Advanced Function Presentation Consortium
Data Stream and Object Architectures

AFP Programming Guide and Line Data Reference

AFPC-0010-05

AFPConsortium
Advanced Function Presentation

Note:

Before using this information, read the information in “Notices” on page 191.

AFPC-0010-05
Sixth Edition (February 2018)

This edition applies to the AFP™ Line Data architecture. It is the first edition produced by the AFP Consortium™ (AFPC™) and replaces and makes obsolete the previous edition (S544-3884-04) that was published by IBM®. This edition remains current until a new edition is published.

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

Internet

Visit our home page: `www.afpcinc.org`

Copyright © AFP Consortium 1994, 2018

Preface

This book is a reference for printing line-mode and mixed-mode data in an AFP environment. It describes the presentation of line-mode data streams using the Page Definition (PageDef) print control object. Line-mode data streams that adhere to the specifications in this document are accepted for printing by presentation services programs in most system environments.

This book also defines structured fields and objects that are used exclusively for processing line data. Because many of these objects are either defined directly by the Mixed Object Document Content Architecture™ (MO:DCA™) or based on MO:DCA definitions, readers should also familiarize themselves with the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Note: The AFP Line Data architecture has been stabilized such that it can be fully used within AFP products and environments, but will not be extended. Many AFP products support both line data and Mixed Object Document Content Architecture (MO:DCA) documents.

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

This book is intended for programmers who write applications that generate line-mode and mixed-mode data streams or data stream objects for printing across AFP system environments.

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.

Related Publications

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

Table 1. AFP Consortium Architecture References

AFP Architecture Publication	Order Number
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

Table 3. AFP Font-Related Documentation

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

Table 4. UP³I™ Architecture Documentation

UP³I Publication	Order Number
Universal Printer Pre- and Post-Processing Interface (UP³I) Specification	Available at `www.afpcinc.org`

Summary of Changes

Changes between this edition and the previous edition are marked by a vertical bar “|” in the left margin.

This sixth edition of the *AFP Programming Guide and Line Data Reference* contains the following changes:

- Ability added to create BCOCA™ bar code data from multiple FIELD commands using the new Concatenate Bar Code Data (X'93') triplet
- Ability added to reuse the Concatenate Bar Code Data (X'93') triplet within RCDs and XMDs
- Ability added to specify a desired bar code symbol width
- AFP Consortium information added
- Glossary terms added and improved; previous versions of this book provided terms used by or related to line data, but this version includes a much more complete set of terms related to the entire AFP architecture
- IBM-specific product information removed
- Product information updated to be more inclusive
- Style changes made to match other AFPC books
- Support for ICC DeviceLink Color Management Resources (CMRs)
- Support for PTOCA text objects (with an Object Environment Group) as an OCA object that can be included in line data in a mixed-mode document or with an IOB

Note: The AFP Line Data architecture has been stabilized such that it can be fully used within AFP products and environments, but will not be extended. Many AFP products support both line data and Mixed Object Document Content Architecture (MO:DCA) documents.

Contents

Copyright © AFP Consortium 1994, 2018

Figures

Copyright © AFP Consortium 1994, 2018

Tables

Copyright © AFP Consortium 1994, 2018

Chapter 1. Introduction

Programmers can develop applications for Advanced Function Presentation (AFP) hardware and software, generating either traditional unformatted line data, fully composed Mixed Object Document Content Architecture (MO:DCA) data (also called AFP data), or a combination of both. This book contains examples and suggestions for writing such applications. AFP line data and MO:DCA data streams are supported, for example, in the following environments:

- Advanced Interactive Executive (AIX®)
- Application System/400 (AS/400), iSeries, and System i5®
- Operating System/2® (OS/2)
- IBM mainframe environments, including:
  - OS/390® and z/OS®
  - Virtual Machine (VM) and z/VM®
  - Virtual Storage Extended (VSE) and z/VSE®
- Linux™
- Microsoft® Windows®

The print data streams can include text, images, graphics, and bar code in MO:DCA format. The MO:DCA architecture defines the data stream used by applications to describe documents and object envelopes for interchange with other applications and application services. Documents defined in the MO:DCA format can be archived in a data base, then later retrieved, viewed, and printed in local or distributed systems environments.

Presentation services programs in the zSeries environments accept data in traditional line-printer format and generate page-mode output from the line data, using information contained in a Page Definition (PageDef) resource object. The line data mapped by the Page Definition might or might not include additional MO:DCA structured fields. A file that includes a combination of line data and MO:DCA structured fields is called a *mixed-mode* file. Only certain MO:DCA structured fields can be intermixed with line data. Detailed information on coding those structured fields appears in Chapter 4, “Mixed Documents: Adding MO:DCA Structured Fields to Line Data”, on page 39.

Presentation services programs in the AIX and Windows environments accept non-MO:DCA data streams that can be formatted using Page Definition resource objects. These data streams can be in any of the following formats:

- Traditional line printer format, also called 1403 format
- Unformatted ASCII files without escape sequences
- DBCS (double-byte character set) ASCII files generated for an IBM 5577 or 5587

Presentation services programs in the IBM i environment accept line data or mixed data, either created on a zSeries platform and networked to an IBM i environment or created natively on an IBM i environment. Such data is placed on the printer spool using a *Printer File*, which may also specify the Page Definition and Form Definition (Formdef) to be used for formatting and printing the data.

Copyright © AFP Consortium 1994, 2018

Related Architectures

Mixed-mode data streams can include line data, MO:DCA structured fields, and objects of the following types:

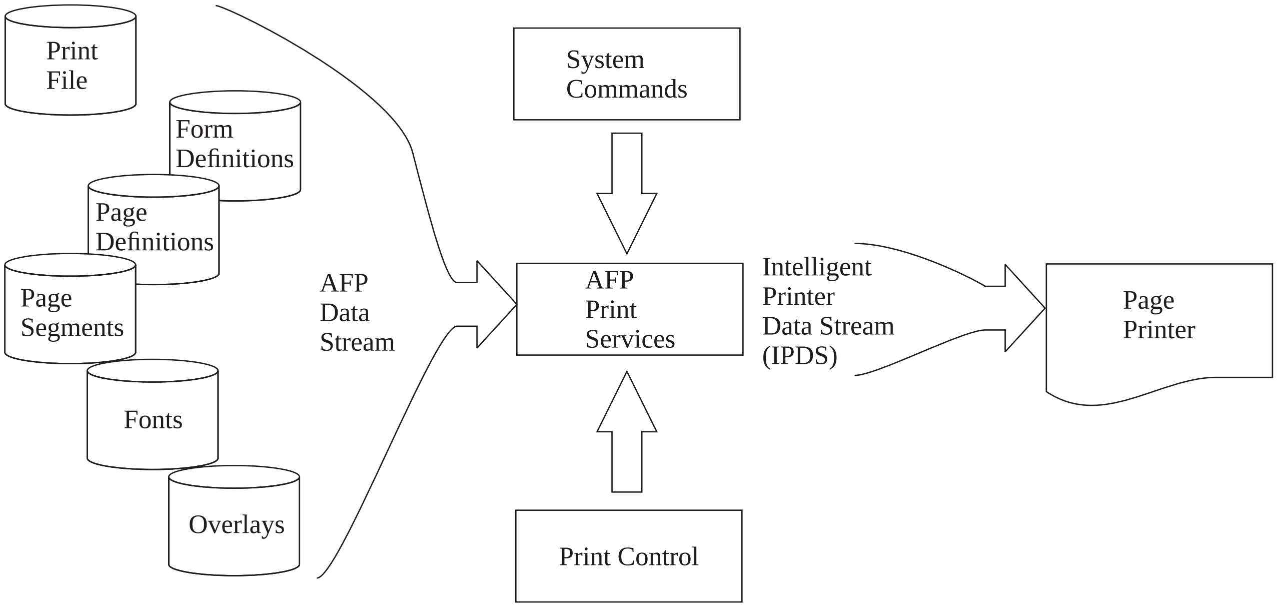
- Bar Code Object Content Architecture™ (BCOCA)
- Color Management Object Content Architecture (CMOCA)
- Font Object Content Architecture (FOCA)
- Graphics Object Content Architecture (GOCA)
- Image Object Content Architecture (IOCA)
- Presentation Text Object Content Architecture (PTOCA)
- Non-OCA paginated presentation objects such as Encapsulated PostScript® (EPS)

A related architecture, but not a user programming language, is the Intelligent Printer Data Stream™ (IPDS™) architecture. This is the data stream architecture used by print server products to manage IPDS printers.

System Model

AFP print servers provide support for interpreting line data, mixed-mode data, and MO:DCA data, for resolving resource references, and for building printer data streams for driving IPDS printers. Figure 1 shows the general relationship between the AFP data streams, the print server products, and IPDS printers.

Figure 1. AFP System Printing Relationships



Each print server product has its own books that describe how to submit print jobs in its system environment. See these books for information on setting up jobs for printing.

Supported Environments

Presentation services programs provide common printer support and print services in the following environments:

- AIX
- AS/400 and System i5
- OS/2
- OS/390 and z/OS
- VM and z/VM
- VSE and z/VSE
- Linux
- Windows

Chapter 2. Line Data and MO:DCA (AFP) Data

The Advanced Function Presentation (AFP) products have been developed to be consistent with a set of architectures that define the format of documents and the nature of the commands sent from the host software to the supported printers. The Mixed Object Document Content Architecture (MO:DCA) defines a device-independent data stream format for interchanging documents among AFP products. Data to be printed can include text, graphics, images, and bar codes.

The *objects* used for Advanced Function Presentation include:

- Font objects, which consist of *font character sets* containing the patterns for letters, numbers, and special characters, and *code pages* that associate a hexadecimal value with each character in the font character set
- Resource objects, including overlays and page segments, which in turn can include text, graphics, image, and bar code
- Print control objects, which include Page Definitions used to format line data and Form Definitions used to control physical aspects of the print environment.

These objects can exist in resource libraries external to the data to be printed or can be included inline with the data that will use them.

