font. See Proportional Spacing Machine font.

PTOCA. See Presentation Text Object Content Architecture.

Q

quantization. The process of reducing an image with many colors to one with fewer colors, usually in preparation for its conversion to a palette-based image. As a result, most parts of the image (that is, most of its pixels) are given slightly different colors that amounts to a certain level of error at each location. Since photographic images usually have extended regions of similar colors that get converted to the same quantized color, a quantized image tends to have a flat or banded (contoured) appearance unless it is also dithered.

quiet zone. A clear space that contains no machine-readable marks preceding the start character of a bar code symbol or following the stop character. Synonymous with clear area. Contrast with intercharacter gap and space.

R

radial gradient. In GOCA, a gradient where the color change takes place between two full arcs. Contrast with linear gradient.

range. A table heading for architecture syntax. The entries under this heading give numeric ranges applicable to a construct. The ranges can be expressed in binary, decimal, or hexadecimal. The range can consist of a single value.

raster. (1) The area of the video display that is covered by sweeping the electron beam of the display horizontally and vertically. Normally the electronics of the display sweep each line horizontally from top to bottom and return to the top during the vertical retrace interval. (2) In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space. (3) In nonimpact printers, an on-or-off pattern of electrostatic images produced by the laser print head under control of the character generator.

raster direction. An attribute that controls the direction in which a character string grows relative to the inline direction. Values are: left-to-right, right-to-left, top-to-bottom, and bottom-to-top.

rasterize. To convert presentation data into raster (bitmap) form for display or printing.

raster pattern. A rectangular array of pels arranged in rows called scan lines.

readability. The characteristics of visual material that determine the degree of comfort with which it can be read over a sustained period of time. Examples of characteristics that influence readability are type quality, spacing, and composition.

reader. In bar code systems, the scanner or combination of scanner and decoder. See also decoder and scanner.

read rate. In bar codes, the ratio of the number of successful reads on the first attempt to the total number of attempts made to obtain a successful read. Synonymous with first read rate.

rearranged file. A file containing the mapping of code points to the character index values used in a CID file and to the character names used in one or more PFB files. This is a special case of the CMAP file that permits linking of multiple font files and formats together. The code points conform to a particular character coding system that is used to identify the characters in a document data stream. The mapping information in this file is in an ASCII file format defined by Adobe Systems Inc.

record-format line data. A form of line data where each record is preceded by a 10-byte identifier. The record is presented by matching its ID to the ID specified on a Record Descriptor in the Data Map of a Page Definition.

recording algorithm. An algorithm that determines the relationship between the physical location and logical location of image points in image data.

recovery-unit group. (1) In the IPDS architecture, a group of pages identified by the XOH Define Group Boundary command and controlled by the Keep-Group-Together-as-a-Recovery-Unit group operation specified by the XOH Specify Group Operation command. The recovery-unit group also includes all copies specified by the Load Copy Control command. (2) In the MO:DCA architecture, a group of pages identified as a unit for error recovery purposes, such as in cases of a printer recovery from an error that occurs in the middle of the group. A recovery-unit group is identified by a Begin Named Group (BNG) and End Named Group (ENG) pair that contains a Keep Group Together (X'9D') triplet.

redaction. The process of applying an opaque mask over a page so that a selected portion of the page is visible. Since this function is typically used to prevent unauthorized viewing of data, an associated security level is also provided.

reflectance. In bar codes, the ratio of the amount of light of a specified wavelength or series of wavelengths reflected from a test surface to the amount of light reflected from a barium oxide or magnesium oxide standard under similar illumination conditions.

relative coordinate. One of the coordinates that identify the location of an addressable point by means of a displacement from some other addressable point. Contrast with absolute coordinate.

relative line. A straight line developed from a specified point by a given displacement.

relative metrics. Graphic character measurements expressed as fractions of a square, called the Em square, whose sides correspond to the vertical size of the font. Because the measurements are relative to the size of the Em square, the same metrics can be used for different point sizes and different raster pattern resolutions. Relative metrics require defining the unit of measure for the Em square, the point size of the font, and, if applicable, the resolution of the raster pattern.

relative move. A method used to establish a new current position. Distance and direction from the current position are used to establish the new current position. The direction of displacement is inline along the I axis in the I direction, or baseline along the B axis in the B direction, or both.