AFP objects can be obtained in a number of different ways. For example, a wide variety of fonts are available from IBM and other sources, and Page Definitions, Form Definitions, and overlays can be built using any of several tools available from IBM and other AFP vendors.

MO:DCA documents can be generated using a variety of AFP products. In line data environments, users can add MO:DCA structured fields directly to their line data. Chapter 4, “Mixed Documents: Adding MO:DCA Structured Fields to Line Data” provides information on how to do this.

Presentation services programs, available on IBM mainframe systems, such as z/OS, z/VM, and z/VSE, IBM i (previously known as OS/400®), IBM OS/2, and on AIX, Linux, and Windows systems, accept MO:DCA documents and resources and in turn generate Intelligent Printer Data Stream (IPDS) commands to drive the printers. Presentation services programs can also accept other forms of data as input. One of the most widely used of these is called *line data*.

Line Data

Line data, meaning application output to be printed that is not already in MO:DCA format, is supported by presentation services programs and formatted by Page Definition resource objects in all system environments except IBM OS/2. The nature of line data is slightly different in the system environments where it appears.

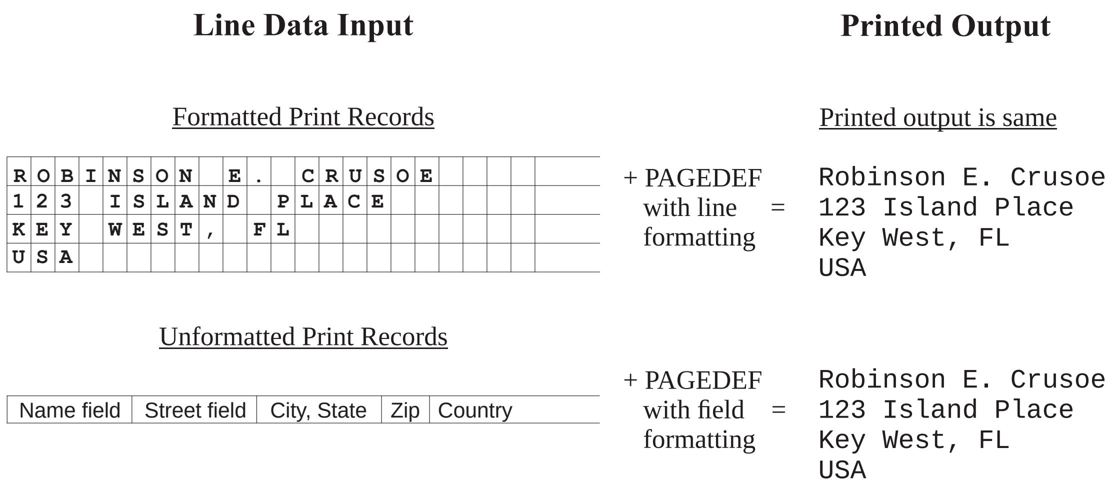
IBM Mainframe Environments

IBM mainframe applications written in programming languages such as Assembler, COBOL, FORTRAN, PL/I, RPG, or others have historically produced output files to be printed on line-mode printers such as the IBM 1403, 3211, or 3800-1. These line data files consist of individual print records, each of which corresponds to one line of data on an impact printer. The application program either formats line data records or leaves them unformatted.

Formatted line data records contain information exactly as it will be printed, because line printers have little or no capability of formatting print output records. Unformatted line data records contain only the fields of data to be printed. With unformatted line data records, the data is not formatted into lines, columns, paragraphs or other structures that determine how the records will appear on paper. AFP print server products support printing of both formatted and unformatted line data records using the Page Definition (also called PageDef) resource object.

Figure 2 illustrates the difference between formatted and unformatted line data records.

Figure 2. Formatted and Unformatted Line Data Records



Traditional impact printers allowed only one variation on the line-by-line format of output. The first character of the line, preceding the actual data characters, could optionally be a *carriage control byte*. This byte indicated whether the data line should be printed using single, double, or triple spacing, whether spacing should be suppressed (creating an overstrike), or whether the line should be placed at a predefined position on the page that was associated with a value of 1 through 12, known as the *channel code*. This page position value was defined in an auxiliary object called a forms control buffer (FCB). The FCB defined the number of lines per page, whether lines were spaced at 6 or 8 per inch, and which line, if any, was associated with the position values of 1 through 12. The IBM 3800 Model 1 added a spacing value of 12 lines per inch to the FCB; and the IBM 3800 Model 3 added 10 lines per inch.

The carriage control character could be represented in either of two coding schemes:

- *American National Standards Institute (ANSI) carriage control* is a standard representation used with printers from many different vendors. Table 5 on page 7 lists the ANSI codes and their functions.
- *Machine code control characters* were defined by IBM. They correspond to the channel command words issued by the operating system to accomplish the desired function. Table 6 on page 7 lists the IBM machine code values and their functions.

ANSI and machine codes may not be intermixed within a single data set.

Line spacing is handled differently by ANSI and machine code carriage controls. ANSI conventions cause spacing to take place first, followed by printing the line, while with machine codes, the line is printed first, and then the spacing action is performed.

Note that if a spacing control action moves the print position past the last line of the current page, processing continues at the first print position of a new page. That is, the spacing action is not carried over to the new page.

Table 5. ANSI Carriage Control Characters

Control Character Value (in hexadecimal)	Function
X'40' (blank)	Space 1 line, then print (single spacing)
X'F0' (zero)	Space 2 lines, then print (double spacing)
X'60' (dash)	Space 3 lines, then print (triple spacing)
X'4E' (plus sign)	Suppress spacing, then print (overstrike previous line)
X'F1'	Print the data at line position defined as Channel 1 (by convention, the first line on a new page)
X'F2'	Print the data at the line position defined as Channel 2 in the FCB
X'F3'	Print the data at the line position defined as Channel 3 in the FCB
X'F4'	Print the data at the line position defined as Channel 4 in the FCB
X'F5'	Print the data at the line position defined as Channel 5 in the FCB
X'F6'	Print the data at the line position defined as Channel 6 in the FCB
X'F7'	Print the data at the line position defined as Channel 7 in the FCB
X'F8'	Print the data at the line position defined as Channel 8 in the FCB
X'F9'	Print the data at the line position defined as Channel 9 in the FCB
X'C1'	Print the data at the line position defined as Channel 10 in the FCB
X'C2'	Print the data at the line position defined as Channel 11 in the FCB
X'C3'	Print the data at the line position defined as Channel 12 in the FCB
Note: When ANSI carriage controls are used, only the values that appear in this table are considered valid by presentation services programs, which treat any other ANSI carriage control value as invalid and print any data on the line using single spacing.	

Table 6. Machine Code Control Characters

Control Character Value (in hexadecimal)	Function
X'03'	No operation
X'09'	Print and space 1 line (single spacing)
X'11'	Print and space 2 lines (double spacing)
X'19'	Print and space 3 lines (triple spacing)
X'01'	Print without spacing (overstrike next line)
X'89'	Print the data, then skip to the line position defined as Channel 1 (by convention, the first line on a new page)
X'91'	Print the data, then skip to the line position defined as Channel 2
X'99'	Print the data, then skip to the line position defined as Channel 3
X'A1'	Print the data, then skip to the line position defined as Channel 4
X'A9'	Print the data, then skip to the line position defined as Channel 5
X'B1'	Print the data, then skip to the line position defined as Channel 6

Table 6 Machine Code Control Characters (cont'd.)

Control Character Value (in hexadecimal)	Function
X'B9'	Print the data, then skip to the line position defined as Channel 7
X'C1'	Print the data, then skip to the line position defined as Channel 8
X'C9'	Print the data, then skip to the line position defined as Channel 9
X'D1'	Print the data, then skip to the line position defined as Channel 10
X'D9'	Print the data, then skip to the line position defined as Channel 11
X'E1'	Print the data, then skip to the line position defined as Channel 12
X'0B'	Space 1 line without printing
X'13'	Space 2 lines without printing
X'1B'	Space 3 lines without printing
X'8B'	Skip to Channel 1 immediate (by convention, the first line on a new page)
X'93'	Skip to the Channel 2 FCB position immediate
X'9B'	Skip to the Channel 3 FCB position immediate
X'A3'	Skip to the Channel 4 FCB position immediate
X'AB'	Skip to the Channel 5 FCB position immediate
X'B3'	Skip to the Channel 6 FCB position immediate
X'BB'	Skip to the Channel 7 FCB position immediate
X'C3'	Skip to the Channel 8 FCB position immediate
X'CB'	Skip to the Channel 9 FCB position immediate
X'D3'	Skip to the Channel 10 FCB position immediate
X'DB'	Skip to the Channel 11 FCB position immediate
X'E3'	Skip to the Channel 12 FCB position immediate

Note: Presentation services programs ignore the following hexadecimal machine-code carriage control characters and do not print lines containing them: X'02' through X'07', X'0A', X'12', X'23', X'43', X'63', X'6B', X'73', X'7B', X'EB', X'F3', and X'FB'. Presentation services programs treat any other carriage control value as invalid and print any data on the line using single spacing.

One other modification to printer line data was introduced with the IBM 3800 Model 1. This modification allows an additional byte to appear at the beginning of a line to indicate which one of up to four different character arrangement tables loaded in the printer is used to print the line. This byte, the *table reference character* (TRC), contains a value of X'F0', X'F1', X'F2', or X'F3', corresponding to the relative position of the desired character arrangement table in the list of table names specified in the data set's job control language. If carriage control bytes are used with the data, the table reference character follows the carriage control byte but precedes the data bytes. If carriage control is not used, the table reference character is the first byte of the data record. As with carriage control, if table reference characters are used, every data record must contain a TRC byte. More information on table reference characters can be found in the application programming guides for IBM Print Services Facility™ (PSF) products.

AIX, Linux, and Windows Environments

Data in an AIX, Linux, or Windows environment can be in *stream* format, with each record or line to be printed delimited by a line separator; or it can be in *record-based* format. In record-based format, the *line separator* is not required, as the length of each record is contained in a two-byte prefix to the record. Either of these formats is considered line data and can be mapped for printing by presentation services programs if a Page Definition is used.

Note: The *line separator* is described in Line Data Summary.

IBM i Environment

In IBM i (previously known as OS/400) print environments, line data and mixed data is written to the system spool using a *Printer File*. This file may also reference the Page Definition and Form Definition to be used for formatting and printing the data.

IBM OS/2 Environment

There is no known support on IBM OS/2 systems for Page Definitions. A presentation services program in the IBM OS/2 environment can accept data in a MO:DCA format or it can print IPDS data streams sent from another AFP system.

Table 7. Platform Support of Data formats

Platform	Record-based	Stream
IBM mainframe	X	X (Note 1)
AIX, Linux, and Windows	X (Note 2)	X
IBM i	X	X (Note 1)

Notes:

1. Only supported when input data is XML.
2. Only supported when the length of each record is contained in a two byte prefix to the record or when each record is the same size.

Line Data Summary

To print line data, presentation services programs must know the dimensions of the page, the exact position on that page where each record must be printed, and the fonts to be used. This information is provided for line data records in an AFP resource object called the Page Definition (PageDef). The Page Definition is described in Chapter 3, "Using a Page Definition to Print Data", on page 15.

Figure 3 on page 10, and Figure 4 on page 12 summarize the valid forms of line data.

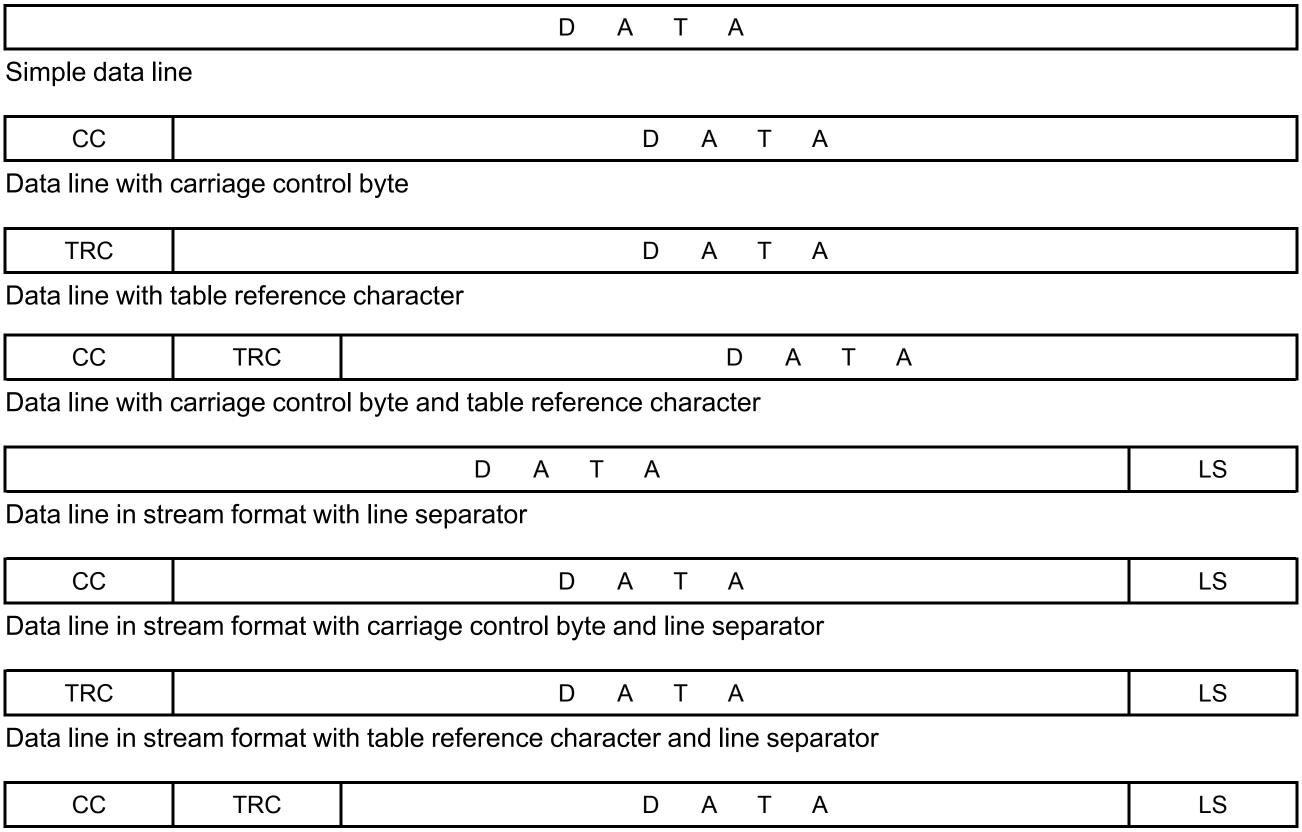
Note: In Figure 3 on page 10 and Figure 4 on page 12, the stream formats are terminated with a *line separator*. A *line separator* is normally a Line Feed character or a combined Carriage Return character and Line Feed character pair. Windows platforms typically use the Carriage Return and Line Feed pair as the *line separator*. The *line separator* code points vary based on the data encoding and platform. The supported *line separators* are:

- EBCDIC data: Line Feed (X'25').
- ASCII and UTF-8 data: Line Feed (X'0A') or Carriage Return (X'0D') and Line Feed (X'0A') pair.
- UTF-16BE: Line Feed (X'000A') or Carriage Return (X'000D') and Line Feed (X'000A') pair.

- UTF-16LE: Line Feed (X'0A00') or Carriage Return (X'0D00') and Line Feed (X'0A00') pair. Note that when the input data is UTF-16LE (little-endian byte order), the program that processes the line data needs to convert the data to big-endian byte order. Big-endian byte order is the only byte order supported in AFP environments.

Figure 3. Valid Line Data Records

(Part 1 of figure)



Simple data line

Data line with carriage control byte

Data line with table reference character

Data line with carriage control byte and table reference character

Data line in stream format with line separator

Data line in stream format with carriage control byte and line separator

Data line in stream format with table reference character and line separator

Data line in stream format with carriage control byte, table reference character, and line separator

Note: The data portion and line separators of the valid records above can be encoded using Unicode Standard encoding UTF-16 or UTF-8.

(Part 2 of figure)

BOM	DATA

Unicode data line with Byte Order Mark

CC	BOM	DATA

Unicode data line with carriage control byte and Byte Order Mark

TRC	BOM	DATA

Unicode data line with table reference character and Byte Order Mark

CC	TRC	BOM	DATA

Unicode data line with carriage control byte, table reference character, and Byte Order Mark

BOM	DATA	LS

Unicode data line in stream format with Byte Order Mark and line separator

CC	BOM	DATA	LS

Unicode data line in stream format with carriage control byte, Byte Order Mark, and line separator

TRC	BOM	DATA	LS

Unicode data line in stream format with table reference character, Byte Order Mark, and line separator

CC	TRC	BOM	DATA	LS

Unicode data line in stream format with carriage control byte, table reference character, Byte Order Mark, and line separator

Note: For a description of the BOM (Byte Order Mark) see “Unicode Line Data” on page 13. The BOM is allowed only on the first record and applies to all records in the print file.

Record-Format Line Data

Another form of line data that is supported by presentation services programs and formatted by a Page Definition resource is *record-format* line data. With this format, each line data record contains a 1 to 250-byte record identifier, which selects the Record Descriptor (RCD) in a record-format Data Map that is used to format the line data. A carriage control (CC) byte is optional but is ignored if specified; table reference characters (TRCs) are not supported. Many functions used in the Line Descriptor (LND) to format traditional line data are also used in the RCD to format record-format line data. Others, such as header and trailer processing, are unique to RCDs.

Figure 4 summarizes the valid forms of record-format line data.

Figure 4. Valid Record-Format Line Data

Record ID	DATA

Record format line data

CC	Record ID	DATA

Record format line data with carriage control byte

Record ID	DATA	LS

Record format line data in stream format with line separator

CC	Record ID	DATA	LS

Record format line data in stream format with carriage control byte and line separator

Note: The data portion and line separators of the valid records above can be encoded using Unicode Standard encodings UTF-16 or UTF-8.

BOM	Record ID	DATA

Unicode Record format line data with Byte Order Mark

CC	BOM	Record ID	DATA

Unicode Record format line data with carriage control byte and Byte Order Mark

BOM	Record ID	DATA	LS

Unicode Record format line data in stream format with Byte Order Mark and line separator

CC	BOM	Record ID	DATA	LS

Unicode Record format line data in stream format with carriage control byte, Byte Order Mark, and line separator

Note: For a description of the BOM (Byte Order Mark) see "Unicode Line Data" on page 13. The BOM is allowed only on the first record and applies to all records in the print file.

Unicode Line Data

The data portion of the valid line data formats shown in Figure 3 on page 10 and in Figure 4 on page 12 can be encoded using Unicode Standard encodings UTF-16 or UTF-8. The Unicode Standard recommends that a byte order mark (BOM) be the first sequence of bytes in the data. This is to accommodate platforms, such as Windows, that use the little-endian byte order. It also serves as a signature to identify Unicode text. The Byte Order Marks supported are:

- UTF-8: X'EFBBBF'
- UTF-16BE (big-endian byte order): X'FEFF'
- UTF-16LE (little-endian byte order): X'FFFE'

The Byte Order Mark is optional. If used, the BOM is only on the first line or record of line data in the print file. It is recommended that switching the encoding in the page definition be avoided. If the font selected indicates the data is UTF-16 and the BOM is not used, big-endian byte order is assumed. When the BOM is not used, switching the encoding in the page definition should not pose any problems.

Unicode encoding is subject to these restrictions in an AFP environment:

- Shift-out/Shift-in (SOSI) controls are not used in Unicode to signify a shift into and out-of DBCS processing. Therefore, it is not possible to switch processing between Unicode encoding and single-byte (SBCS) encoding within a line data field or record using SOSI as described in "Processing Line Data with Shift-Out/ Shift-In (SOSI) Controls" on page 32. That is, when a line data field is processed with a Page Definition, either the whole field is treated as Unicode-encoded, or none of it is treated as Unicode-encoded.
- If the Byte Order Mark used in UTF-16 data indicates the data is in little-endian byte order, programs that process the UTF-16 data will need to convert little-endian to big-endian byte order.
- If carriage control bytes (CC) or table reference character bytes (TRC) are used with UTF-16 encoded data, the CC and TRC bytes remain 1 byte fields and are not encoded in UTF-16.