relative positioning. The establishment of a position within a coordinate system as an offset from the current position. Contrast with absolute positioning.

rendering intent. A particular gamut-mapping style or method of converting colors in one gamut to colors in another gamut. ICC profiles support four different rendering intents: perceptual, media-relative colorimetric, saturation, and ICC-absolute colorimetric.

repeating group. A group of parameter specifications that can be repeated.

repeat string. A method used to repeat the character content of text data until a given number of characters has been processed. Any control sequences in the text data are ignored. This method provides the functional equivalence of a Transparent Data control sequence when the given number of repeated characters is equal to the number of characters in the text data.

reserved. Having no assigned meaning and put aside for future use. The content of reserved fields is not used by receivers, and should be set by generators to a specified value, if given, or to binary zeros. A reserved field or value can be assigned a meaning by an architecture at any time.

reset color. The color of a presentation space before any data is added to it. Synonymous with color of medium.

resident resource. In the IPDS architecture, a resource in a printer or in a resource-caching intermediate device. A resident resource can be installed manually or can be captured by the device if it is intended for public use. A resident resource is referenced by a global ID that is valid for the duration of the resource's presence in the device. Contrast with downloaded resource.

resolution. (1) A measure of the sharpness of an input or output device capability, as given by some measure relative to the distance between two points or lines that can just be distinguished. (2) The number of addressable pels per unit of length.

resolution correction. A method used to present an image on a printer without changing the physical size or proportions of the image when the resolutions of the printer and the image are different.

resolution-correction ratio. The ratio of a device resolution to an image presentation space resolution.

resolution modification. A method used to write an image on an image presentation space without changing the physical size of the image when the resolutions of the presentation space and the image are different.

resource. An object that is referenced by a data stream or by another object to provide data or information. Resource objects can be stored in libraries. In MO:DCA, resource objects can be contained within a resource group. Examples of resources are fonts, overlays, and page segments. See also downloaded resource, resident resource, and secondary resource.

resource caching. In the IPDS architecture, a function in a printer or intermediate device whereby downloaded resources are captured and made resident in the printer or intermediate device.

retired. Set aside for a particular purpose, and not available for any other purpose. Retired fields and values are specified for compatibility with existing products and identify one of the following:

- Fields or values that have been used by a product in a manner not compliant with the architected definition
- Fields or values that have been removed from an architecture

reuse LND. A Line Descriptor (LND) in a chain of LNDs, also called a reuse chain, where all LNDs process fields in the same line-data record. See also base LND.

RGB. Red, green and blue, the additive primary colors.

RGB color space. The basic additive color model used for color video display, as on a computer monitor.

RIP. A raster image processor (RIP) is a hardware or software tool that processes a presentation data stream and converts it—rasterizes it—to a printable format.

RM4SCC. See Royal Mail 4 State Customer Code.

Roman. Relating to a type style with upright letters.

root segment. A segment in the picture chain that is not called by any other segment. If a single segment that is not in a segment chain is drawn, it is treated as a root segment for the duration of the drawing process.

rotating. In computer graphics, turning all or part of a picture about an axis perpendicular to the presentation space.

rotation. The orientation of a presentation space with respect to the coordinate system of a containing presentation space. Rotation is measured in degrees in a clockwise direction. Zero-degree rotation exists when the angle between a presentation space's positive X axis and the containing presentation space's positive X axis is zero degrees. Contrast with character rotation.

row. A subarray that consists of all elements that have an identical position within the high dimension of a regular two-dimensional array.

Royal Mail 4 State Customer Code (RM4SCC). A two-dimensional bar code symbology developed by the United Kingdom's Royal Mail postal service for use in automated mail-sorting processes.

rule. A solid line of any line width.

S

sans serif. A type style characterized by strokes that end with no flaring or crossing of lines at the stroke ends. Contrast with serif.

saturation rendering intent. The exact gamut mapping of the saturation rendering intent is vendor specific and involves compromises such as trading off preservation of hue in order to preserve the vividness of pure colors. It is useful for images that contain objects such as charts or diagrams.

SBCS. See single-byte character set.

SBIN. A data type for architecture syntax, that indicates that one or more bytes be interpreted as a signed binary number, with the sign bit in the high-order position of the leftmost byte. Positive numbers are represented in true binary notation with the sign bit set to B'0'. Negative numbers are represented in twos-complement binary notation with a B'1' in the sign-bit position.