XML Data

XML data may be formatted using a Page Definition resource, however this is subject to the following restrictions:

- Carriage Control (CC) and Table Reference Characters (TRC) are not supported.
- The data is encoded using one of the following:
  - EBCDIC (Single-byte only)
  - ASCII (Single-byte only)
  - UTF-8
  - UTF-16 (including surrogates; see Unicode Line Data for information about byte order)

  Application Note: When using FOCA fonts (fonts mapped with an MCF in the AEG) to process XML data, the following can occur:
  - If the data is encoded in ASCII or UTF-8 and the font specifies an encoding scheme of Unicode Presentation, the processor of the XML data may convert the data to UTF-16BE.
  - If the data is encoded in UTF-16 and the font specifies an encoding scheme of PC Data SBCS, the processor of the XML data may convert the data to ASCII. This conversion might result in unprintable code points.
- MO:DCA data cannot be mixed with XML data.

For a description of XML Data, refer to the XML specification, *Extensible Markup Language (XML) 1.0*, which can be found at the World Wide Web Consortium web site, `http://www.w3.org/`.

MO:DCA Data Summary

In contrast to line data, fully composed page data, or MO:DCA (AFP) data, contains control information such as position, orientation, and font selection intermixed with the data to be printed. Presentation services programs accept a MO:DCA document and generate the corresponding IPDS commands needed to drive a printer. An external Page Definition resource is not needed with MO:DCA data because all the formatting information is already included in the document.

MO:DCA data is fully described in the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Combining Line Data with MO:DCA Structured Fields

It is possible to intermix line data records and MO:DCA structured fields in a single file and send the mixed data to a presentation services program for printing. This permits the addition of image, graphics, and bar code objects to existing line data output. Applications can be written to generate line data or other objects as needed to produce the desired final print product.

Note: MO:DCA structured fields cannot be combined with XML data.

Line data and MO:DCA records cannot be mixed haphazardly. Chapter 4, “Mixed Documents: Adding MO:DCA Structured Fields to Line Data”, on page 39 provides guidelines on the valid combinations.

The Function of the Page Definition

Any print file that contains line data, whether alone or in combination with MO:DCA structured fields, requires a Page Definition for printing using presentation services programs. The Page Definition is necessary to establish the environment for each page and to position each line of print.

A number of Page Definitions mapping common page layouts are provided with some AFP software products and some AFP products allow users to create their own Page Definitions.

Chapter 3. Using a Page Definition to Print Data

One of the major enhancements provided by AFP to existing line-data applications is the ability to print application-generated output in different formats without making any application changes. This function is called *outboard formatting*, and is provided by the Page Definition resource object.

Page Definitions are supported by presentation services programs for z/OS, z/VM, z/VSE, AIX, Linux, Windows, and iSeries. A presentation services program uses Page Definitions to map the line data produced by application programs. Page Definitions are *not* used when printing fully composed MO:DCA documents, as formatting information is already included internally in these documents.

Page Definition names are provided in job control statements. Any print file can be associated with a Page Definition by using the appropriate parameters in the job control statements for the applicable operating system. See the reference publications for your print server and operating system for complete information.

Common Examples of Page Definition Use

Many users want to take advantage of AFP capabilities that provide multiple-up printing or rotated printing without any need to change the application that generates the output. Line data can be printed in any desired format by creating a Page Definition that describes that format. Presentation services program software includes many Page Definitions that address common user needs, such as printing two-up output on 8.5 by 11 inch paper.

One example of multiple-up printing is provided by IBM-supplied Page Definition W12883. This Page Definition prints two pages of 64 lines each, side by side in landscape mode on letter-sized paper. The data is printed at 8.2 lines per inch, so a 15-pitch or smaller font must be used to prevent the lines from overlapping.

Another example applies to users of continuous-forms impact printers who install a cut-sheet laser printer and begin to use letter-size sheets of paper, rather than the larger forms found on impact printers. The impact printers always printed in the ACROSS direction on paper whose width exceeded its length. But ACROSS printing on cut-sheet paper is generally considered to mean printing parallel to the narrow edge, not the wide edge. A Page Definition that prints in the DOWN direction, or in the landscape orientation, can be used to get the same result with letter-size paper on a cut-sheet printer as with larger forms on an impact printer.

Using More than One Page Definition

When a line-data file (such as a SYSOUT file produced by a System/370 application program) is printed, only one Page Definition can be used to map the output format of that file. However, multiple copies of the file can be printed, each one using a different Page Definition, if the appropriate job control statements are used. The actual syntax varies depending on the operating system. An example for z/OS is shown in Figure 5 on page 16. By using a job stream similar to the one shown in the figure, multiple copies of a line-mode data set can be generated, each one in a different format.

This example produces three different collated copies of the entire output file, each one formatted using a different Page Definition. The same approach can be used with Form Definitions. Each OUTPUT statement includes a different Form Definition name to invoke various options such as overlays, paper source, simplex, or duplex.

Although only one Page Definition and Form Definition can be used when printing a single file, the internal structure of Page Definitions and Form Definitions includes multiple sets of formatting rules. These sets of rules are called *Data Maps* (also called *Page Formats*) in the Page Definition and *Medium Maps* (also called *copy groups*) in the Form Definition. The *Invoke Data Map* and *Invoke Medium Map* structured fields can be written in the output by an application program and used to switch between maps as printing proceeds. This makes it possible for subsets of the file to be presented in different formats. Examples will be provided later in this chapter and in Chapter 4, “Mixed Documents: Adding MO:DCA Structured Fields to Line Data”, on page 39.

When a Page Definition containing more than one Data Map, or a Form Definition containing more than one Medium Map, is used, the one that appears physically first in the resource object is selected as the default.

Figure 5. Printing a Data Set in z/OS Multiple Times with Different Page Definitions

```
//PRINTJOB  JOB ...//STEP1     EXEC  PGM=IEBGENER
//OUT1      OUTPUT PAGEDEF=PD1
//OUT2      OUTPUT PAGEDEF=PD2
//OUT3      OUTPUT PAGEDEF=PD3
  ⋮
//SYSUT2    DD  SYSOUT=A,OUTPUT=(*.OUT1,*.OUT2,*.OUT3)
```

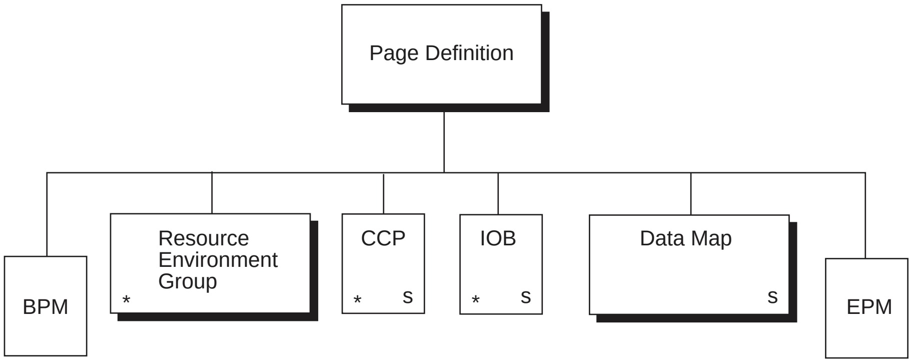
Page Definition Structure

A Page Definition is required to compose line data into pages for printing on page printers. A Page Definition consists of one or more *Data Maps* that define the page environment and provide instructions for mapping each line of data to the page.

A Page Definition object can be referenced from a library defined to a presentation services program or can be included inline at the beginning of a print file in some system environments. The structured fields in the Page Definition conform to the MO:DCA architecture rules for structured fields. These rules are summarized in Chapter 5, “Structured Fields in a Page Definition and in Line Data”, on page 65 of this publication and are formally defined in the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

A Page Definition optionally can contain one or more Conditional Processing Control (CCP) structured fields. Conditional processing permits the application programmer to define tests on selected data fields in the input line records and to specify actions to take when the conditions of the test are satisfied. Figure 6 shows the structure of a Page Definition.

Figure 6. Page Definition Structure



* = optional
s = can appear more than once

The structured fields and objects that compose a Page Definition are as follows. (Chapter 5, “Structured Fields in a Page Definition and in Line Data”, on page 65 describes the structured fields.)

BPM (Begin Page Map)

Begins a Page Definition resource object. An optional token name may be specified to identify the object.

Resource Environment Group

The Resource Environment Group (REG) identifies complex resources that need to be loaded in the presentation device before the pages that follow are processed.

CCP (Conditional Processing Control)

The CCP structured field is optional but can occur multiple times in the Page Definition. This structured field appears at the beginning of the Page Definition, outside any of the Data Map definitions, since it can be used by any Data Map to control switching among Data Maps in the Page Definition and Medium Maps in the Form Definition. The CCP defines the condition to be tested, the data value to be used to compare against the application data, the action to be taken based on the result of the test, and when the action is to be taken.

A single CCP can make multiple tests, and Page Definitions can contain multiple conditions to form complex testing sequences. These multiple conditions are reflected in multiple CCPs.

Each CCP in a Page Definition object has a unique identifier. The LND structured fields of a Data Map use this identifier to invoke conditional processing. Each LND using conditional processing specifies the length and position of the field in the application data record to be tested. Different LNDs can invoke the same CCP multiple times in the same Data Map definition.

See “Conditional Processing Control (CCP)” on page 78 for details about the CCP structured field.

IOB (Include Object)

The IOB structured field is optional but can occur multiple times in the Page Definition. The IOB appears at the beginning of the Page Definition, following the CCP structured fields. The IOB is processed when it is referenced by an LND or RCD. The reference consists of an ID that is specified on the LND or RCD and that must match the ID on an IOB.