Scalable Vector Graphics (SVG). An XML-based vector image format.

scaling. Making all or part of a picture smaller or larger by multiplying the coordinate values of the picture by a constant amount. If the same multiplier is applied along both dimensions, the scaling is uniform, and the proportions of the picture are unaffected. Otherwise, the scaling is anamorphic, and the proportions of the picture are changed. See also anamorphic scaling.

scaling ratio. (1) The ratio of an image-object-area size to its image-presentation-space size. (2) In FOCA, the ratio of horizontal to vertical scaling of the graphic characters. See also horizontal scale factor.

scan line. A series of picture elements. Scan lines in raster patterns form images. See also picture element and raster pattern.

scanner. In bar codes, an electronic device that converts optical information into electrical signals. See also reader.

screen. (1) A halftone-threshold array. (2) The display surface of a display device such as a computer monitor.

scrolling. A method used to move a displayed image vertically or horizontally so that new data appears at one edge as old data disappears at the opposite edge. Data disappears at the edge toward which an image is moved and appears at the edge away from which the data is moved.

SDA. See special data area.

secondary resource. A resource for an object that is itself a resource.

section. A portion of a double-byte code page that consists of 256 consecutive entries. The first byte of a two-byte code point is the section identifier. A code-page section is also called a code-page ward in some environments. See also code page and code point.

section identifier. A value that identifies a section. Synonymous with section number.

section number. A value that identifies a section. Synonymous with section identifier.

secure overlay. An overlay that can be printed anywhere within the physical printable area. A secure overlay is not affected by an IPDS Define User Area command.

segment. (1) In GOCA, a set of graphics drawing orders contained within a Begin Segment command. See also graphics segment. (2) In IOCA, image content bracketed by Begin Segment and End Segment self-defining fields. See also image segment.

segment chain. A string of segments that defines a picture. Synonymous with picture chain.

segment exception condition. An architecture-provided classification of the errors that can occur in a segment. Segment exception conditions are raised when a segment error is detected. Examples of segment errors are segment format, parameter content, and sequence errors.

segment offset. A position within a segment, measured in bytes from the beginning of the segment. The beginning of a segment is always at offset zero.

segment prolog. The first portion of a segment's data. Prologs are optional. They contain attribute settings and drawing controls. Synonymous with prolog.

segment properties. The segment characteristics used by a drawing process. Examples of segment properties are segment name, segment length, chained, dynamic, highlighted, propagated, and visible.

segment transform. A model transform that is applied to a whole segment.

self checking. In bar codes, using a checking algorithm that can be applied to each character independently to guard against undetected errors.

semantics. The meaning of the parameters of a construct. See also pragmatics and syntax.

sequential baseline. A conceptual line with respect to which successive characters are aligned. See also character baseline. Synonymous with baseline and printing baseline.

sequential baseline position. The current addressable position for a baseline in a presentation space or on a physical medium. See also baseline coordinate and current baseline presentation coordinate.

serif. A short line angling from or crossing the free end of a stroke. Examples are horizontal lines at the tops and bottoms of vertical strokes on capital letters, for example, *I* and *H*, and the decorative strokes at the ends of the horizontal members of a capital E. Contrast with sans serif.

server. In a network, hardware or software that provides facilities to other stations. Examples include: a file server, a printer server, and a mail server.

session. In the IPDS architecture, the period of time during which a presentation services program has a two-way communication with an IPDS device. The session consists of a physical attachment and a communications protocol; the communications protocol carries an IPDS dialog by transparently transmitting IPDS commands and Acknowledge Replies. See also IPDS dialog.

setup file. In the IPDS architecture, an object container that provides setup information for a printer. Setup files are downloaded in home state and take effect immediately. Setup files are not managed as resources.

shade. Variation of a color produced by mixing it with black.

shape compression. A method used to compress digitally encoded character shapes using a specified algorithm.

shape technology. A method used to encode character shapes digitally using a specified algorithm.

shear. The angle of slant of a character cell that is not perpendicular to a baseline. Synonymous with character shear.

shearline direction. In GOCA, the direction specified by the character shear and character angle attributes.