Data Map Object

Provides specific line definitions and mapping instructions for composing line data into a presentation page format. A single Page Definition object may specify multiple Data Maps. Different Data Maps in the Page Definition can be selected by using the Invoke Data Map structured field or by using conditional processing.

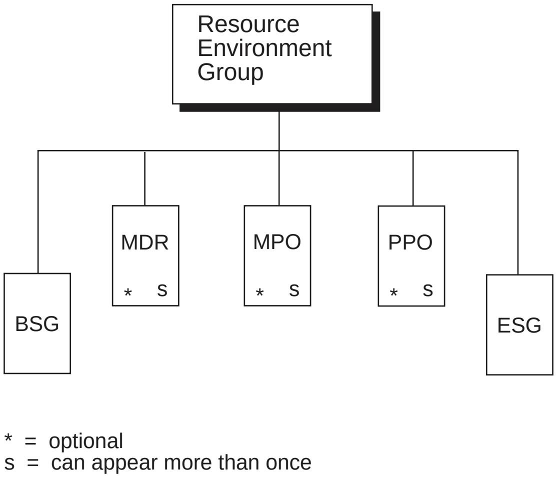
EPM (End Page Map)

Ends a Page Definition resource object. Any name specified in the EPM must match the name specified in the BPM.

Resource Environment Group

Figure 7 shows the structure of a Resource Environment Group (REG) in the Page Definition.

Figure 7. Resource Environment Group Structure for a Page Definition



A Resource Environment Group (REG), when specified in a Page Definition, is associated with a print file. The REG is used to identify complex resources, such as high-resolution color images, that need to be downloaded to the presentation device before the pages that follow are processed. The scope of a REG in the Page Definition is the line-format data in the print file. When a print file contains multiple line-format data and mixed data documents, the REG applies only to the line-format data documents in the print file. For a definition of line-format data, see Figure 33 on page 172. Line-format data may be bounded by explicit BDT/EDT pairs or by implicit BDT/EDT pairs.

Architecture Note: To get the optimum performance benefit from the REG in the Page Definition, the print file should contain only line-format data, and only large, complex objects should be mapped in the REG. This will allow the line-format data to be treated as a single document, and the REG will cause all mapped objects to be preloaded to the printer at the start of that document.

The REG in the Page Definition is not applied to MO:DCA documents in the print file. The mapping of resources in a REG is optional. Resources mapped in a REG must still be mapped in the AEG for the Data Map that uses the resources. The structured fields that compose a Resource Environment Group are as follows.

BSG (Begin Resource Environment Group)

Begins a Resource Environment Group. A token name may be specified to identify the REG.

MDR (Map Data Resource)

The MDR structured field is optional but can occur multiple times in a REG. The MDR specifies a resource that is required for presentation. The resource is identified with a file name, the identifier of a begin structured field for the resource, or any other identifier associated with the resource. The MDR may additionally specify a local or internal identifier for the resource object. Such a local identifier may be embedded one or more times within an object's data.

MPO (Map Page Overlay)

The MPO specifies overlay resources required for presentation. It is optional and can occur multiple times in a REG.

PPO (Preprocess Presentation Object)

The PPO structured field is optional but can occur multiple times in a REG. The PPO specifies presentation parameters for a data object that has been mapped as a resource. These parameters allow the presentation device to preprocess and cache the object so that it is in presentation-ready format when it is included with a subsequent include structured field in the document. Such preprocessing may involve a rasterization or RIP of the object, but is not limited to that. The resource is identified with a file name, the identifier of a begin structured field for the resource, or any other identifier associated with the resource. The referenced resource and all required secondary resources must previously have been mapped with an MDR or an MPO in this environment group.

Note: Preprocessing is not supported for objects that are included with structures that are outside the document. Examples of such objects are medium overlays and PMC overlays, both of which are included with structures in the Form Definition.

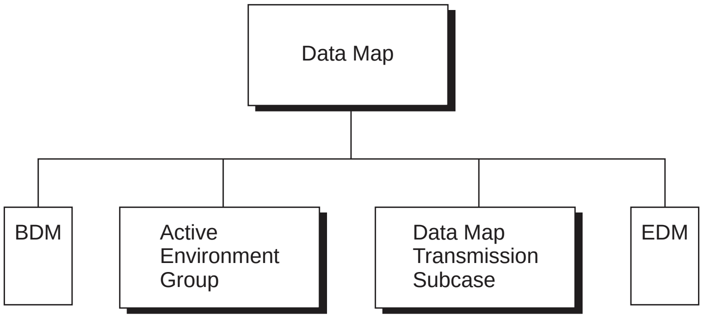
ESG (End Resource Environment Group)

Ends a Resource Environment Group.

Data Map Structure

Figure 8 shows the structure of a Data Map, also called a Page Format.

Figure 8. Data Map Structure for a Page Definition



Each Page Definition must include at least one Data Map. Structured fields in the Data Map accomplish the page layout functions similar to those provided by FCBs used with non-AFP printers, but many additional functions are available.

Each Data Map provides instructions for mapping line data to a page. The number of Data Maps that can be included in a Page Definition is limited only by practical considerations such as whether the total size of the Page Definition will be so large that it might not fit in a presentation services program's program storage. Each Data Map in the Page Definition can contain entirely different information about how a page should appear, so different Data Maps can be used from one page to the next with output produced by a line-data application.

The Data Maps in a Page Definition can select two types of line data processing:

- Traditional line data containing optional CCs and TRCs are processed using LNDs in the Data Map Transmission Subcase.

- Record-format line data containing record IDs and optional CCs are processed using RCDs in the Data Map Transmission Subcase.

All Data Maps in the Page Definition must specify the same line data processing.

The application can select which Data Map to use by writing an Invoke Data Map structured field in the output file or by using conditional processing in the Page Definition to select a Data Map based on the value of a field in the application data stream. Examples of using an IDM can be found in Chapter 4, “Mixed Documents: Adding MO:DCA Structured Fields to Line Data”, on page 39. Examples of conditional processing appear at the end of this chapter.

The Data Map consists of two parts: the Active Environment Group and the Data Map Transmission Subcase. Bracketing them are the Begin Data Map and End Data Map structured fields. The format of these structured fields is as follows:

BDM (Begin Data Map)
Begins a Data Map. A one-to-eight character token name is required to identify the Data Map. A one-byte code indicates whether the Data Map contains LNDs or RCDs. For the latter, the BDM may contain a triplet that specifies the page margins, a triplet that specifies page count controls, and a triplet that specifies an encoding scheme.

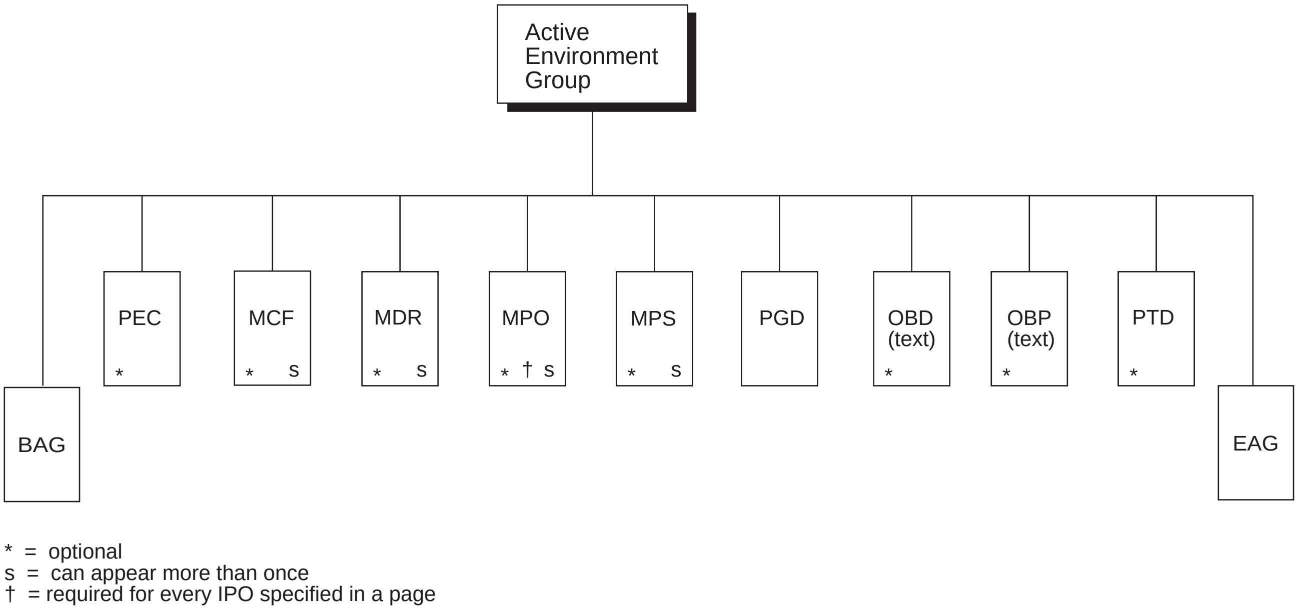
EDM (End Data Map)
Ends a Data Map. Any name specified in the EDM must match the name specified in the BDM.

The Active Environment Group establishes the page environment, including page size, and can contain the names of resources, such as fonts and page segments, that are to be mapped. The Data Map Transmission Subcase specifies the position, orientation, color, and font selection for text, the identification of data fields to be suppressed, any “fixed text” for the page, and any conditional processing tests and actions. It may also specify objects to be included on the page.

Active Environment Group Structure

Figure 9 shows the structure of an Active Environment Group (AEG) in the Data Map.

Figure 9. Data Map Active Environment Group Structure for a Page Definition



The Active Environment Group contains structured fields that describe the features and characteristics of the entire page. Page size, names of fonts, page segments, page overlays, and identifiers of objects to be used are all part of the AEG. Because a page might consist entirely of text or entirely of image (as in a page segment), most of these fields are optional. The only required structured field in the AEG is the Page Descriptor, which contains the size of the page. If an application attempts to place data outside the page boundaries, a positioning data-check error will be generated by the printer and reported by the presentation services program.