sheet. A division of the physical medium; multiple sheets can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a sheet. Envelopes are an example of a physical medium that comprises only one sheet. The IPDS architecture defines four types of sheets: cut-sheet media, continuous-form media, envelopes, and computer output on microfilm. Each type of sheet has a top edge. A sheet has two sides, a front side and a back side. Synonymous with form.

show through. In bar codes, the generally undesirable property of a substrate that permits underlying markings to be seen.

side. A physical surface of a sheet. A sheet has a front side and a back side. See also sheet.

signed integers. The positive natural numbers (1, 2, 3, ...), their negatives (-1, -2, -3, ...) and the number zero. The set of all integers is usually denoted in mathematics by Z, which stands for *Zahlen* (German for "numbers").

simplex printing. A method used to print data on one side of a sheet; the other side is left blank. Contrast with duplex printing.

single-byte character set (SBCS). A character set that can contain up to 256 characters.

single-byte coded font. A coded font in which the code points are one byte long.

slope. The posture, or incline, of the main strokes in the graphic characters of a font. Slope is specified in degrees by a font designer.

soft object. An object that is not mapped in an environment group and is therefore not sent to the presentation device until it is referenced within a page or overlay. Contrast with hard object.

space. In bar codes, the lighter element of a printed bar code symbol, usually formed by the background between bars. See also element. Contrast with bar, clear area, intercharacter gap, and quiet zone.

space width. In bar codes, the thickness of a bar code symbol space measured from the edge closest to the symbol start character to the trailing edge of the same space.

spanning. In the IPDS architecture, a method in which one command is used to start a sequence of constructs. Subsequent commands continue and terminate that sequence. See also control sequence chaining.

special data area (SDA). The data area in an IPDS Acknowledge Reply that contains data requested by the host or generated by a printer as a result of an exception.

Specifications for Web Offset Publications (SWOP). A standard set of specifications for color separations, proofs, and printing to ensure consistency of color printing.

spot. In bar codes, the undesirable presence of ink or dirt in a bar code symbol space.

spot color. A color that is specified with a unique identifier such as a number. A spot color is normally rendered with a custom colorant instead of with a combination of process color primaries. See also highlight color. Contrast with process color.

sRGB. One of the standard RGB color spaces, a means of specifying precisely how any given RGB value should appear on a display or printed paper or any other output device. sRGB was promoted by the ICC and submitted for standardization by the International Electrotechnical Commission (IEC).

stack. A list that is constructed and maintained so that the next item to be retrieved and removed is the most recently stored item still in the list. This is sometimes called last-in-first-out (LIFO).

standard action. The architecture-defined action to be taken on detecting an exception condition, when the controlling environment specifies that processing should continue.

standard line type value. A predefined line type, like solid, invisible, or dash dot. Contrast with custom line type value.

start-stop character or pattern. In bar codes, a special bar code character that provides the scanner with start and stop reading instructions as well as a scanning direction indicator. The start character is normally at the left end and the stop character at the right end of a horizontally oriented symbol.

stochastic. A method that uses a pseudo-random dot size and/or frequency to create halftone images, but without the visible regularity in the dot patterns found in traditional screening.

store mode. A mode in which segments are stored for later execution. Contrast with immediate mode.

stroke. A straight or curved line used to create the shape of a letter.

structured field. A self-identifying, variable-length, bounded record, that can have a content portion that provides control information, data, or both. See also document element.

structured field introducer. In MO:DCA, the header component of a structured field that provides information that is common for all structured fields. Examples of information that is common for all structured fields are length, function type, and category type. Examples of structured field function types are begin, end, data, and descriptor. Examples of structured field category types are presentation text, image, graphics, and page.

subordinate object. An object that is lower in the document-component hierarchy than a given object. For example, a page is a subordinate object to a page group, and a page group is a subordinate object to a document.

subpage. A part of a logical page on which line data may be placed. A line data record is identified as belonging to a particular subpage with the subpage identifier byte in the Line Descriptor (LND) structured field. Conditional processing can be used with a Page Definition to select a new Data Map and/or Medium Map to take effect before or after the current subpage is printed.

subset. Within the base-and-towers concept, a portion of architecture represented by a particular level in a tower or by a base. See also subsetting tower.

subsetting tower. Within the base-and-towers concept, a tower representing an aspect of function achieved by an architecture. A tower is independent of any other towers. A tower can be subdivided into subsets. A subset contains all the function of any subsets below it in the tower. See also subset.