Font Lists

The Map Coded Font (MCF) and Map Data Resource (MDR) structured fields may be used in the AEG to list fonts to be used on the page. If either is used, each font is assigned a local identifier that is used in the body of the page to select the font for a given line or field.

- Record-format processing

  When the Page Definition specifies record-format processing, font specifications external to the Page Definition are ignored.

  Each font that is used must be mapped with an MCF or MDR in the AEG, and the MCF or MDR should specify the encoding scheme with an Encoding Scheme ID (X'50') triplet. The values supported in the ESidCP field of the Encoding Scheme ID triplet when printing page numbers with record-format processing are:
  – X'6100'—EBCDIC Presentation SBCS
  – X'6200'—EBCDIC Presentation DBCS
  – X'2100'—PC Data SBCS (ASCII)
  – X'8200'—Unicode Presentation

  The values supported in the ESidUD field of the Encoding Scheme ID triplet when printing page numbers with record-format processing are:
  – X'7200'—UTF-16, including surrogates; byte order is big-endian (UTF-16BE)
  – X'7807'—UTF-8

  The code points used for printing page numbers are:
  – X'F0'–X'F9' for EBCDIC Presentation SBCS
  – X'42F0'–X'42F9' for EBCDIC Presentation DBCS
  – X'30'–X'39' for PC Data SBCS (ASCII) or UTF-8
  – X'0030'–X'0039' for Unicode Presentation or UTF-16

- XML processing

  When the Page Definition specifies XML Data processing, font specifications external to the PageDef are ignored.

  Each font that is used must be mapped with an MCF or MDR in the AEG, and the MCF or MDR must specify the Encoding Scheme ID (X'50') triplet. The values supported in the ESidCP field of the Encoding Scheme ID triplet when formatting XML data with a Page Definition are:
  – X'6100'—EBCDIC Presentation SBCS
  – X'2100'—PC Data SBCS (ASCII)
  – X'8200'—EBCDIC Presentation

  The values supported in the ESidUD field of the Encoding Scheme ID triplet when formatting XML data with a Page Definition are:
  – X'7200'—UTF-16, including surrogates; byte order is big-endian (UTF-16BE)
  – X'7807'—UTF-8

  The code points used for printing page numbers are:
  – X'F0'–X'F9' for EBCDIC Presentation SBCS
  – X'30'–X'39' for PC Data SBCS (ASCII) or UTF-8
  – X'0030'–X'0039' for Unicode Presentation or UTF-16

Table Reference Characters

If the data to be printed contains 3800-style table reference characters, font information is required to map each table reference character to the name of the font to be used when the data is printed. This information can be provided either by font character-set names in job control statements accompanying the data to be printed or by the fonts mapped in the AEG in the Page Definition. When no fonts are mapped in the AEG but font character-set names are specified in the job control, the first character set specified corresponds to TRC 0, the second to TRC 1, and so forth.

In z/OS, z/VM, and z/VSE, the maximum number of characters allowed in the character-set name (CHARS) parameter was four. This presented no problem when 3800 compatibility-mode character sets were used, as none of them had names of more than four characters. But the typographic fonts available for page-mode printing have eight-character names (including a two-character font prefix), and as a result cannot be coded in the CHARS parameter. To associate a table reference character with an eight-character font name, a Page Definition must be built that explicitly makes that mapping. A Page Definition is also required if five or more fonts are to be used. Page Printer Formatting Aid (PPFA) is a software product available from IBM and Ricoh. Figure 10 provides an example of PPFA source code that could be used to build a Page Definition that addresses both requirements. Here, six table reference characters are defined and each one is associated with a different font of the Sonoran Sans Serif family.

Figure 10. PPFA Code for Page Definition with Six TRCs to Select Typographic Fonts

```
SETUNITS LINESP 6 LPI;
PAGEDEF TRCXMP
        WIDTH 8.2 IN HEIGHT 10.8 IN
        REPLACE YES;
  FONT ZERO A0758C;  /* EIGHT-POINT SANS SERIF BOLD   */
  FONT ONE  A0759C;  /* NINE-POINT SANS SERIF BOLD    */
  FONT TWO  A0750C;  /* TEN-POINT SANS SERIF BOLD     */
  FONT THREE A075BC; /* 12-POINT SANS SERIF BOLD      */
  FONT FOUR A0559C;  /* NINE-POINT SANS SERIF ROMAN   */
  FONT FIVE A0550C;  /* TEN-POINT SANS SERIF ROMAN    */

  PAGEFORMAT JSTRC;
    TRCREF 0 FONT ZERO;
    TRCREF 1 FONT ONE;
    TRCREF 2 FONT TWO;
    TRCREF 3 FONT THREE;
    TRCREF 4 FONT FOUR;
    TRCREF 5 FONT FIVE;

    PRINTLINE CHANNEL 1
              POSITION .1 IN .2 IN REPEAT 20;
    ENDSUBPAGE;
```

The rules for coding Table Reference Characters are different for page mode printers and for the 3800 running in compatibility mode. Table 8 on page 23 summarizes the differences.

Table 8. Use of TRCs in Page Mode and 3800 Compatibility Mode

Function	Compatibility Mode	Page Mode
Number of table reference characters supported for a single output file	4	127
Valid hexadecimal values for table reference characters	X'F0'–X'F3'	X'F0'–X'F3' or X'00'–X'7E' for 4 or fewer TRCs; X'00'–X'7E' for more than 4 TRCs
How table reference characters are associated with fonts	With character set names in job control language	Same as compatibility mode for 4 or fewer TRCs; with font names in the AEG of the Data Map for more than 4 TRCs

For compatibility with 3800-1 applications, AFP print servers accept TRC values of X'F0' through X'F3' when four or fewer TRCs are used. TRC values of X'00' through X'7E' are valid regardless of how many fonts are used.

The Line Descriptor structured fields in the Data Map contain a bit that indicates which type of TRC to use. PPFA and PMF set this bit to B'1' to indicate compatibility TRCs when four or fewer TRCs are described in the Page Definition. These software programs set the bit to B'0' when more than four TRCs are used.

Note: Regardless of whether font character set names are specified in the job control or not, if fonts are mapped in the AEG:

- Table reference character 0 (X'00' or X'F0') selects the first font mapped in the Active Environment Group (AEG) of the Data Map; table reference character 1 (X'01' or X'F1') selects the second font mapped in the AEG; and so on. This historically positional selection of fonts mapped in the AEG precludes the use of a mixture of fonts mapped with MCFs and fonts mapped with MDRs. TRCs may be used when all fonts in the AEG are mapped using MCFs only or MDRs only.
- A table reference character higher than 127 selects the first font mapped in the AEG of the Data Map.
- A table reference character higher than the number of fonts mapped defaults to the first font mapped in the AEG of the Data Map.

Page Overlays

If the application uses the Include Page Overlay structured field to place overlays dynamically at any point on the page, a Map Page Overlay structured field must be included in the Active Environment Group containing the name of each overlay to be used.

Page Segments

A Map Page Segment structured field is not required in the Active Environment Group for each page segment to be used by the application, but printer throughput improves if MPS structured fields are included. Mapped page segments are loaded to the printer the first time they are included and are not reloaded on subsequent invocations. These are called *hard* page segments. When a page segment is not mapped in the Active Environment Group of the currently active Data Map, the page segment data is loaded to the printer every time the segment is included by an Include Page Segment structured field. Such segments are called *soft* page segments.

Data Objects

Data objects that are included with an IOB structured field, such as EPS objects and IOCA objects, can be mapped using the MDR structured field. An MDR for such objects is not required in the AEG of the Data Map, but might improve printer throughput if used. Mapped data objects are loaded to the printer the first time they are included and are not reloaded on subsequent includes.

Color Management

A Color Management Resource (CMR) can be associated with a page, a data object included on the page with an Include Object (IOB) structured field, or a GOCA or BCOCA object generated by the page definition. Associating a CMR is accomplished by using the MDR structured field to reference the CMR as a resource in the AEG for the Data Map. The CMR reference will be identified as targeted to the page or data object. If a data object is included on a page with an Include Object (IOB) structured field or generated by the page definition, a CMR can be associated with this object by specifying the name of the CMR on the IOB, LND, RCD, or XMD as an external resource reference and then referencing the CMR with a MDR in the AEG of the Data Map. See the *Mixed Object Document Content Architecture (MO:DCA) Reference* and *Color Management Object Content Architecture Reference* for more information on Color Management in an AFP environment.

Structured Fields

The structured fields that comprise the Active Environment Group in a Data Map are as follows: (See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for a complete description of these structured fields.)

BAG (Begin Active Environment Group)
Begins an Active Environment Group. A token name may be specified to identify the AEG.

PEC (Presentation Environment Control)
The Presentation Environment Control structured field specifies parameters that affect the rendering of presentation data and the appearance that is to be assumed by the presentation device. The scope of the Presentation Environment Control structured field is the page generated using the Data Map that contains this structured field.

Note: The PEC structured field in the AEG for the Data Map is only used to specify the rendering intent for the page using the Rendering Intent (X'95') triplet; all other PEC triplets are ignored.

MCF (Map Coded Font)
Identifies each font resource object used in the page and assigns each a 1-byte local identifier. The strategic format of the MCF structured field is called the MCF-2; the coexistence format is called the MCF-1. For any reference to MCF throughout this book, refer to the *Mixed Object Document Content Architecture (MO:DCA) Reference* for further description.

MDR (Map Data Resource)
Identifies data object resources that are to be downloaded to the printer for subsequent use in the page.

MPO (Map Page Overlay)
Identifies overlay object resources used in the page. Each overlay referenced by an Include Page Overlay structured field in the page must be mapped in an MPO structured field.

MPS (Map Page Segment)
Identifies page segments used on the page that are to be downloaded to the printer.

PGD (Page Descriptor)
Specifies the units of measure for the page presentation space and the size of the page. This parameter is required.

OBD (Object Area Descriptor)
Specifies the units of measure for the text output area and the size of the area. The OBD is optional. If specified, the units of measure must be the same as those specified for the page in the PGD, and the output area size must be the same size as the page.