substrate. In bar codes, the surface on which a bar code symbol is printed.

subtractive primary colors. Cyan, magenta, and yellow colorants used to subtract a portion of the white light that is illuminating an object. Subtractive colors are reflective on paper and printed media. When used together with various degrees of coverage and variation, they have the ability to create billions of other colors. Contrast with additive primary colors.

suppression. A method used to prevent presentation of specified data. Examples of suppression are the processing of text data without placing characters on a physical medium and the electronic equivalent of the “spot carbon,” that prevents selected data from being presented on certain copies of a presentation space or a physical medium.

surrogate pair. A sequence of two Unicode code points that allow for the encoding of as many as 1 million additional characters without any use of escape codes.

surrogates. A way to refer to one or more surrogate pairs.

SVG. See Scalable Vector Graphics.

SWOP. See Specifications for Web Offset Publications.

symbol. (1) A visual representation of something by reason of relationship, association, or convention. (2) In GOCA, the subpicture referenced as a character definition within a font character set and used as a character, marker, or fill pattern. A bitmap can also be referenced as a symbol for use as a fill pattern. See also bar code symbol.

symbol length. In bar codes, the distance between the outside edges of the quiet zones of a bar code symbol.

symbology. A bar code language. Bar code symbologies are defined and controlled by various industry groups and standards organizations. Bar code symbologies are described in public domain bar code specification documents. Synonymous with bar code symbology. See also Canadian Grocery Product Code (CGPC), European Article Numbering (EAN), Japanese Article Numbering (JAN), and Universal Product Code (UPC).

symbol set. A coded font that is usually simpler in structure than a fully described font. Symbol sets are used where typographic quality is not required. Examples of devices that might not provide typographic quality are dot-matrix printers and displays. See also character set, marker set, and pattern set.

synchronous exception. In the IPDS architecture, a data-stream, function no longer achievable, or resource-storage exception that must be reported to the host before a printer can return a Positive Acknowledge Reply or can increment the received-page counter for a page containing the exception. Synchronous exceptions are those with action code X'01', X'06', X'0C', or X'1F'. See also data-stream exception. Contrast with asynchronous exception.

syntax. The rules governing the structure of a construct. See also pragmatics and semantics.

system-level font resource. A common-source font from which:
- Document-processing applications can obtain resolution-independent formatting information.
- Device-service applications can obtain device-specific presentation information.

T

tag. A data structure that is used within the data portion of a color management resource (CMR). A CMR tag consists of TagID, FieldType, Count, and ValueOffset.

Tagged Image File Format (TIFF). A rich and flexible graphics image format.

temporary baseline. The shifted baseline used for subscript and superscript.

temporary baseline coordinate. The B value of the I,B coordinate pair of an addressable position on the temporary baseline.

temporary baseline increment. A positive or negative value that is added to the current baseline presentation coordinate to specify the position of a temporary baseline in a presentation space or on a physical medium. Several increments might have been used to place a temporary baseline at the current baseline presentation coordinate.

text. A graphic representation of information. Text can consist of alphanumeric characters and symbols arranged in paragraphs, tables, columns, and other shapes. An example of text is the data sent in an IPDS Write Text command.

Text command set. In the IPDS architecture, a collection of commands used to present PTOCA text data in a page, page segment, or overlay.

text major. A description for text where the Presentation Text Data Descriptor (PTD) is specified in page controls. In MO:DCA, the PTD is in the Active Environment Group (AEG) for the page; in IPDS, the PTD is specified as initial text-major conditions in the Logical Page Descriptor command.

text object. (1) An object that contains text data. (2) A presentation-system-independent, self-defining representation of a two-dimensional presentation space, called the text object space, that contains presentation text data.

text object space. Synonymous with text presentation space.

text orientation. A description of the appearance of text as a combination of inline direction and baseline direction. See also baseline direction, inline direction, orientation, and presentation space orientation.

text presentation. The transformation of document graphic character content and its associated font information into a visible form. An example of a visible form of text is character shapes on a physical medium.

text presentation space. A two-dimensional conceptual space in which text is generated for presentation on an output medium.

throughscore. A line parallel to the baseline and placed through the character.

TIFF. See Tagged Image File Format.

tint. Variation of a color produced by mixing it with white.

toned. Containing marking agents such as toner or ink. Contrast with untoned.

tone transfer curve. A mathematical representation of the relationship between the input and output of a system, subsystem, or equipment. The function is normally one dimensional consisting of a single channel of input corresponding to a single channel of output. In imaging systems, it is mainly used for contrast adjustments. In printing, the tone transfer curve is also used to modify images to compensate for dot gain.

transform. A modification of one or more characteristics of a picture. Examples of picture characteristics that can be transformed are position, orientation, and size. See also model transform, segment transform, and viewing transform.

transform matrix. A matrix that is applied to a set of coordinates to produce a transform.

translating. In computer graphics, moving all or part of a picture in the presentation space from one location to another without rotating.

transparent data. A method used to indicate that any control sequences occurring in a specified portion of data can be ignored.

trimming. Eliminating those parts of a picture that are outside of a clipping boundary such as a viewing window or presentation space. See also viewing window. Synonymous with clipping.

triplet. A three-part self-defining variable-length parameter consisting of a length byte, an identifier byte, and parameter-value bytes.

triplet identifier. A one-byte type identifier for a triplet.

tristimulus values. Three values that together are used to describe a specific color. These values are the amounts of three reference colors (such as red, green, and blue) that can be mixed to give the same visual sensation as the specific color.

truncation. Planned or unplanned end of a presentation space or data presentation. This can occur when the presentation space extends beyond one or more boundaries of its containing presentation space or when there is more data than can be contained in the presentation space.

tumble-duplex printing. A method used to simulate the effect of physically turning a sheet around the X_m axis.

twip. A unit of measure equal to 1/20 of a point. There are 1440 twips in one inch.

type. A table heading for architecture syntax. The entries under this heading indicate the types of data present in a construct. Examples include: BITS, CHAR, CODE, SBIN, UBIN, UNDF.

typeface. All characters of a single type family or style, weight class, width class, and posture, regardless of size. For example, Helvetica Bold Condensed Italics, in any point size.

type family. All characters of a single design, regardless of attributes such as width, weight, posture, and size. Examples are Courier and Gothic.

type structure. Attributes of characters other than type family or typeface. Examples are solid shape, hollow shape, and overstruck.

type style. The form of characters within the same font, for example, Courier or Gothic.

type weight. A parameter indicating the degree of boldness of a typeface. A character's stroke thickness determines its type weight. Examples are light, medium, and bold. Synonymous with weight class.

type width. A parameter indicating a relative change from the font's normal width-to-height ratio. Examples are normal, condensed, and expanded. Synonymous with width class.

typographic font. A font with graphic characters that have varying character increments. Proportional spacing can be used to provide the appearance of even spacing between presented characters and to eliminate excess blank space around narrow characters. An example of a narrow character is the letter *i*. Synonymous with proportionally spaced font. Contrast with monospaced font and uniformly spaced font.

U

UBIN. A data type for architecture syntax, indicating one or more bytes to be interpreted as an unsigned binary number.

unarchitected. Identifies data that is neither defined nor controlled by an architecture. Contrast with architected.

unbounded character box. A character box that can have blank space on any sides of the character shape.

underpaint. A mixing rule in which the intersection of part of a new presentation space P_{new} with part of an existing presentation space $P_{existing}$ keeps the color attribute of $P_{existing}$. This is also referred to as “transparent” or “leave alone” mixing. See also mixing rule. Contrast with blend and overpaint.

underscore. A method used to create an underline beneath the characters in a specified text field. An example of underscore is the line presented under one or more characters. Also a special graphic character used to implement the underscoring function.

UNDF. A data type for architecture syntax, indicating one or more bytes that are undefined by the architecture.

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 characters having a uniform character increment. The distance between reference points of adjacent graphic characters is constant in the escapement direction. The blank space between the graphic characters can vary. Synonymous with monospaced font. Contrast with proportionally spaced font and typographic font.

Uniform Symbol Specification (USS). A series of bar code symbology specifications published by AIM; currently included are USS-Interleaved 2 of 5, USS-39, USS-93, USS-Codabar, and USS-128.

unit base. A one-byte code that represents the length of the measurement base. For example, X'00' might specify that the measurement base is ten inches.

Universal Product Code (UPC). A standard bar code symbology, commonly used to mark the price of items in stores, that can be read and interpreted by a computer.

untoned. Unmarked portion of a physical medium. Contrast with toned.