OBP (Object Area Position)

Specifies the origin and orientation of the text output area on the page, as well as the origin and orientation of the text presentation space on the output area. The OBP is optional. If specified, the origin of the output area and the origin of the text presentation space must be the same as the origin of the page and the orientation of the output area and of the text presentation space must be 0°.

PTD (Presentation Text Descriptor)

Specifies the units of measure for the text presentation space and the size of the space. For composed page text objects enveloped with BPT and EPT structured fields, the PTD may also specify initial text conditions for the text object. The PTD is required in the AEG if the page contains presentation text objects. If the PTD is specified, the text presentation space units of measure and size must match the page presentation space units and size specified in the PGD. This descriptor has two formats:

- PTD-1, formerly called CTD, specifies only the text presentation space units of measure and size.
- PTD-2 specifies the text presentation space units of measure, expands the fields containing the presentation space extents from two bytes to three bytes, and allows initial text conditions to be specified for composed page text objects enveloped with BPT and EPT. These initial text conditions are set whenever a BPT structured field starts a new text object and may be specified on the PTD-2 using the PTOCA control sequences shown in Table 9.

Note that whenever a BPT is encountered in the data stream, AFP servers set the following default page-level initial text conditions *before* the PTD initial conditions are set:

Parameter	**Value**
Initial (I,B) Presentation Position	(0,0)
Text Orientation	0°,90°
Coded Font Local ID	X'FF' (default font)
Baseline Increment	6 lpi
Inline Margin	0
Intercharacter Adjustment	0
Text Color	X'FFFF' (default color)

EAG (End Active Environment Group)

Ends the AEG. Any name specified in the EAG must match the name specified in the BAG.

Table 9. Initial Text Conditions in PTD-2

Initial Text Condition Parameter	Control Sequence
Baseline Increment	Set Baseline Increment (SBI)
Coded Font Local ID	Set Coded Font Local (SCFL)
Initial Baseline Coordinate	Absolute Move Baseline (AMB)
Initial Inline Coordinate	Absolute Move Inline (AMI)
Inline Margin	Set Inline Margin (SIM)
Intercharacter Adjustment	Set Intercharacter Adjustment (SIA)
Extended Text Color	Set Extended Text Color (SEC)
Text Color	Set Text Color (STC)
Text Orientation	Set Text Orientation (STO)

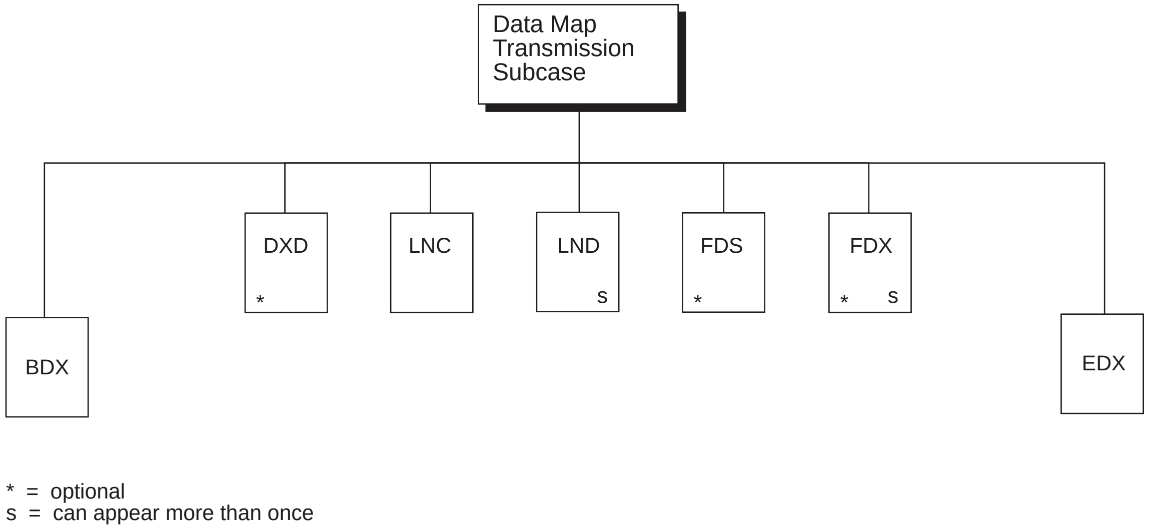
Data Map Transmission Subcase

A Data Map Transmission Subcase can contain LNDs, RCDs, or XMDs, but not a mixture.

Data Map Transmission Subcase with LNDs

Figure 11 shows the structure of a Data Map Transmission Subcase with LNDs.

Figure 11. Data Map Transmission Subcase with LNDs



The principal function of the Data Map Transmission Subcase with LNDs is to map the lines of data to the page. Each line on a page is represented by a Line Descriptor structured field, which contains the horizontal and vertical position on the page where the line is to appear. Rotation and font information is also contained in the Line Descriptors, as is the association with any conditional processing controls used to test data on the current line. The Line Descriptor structured fields are generally used to map lines of text, but they can also be used to position page segments or page overlays or to present line data code points as a bar code. An Include Page Segment or Include Page Overlay structured field that contains a value of X'FFFFFF' in the X-axis positioning parameter, the Y-axis positioning parameter, or both indicates that the page segment or overlay is to be placed at the X-axis or Y-axis position specified by the current LND.

Functions that originated with older printers are also implemented in Line Descriptors. If carriage control skip-to-channel codes are used with the data, each channel code must be defined in at least one Line Descriptor (LND) in the Page Definition; this LND defines the page position associated with that channel code number. Carriage control characters that call for double spacing, triple spacing, or space suppression cause a presentation services program to skip over Line Descriptors or reuse the same Line Descriptor in the Data Map, in a manner analogous to FCBs used with impact printers. Either ANSI or machine code carriage controls can be used, but whichever type is selected must be used for the entire print file. Part of the data cannot contain ANSI carriage controls and another part contain machine code carriage controls. In addition, if carriage control characters are used, every record in the print file must begin with a carriage-control byte, even if it is only “print with single spacing”. The type of carriage control being used must be identified in the job control associated with the print file, just as in a non-AFP environment.

The following new functions are provided in Page Definitions that are not available in FCBs. These functions are triggered by information in the Line Descriptor structured field.

- Field formatting, which is the ability to separate specific fields in a line-data record and place them anywhere on a page. Field size, orientation, placement, color, and font to be used are specified in the Line Descriptor. Fixed text may be specified in the Line Descriptor and added to data from application programs.
- Multiple-up printing, which is the ability to divide the page into sections, each with the appearance of a smaller page. This can be accomplished by defining *subpages*, which are subsets of the page, using Line Descriptors.
- Conditional processing, which is the ability to define tests on certain characters of the line data and perform actions based on the results of the tests.
- Resource object include, which is the ability to include an overlay or page segment and position it relative to the current line.
- Bar code generation, which is the ability to select a field in a record and present it as a bar code.
- Specification of spot (highlight) colors or process colors for a record or field.
- Object include, which is the ability to include a data object relative to the current line and change its position, size, and orientation.

Each Line Descriptor formats only one line-data record. The same record may be formatted multiple times on a page using the “reuse record” function in the Line Descriptor. Since Line Descriptors are contained in a Data Map Transmission Subcase whose scope is a page, they cannot be used to place a single record on more than one page.

The *text suppression* function in AFP is an implementation of the 3800-1 COPYMOD function. A combination of information in the Line Descriptor structured field in the Page Definition and the Medium Modification Control structured field in the Form Definition, it provides the same function as “spot carbons” with impact printers, where multi-part forms with carbon paper were often used. Some of these applications required that selected fields not be printed on certain copies of the output (for example, internal prices should not appear on customer copies). The same effect can be obtained with AFP printers by defining fields as *suppressible* in the Page Definition and then suppressing these fields on certain copies in the Form Definition.

Data Map Transmission Subcase with RCDs

A Data Map Transmission Subcase with RCDs has the same structure as one with LNDs except that the LNDs are replaced with RCDs. A Data Map Transmission Subcase with RCDs is used to process record-format line data instead of traditional line data.

Each record in the data contains a 1 to 250-byte Record ID and is processed by the RCD in the Data Map Transmission Subcase that contains a matching Record ID. If a CC byte is specified in the record, it must precede the Record ID and is not part of the compare. In addition to providing LND-like functions such as data position, orientation, font selection, coloring, bar code generation, and object includes, RCDs support additional functions like headers, trailers, page numbering, and graphics generation.

Data Map Transmission Subcase with XMDs

A Data Map Transmission Subcase with XMDs has the same structure as one with LNDs except that the LNDs are replaced with XMDs. A Data Map Transmission Subcase with XMDs is used to process XML data instead of traditional line data.

To process XML data, the processor must build a Qualified Tag by concatenating XML start tags. These Qualified Tags are then compared to Qualified Tags in the Data Map. The Qualified Tags in the Data Map are built by specifying a separate XML Name (X'8A') triplet on each XML Descriptor (XMD) for each XML Start tag that has to be traversed in order to process the content of an XML element. If an XMD with a matching Qualified Tag is found, the content of the XML element is formatted with that XMD. If an XMD with a matching

Qualified Tag is not found, processing resumes with the next start tag. Note that as the processor parses the XML, it must buffer the XML start tags traversed in order to have a current Qualified Tag. Each time an end tag is found, the last matching start tag is removed. For example, in the following XML hierarchy:

```
<person>
   <name>
      <first>John</first>
      <last>Doe</last>
  </name>
</person>
```

The Qualified Tag for the element <first> is {person name first}. The Qualified Tag for the element <last> is {person name last}. Notice that the tag for element <first> has been removed since its end was received prior to the start tag for element <last>. To process this “current” Qualified Tag, an XMD in the Data Map would also have a Qualified Tag made up from separate XML Name (X'8A') triplets for each XML start tag. This Qualified Tag for this XMD would match the current Qualified Tag and the XMD would be used to present the XML element content “John” on the page.

In addition to providing LND-like functions such as data position, orientation, font selection, coloring, bar code generation, and object includes, XMDs support additional functions like headers, trailers, page numbering, and graphics generation.