UP³I. Universal Printer Pre- and Post-Processing Interface; an industry standard interface designed for use in complex printing systems. A specification for this interface can be obtained at `www.afpconsortium.org`.

UPA. See user printable area.

UPC. See Universal Product Code.

uppercase. Pertaining to capital letters. Examples of capital letters are *A*, *B*, and *C*. Contrast with lowercase.

upstream data. IPDS commands that exist in a logical path from a specific point in a printer back to, but not including, host presentation services.

usable area. An area on a physical medium that can be used to present data. See also viewport.

user printable area (UPA). The portion of the physical printable area to which user-generated data is restricted. See also logical page, physical printable area, and valid printable area.

USS. See Uniform Symbol Specification.

UTC. Coordinated Universal Time, the standard time reference for Earth and the human race. Knowing the UTC time and one's time zone offset from it, makes it possible to calculate the local time; for example, 1:00 PM UTC corresponds to 5:00 AM Pacific Standard Time (on the same day). UTC is almost the same thing as Greenwich Mean Time (GMT), that was originally used as the standard time reference.

V

valid printable area (VPA). The intersection of a logical page with the area of the medium presentation space in which printing is allowed. If the logical page is a secure overlay, the area in which printing is allowed is the physical printable area. If the logical page is not a secure overlay and if a user printable area is defined, the area in which printing is allowed is the intersection of the physical printable area with the user printable area. If a user printable area is not defined, the area in which printing is allowed is the physical printable area. See also logical page, physical printable area, secure overlay, and user printable area.

variable space. A method used to assign a character increment dimension of varying size to space characters. The space characters are used to distribute white space within a text line. The white space is distributed by expanding or contracting the dimension of the variable space character's increment dependent upon the amount of white space to be distributed. See also variable space character and variable space character increment.

variable space character. The code point assigned by the data stream for which the character increment varies according to the semantics and pragmatics of the variable space function. This code point is not presented, but its character increment parameter is used to provide spacing. See also variable space character increment.

variable space character increment. The variable value associated with a variable space character. The variable space character increment is used to calculate the dimension from the current presentation position to a new presentation position when a variable space character is found. See also variable space character.

vector graphics. A vector has a defined starting point, a designated direction, and a specified distance. Vector graphics is line-based graphics data, where vectors determine how straight and curved lines are shaped between specific points. A picture consists of lines and colors to fill the areas enclosed by the lines.

verifier. In bar code systems, a device that measures the bars, spaces, quiet zones, and optical characteristics of a bar code symbol to determine if the symbol meets the requirements of a bar code symbology, specification, or standard.

vertical bar code. A bar code pattern that presents the axis of the symbol in its length dimension parallel to the Y_{bc} axis of the bar code presentation space. Synonymous with ladder bar code.

vertical font size. (1) A characteristic value, perpendicular to the character baseline, that represents the size of all graphic characters in a font. Synonymous with font height. (2) In a font character set, nominal vertical font size is a font-designer defined value corresponding to the nominal distance between adjacent baselines when character rotation is zero degrees and no external leading is used. This distance represents the baseline-to-baseline increment that includes the font's maximum baseline extent and the designer's recommendation for internal leading. The font designer can also define a minimum and a maximum vertical font size to represent the limits of scaling. (3) In font referencing, the specified vertical font size is the desired size of the font when the characters are presented. If this size is different from the nominal vertical font size specified in a font character set, the character shapes and character metrics might need to be scaled prior to presentation.

vertical scale factor. In outline-font referencing, the specified vertical adjustment of the Em square. The vertical scale factor is specified in 1440ths of an inch. When the horizontal and vertical scale factors are different, anamorphic scaling occurs. See also horizontal scale factor.

viewing transform. A transform that is applied to model-space coordinates. Contrast with model transform.

viewing window. That part of a model space that is transformed, clipped, and moved into a graphics presentation space.

viewport. The portion of a usable area that is mapped to the graphics presentation space window. See also graphics model space and graphics presentation space.

visibility. The property of a segment that declares whether the part of a picture defined by the segment is to be displayed or not displayed during the drawing process.

void. In bar codes, the undesirable absence of ink in a bar code symbol bar element.

VPA. See valid printable area.

W

ward. A deprecated term for section.

weight class. A parameter indicating the degree of boldness of a typeface. A character's stroke thickness determines its weight class. Examples are light, medium, and bold. Synonymous with type weight.

white point. One of a number of reference illuminants used in colorimetry that serve to define the color “white”. Depending on the application, different definitions of white are needed to give acceptable results. For example, photographs taken indoors might be lit by incandescent lights, that are relatively orange compared to daylight. Defining “white” as daylight will give unacceptable results when attempting to color correct a photograph taken with incandescent lighting.

white space. The portion of a line that is not occupied by characters when the characters of all the words that can be placed on a line and the spaces between those words are assembled or formatted on a line. When a line is justified, the white space is distributed among the words, characters, or both on the line in some specified manner. See also controlled white space.

width class. A parameter indicating a relative change from the font's normal width-to-height ratio. Examples are normal, condensed, and expanded. Synonymous with type width.

window. A predefined part of a graphics presentation space. See also graphics presentation space window.

writing mode. An identified mode for the setting of text in a writing system, usually corresponding to a nominal escapement direction of the graphic characters in that mode; for example, left-to-right, right-to-left, top-to-bottom.

X

X_{bc} extent. The size of a bar code presentation space in the X_{bc} dimension. See also bar code presentation space.

X_{bc},Y_{bc} coordinate system. The bar code presentation space coordinate system.

X dimension. In bar codes, the nominal dimension of the narrow bars and spaces in a bar code symbol.

X_g,Y_g coordinate system. In the IPDS architecture, the graphics presentation space coordinate system.

X height. The nominal height above the baseline, ignoring the ascender, of the lowercase characters in a font. X height is usually the height of the lowercase letter *x*. See also lowercase and ascender.

X_{io},Y_{io} coordinate system. The IO-Image presentation space coordinate system.

XML. See Extensible Markup Language.

XMP. See Extensible Metadata Platform.

X_m,Y_m coordinate system. (1) In the IPDS architecture, the medium presentation space coordinate system. (2) In MO:DCA, the medium coordinate system.

X_{oa},Y_{oa} coordinate system. The object area coordinate system.

X_{ol},Y_{ol} coordinate system. The overlay coordinate system.

X_p extent. The size of a presentation space or logical page in the X_p dimension. See also presentation space and logical page.

X_{pg},Y_{pg} coordinate system. The coordinate system of a page presentation space. This coordinate system describes the size, position, and orientation of a page presentation space. Orientation of an X_{pg},Y_{pg} coordinate system is relative to an environment specified coordinate system, for example, an X_m,Y_m coordinate system.

X_p,Y_p coordinate system. The coordinate system of a presentation space or a logical page. This coordinate system describes the size, position, and orientation of a presentation space or a logical page. Orientation of an X_p, Y_p coordinate system is relative to an environment-specified coordinate system. An example of an environment-specified coordinate system is the X_m,Y_m coordinate system. The X_p,Y_p coordinate system origin is specified by an IPDS Logical Page Position command. See also logical page, medium presentation space, and presentation space.

Y

Y_{bc} extent. The size of a bar code presentation space in the Y_{bc} dimension. See also bar code presentation space.

YCbCr. A three-component color space that approximately models how color is interpreted by the human visual system, with an intensity value and two color values. YCbCr and YCrCb use the same three values, but in a different order.

YCCK. CMYK data carried in the luminance-chrominance form. YCC are computed from CMY, while K is the black channel carried in the reverse-video form (K = 255 - K). See Appendix B, “Adobe APP14 JPEG Marker” in *Presentation Object Subsets for AFP*.

YCrCb. A three-component color space that approximately models how color is interpreted by the human visual system, with an intensity value and two color values. YCbCr and YCrCb use the same three values, but in a different order.

Y_p extent. The size of a presentation space or logical page in the Y_p dimension. See also presentation space and logical page.

Yxy color space. A color space belonging to the XYZ base family that expresses the XYZ values in terms of x and y chromaticity coordinates, somewhat analogous to the hue and saturation coordinates of the HSV color space.

Index

A

B

C

D

E

F

G

I

J

L

M

N

O

P

R

S

T

U

W

Y

Advanced Function Presentation Consortium

Image Object Content Architecture Reference

AFPC-0003-08

AFPConsortium™

Advanced Function Presentation