Data Map Transmission Subcase Structure

The structured fields that compose the Data Map Transmission Subcase are as follows. (See Chapter 5, “Structured Fields in a Page Definition and in Line Data”, on page 65 for a formal description of these structured fields.)

BDX (Begin Data Map Transmission Subcase)
Begins the Data Map Transmission Subcase.

DXD (Data Map Transmission Subcase Descriptor)
Contains constant data.

LNC (Line Descriptor Count)
Specifies the number of Line Descriptor (LND) or Record Descriptor (RCD) structured fields in the Data Map Transmission Subcase.

LND (Line Descriptor)
Specifies how the current line data should be processed. The Data Map Transmission Subcase can contain more than one LND and each LND points to the next LND used.

When the print file does not use carriage control characters, processing begins with the first LND structured field. When the data record contains a carriage control character that specifies a channel code, the first LND containing that channel code is selected to control processing. If there is no LND in the Data Map containing a channel code matching the channel code specified in the data record, an error is generated.

If an LND specifies that a conditional processing test should be performed on the current record, the LND specifies the field to be tested and the ID of the Conditional Processing Control (CCP) structured field that contains the test criteria and actions. Such LNDs do not place data on the page.

When the LND specifies that fixed text data should be printed, the data is located in the Fixed Data Text (FDX) structured field.

RCD (Record Descriptor)
Specifies how the record with matching record ID should be processed. The Data Map Transmission Subcase can contain more than one RCD.

With RCD processing, carriage controls in the data record are ignored. Processing begins with the first RCD that matches the Record ID of the first record. If a matching RCD is not found, an error is generated.

If conditional processing is to be performed on the current record, the RCD specifies the field to be tested and the ID of the CCP that contains the test criteria and actions. Such RCDs are called *conditional processing* RCDs and do not place data on the page.

When the RCD specifies that fixed text data should be printed, the data is located in the Fixed Data Text (FDX) structured field.

XMD (XML Descriptor)

Specifies how the data with matching start tags should be processed. The Data Map Transmission Subcase can contain more than one XMD.

With XMD processing, carriage controls and table reference characters in the data record are not allowed. Processing begins with the first XMD that matches the start tag. If a matching XMD is not found, the data is ignored and processing resumes with the next start tag.

If conditional processing is to be performed on the current element, the XMD specifies the field to be tested and the ID of the CCP that contains the test criteria and actions. Such XMDs are called *conditional processing* XMDs and do not place data on the page.

When the XMD specifies that fixed text data should be printed, the data is located in the Fixed Data Text (FDX) structured field.

FDS (Fixed Data Size)

If constant text is to be included in line format data, this structured field is required. The FDS specifies the number of bytes of the text that will be found in the following Fixed Data Text (FDX) structured fields. One FDS structured field is used for all FDX structured fields.

FDX (Fixed Data Text)

Must follow an FDS structured field and contains data that can be added to or used instead of line data. More than one FDX structured field is allowed.

EDX (End Data Map Transmission Subcase)

Ends the Data Map Transmission Subcase. Any name specified in the EDX must match the name specified in the BDX.

Field Formatting—LND Processing

A Page Definition may be used to break line-data records into fields that are formatted individually. This is done by building a chain of LND structured fields called a *reuse* chain.

The first LND used to process an input record is called the *base* LND. If this LND specifies flag byte bit 6=B'1' (reuse record), it is also the head of a reuse chain and points to the next LND in the chain with bytes 16–17. This next LND is used to select and process a field in the same record. If additional field processing is required, the next LND also specifies flag byte bit 6=B'1' and points to another LND to select and process another field in the record, and so on. All LNDs in a reuse chain are called *reuse LNDs*. The last LND in a reuse chain specifies flag byte bit 6=B'0' and bytes 16–17=X'0000'. This LND terminates the reuse chain.

Field Formatting—RCD Processing

Field formatting is also supported when RCDs are used to process record-format line data. The first RCD used to process an input record is called a *record* RCD. It is identified by RCDFlgs bit 6=B'0' and RCDFlgs bit 11= B'0'. If the FLDrcd parameter in a record RCD is non-zero, it specifies the RCD number of a *field* RCD that is to be used to process a field in this record. A field RCD is identified by RCDFlgs bit 6=B'1' and RCDFlgs bit 11=

B'0'. Multiple field RCDs can be chained to a record RCD in this manner. The last field RCD in this chain must specify FLDrcd= X'0000'.

Field Formatting—XMD Processing

Field formatting is also supported when XMDs are used to process XML data. The first XMD used to process a start tag is called an *element* XMD. It is identified by XMDFlgs bit 6=B'0', XMDFlgs bit 10=B'0', and XMDFlgs bit 11=B'0'. If the FLDxmd parameter in an element XMD is non-zero, it specifies the XMD number of a *field* XMD that is to be used to process a field in this element data. A field XMD is identified by XMDFlgs bit 6=B'1' and XMDFlgs bit 11=B'0'. Multiple field XMDs can be chained to an element XMD in this manner. The last field XMD in this chain must specify FLDxmd=X'0000'.

Using Conditional Processing in a Page Definition

The conditional processing function allows a different Data Map in the current Page Definition, a different Medium Map in the current Form Definition, or both to be selected for use with the next page based on characteristics of the application data stream. This provides a way to change Data Maps or Medium Maps as necessary without having to make application programming changes. The new format can take effect either before or after a specified line or a specified subpage. With LND-based Data Maps, a *subpage* is a subset of the lines presented on a page. Subpages are defined in the Data Map by the user when coding a Page Definition and are often used to create multiple-up Page Definitions. Subpages are ignored with RCD-based and XMD-based Data Maps, that is, each page is a single subpage.

Conditional processing is implemented by a combination of structured fields in the Page Definition. CCP structured fields specify the test to be performed and action to be taken, while LND, RCD, and XMD structured fields include the location and length of data fields to be tested and a pointer to the CCP. When the conditions of a test are satisfied, the actions that can be taken are switching to a new Data Map, switching to a new Medium Map, or both, either before or after the current line or subpage is printed. When the action takes effect, printing of the current page ends. If a new Data Map is selected, printing resumes on a new side of a sheet of paper. If a new Medium Map is selected, printing resumes on a new physical sheet. As a result, it is *not* possible to format part of a page with one Data Map and format another part of the same page with a different Data Map. Note that the Medium Map might specify the N-up function, which places multiple pages into partitions on a sheet side. When N-up is specified, switching to a new Data Map or a new Medium Map might cause printing to resume in a new partition instead of on a new sheet-side or a new sheet. For more information on N-up printing, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Conditional processing can be used to format subsets of a single output file differently or it can be used to force an eject to a new page or sheet based on some condition. Examples of these uses of conditional processing are shown below.

Using Different Formats for Different Subsets of Output

A common example of this is an application program requirement to print detail pages of a report in a different format from summary pages. Assuming that a known field in the application data stream can be tested to identify the detail records and the summary records, a Page Definition with two Data Maps can be constructed to provide the different formats without changes to the application program. Figure 12 on page 31 assumes a file where each record has identifying information in bytes 2 through 5. Records with the characters `DETL` in these positions are to use Data Map PF1 and Medium Map CG1. Records with the characters `SUMM` in these positions are to use Data Map PF2 and Medium Map CG2. Page Printer Formatting Aid (PPFA) is a software product available from IBM and Ricoh. Figure 12 on page 31 shows the PPFA code that generates a Page Definition to test these positions and to print the detail pages in the ACROSS direction and the summary pages in the DOWN direction.

Figure 12. PPFA Code for Page Definition with Conditional Processing

```
SETUNITS 1 IN 1 IN ;
LINESP 6 LPI ;
PAGEDEF CPSAM1 REPLACE YES ;
  PAGEFORMAT PF1
             WIDTH 8.2 HEIGHT 10.0
             DIRECTION ACROSS ;
    PRINTLINE REPEAT 1
              CHANNEL 1 ;
    CONDITION TEST1 START 2 LENGTH 4
              WHEN EQ 'SUMM'
                   BEFORE SUBPAGE
                   COPYGROUP CG2
                   PAGEFORMAT PF2 ;
    PRINTLINE REPEAT 59 ;
    ENDSUBPAGE ;

  PAGEFORMAT PF2
             WIDTH 10.0 HEIGHT 8.2
             DIRECTION DOWN;
    PRINTLINE REPEAT 1
              CHANNEL 1 ;
    CONDITION TEST2 START 2 LENGTH 4
              WHEN EQ 'DETL'
                   BEFORE SUBPAGE
                   COPYGROUP CG1
                   PAGEFORMAT PF1 ;
    PRINTLINE REPEAT 34 ;
    ENDSUBPAGE ;
```

Conditionally Skipping to a New Page or a New Sheet

Another common use of conditional processing is to skip to a new page or a new sheet when a control break in the output data occurs. This control break might be the start of a new customer number, a new department, or some other change in the output that requires starting on a new page, or on a new sheet of paper.

Such cases include applications that print using multiple-up format, where having data for one department appear on the left-hand half of the page while having data for a different department appear on the right-hand half of the page is not desirable. This possibility can be avoided by having the application force a completely new page when the department number changes. In PPFA, this condition is coded with the NEWSIDE parameter.

Applications that print duplex output (using both front and back of the form) probably must force a new physical sheet at a control break in the data, to avoid having output for two different user destinations on the front and back of the same sheet. In PPFA, this condition is coded with the NEWFORM parameter. For output printed multiple-up on both sides of the sheet, the NEWFORM parameter forces a new page and a new sheet. Coding both is not necessary.

A new page or sheet can be forced when using a Page Definition with only one Data Map or a Form Definition with only one Medium Map. Conditional processing can be used to re-invoke the currently active Data Map or Medium Map when the condition is satisfied. This is what happens when NEWSIDE or NEWFORM is coded in PPFA. More than one Data Map or Medium Map is required only if subsets of the output are to be formatted or handled differently based on the defined condition. Note that if the Medium Map specifies the N-up function, the new “sheet” might actually be a new N-up partition on the sheet.

The example in Figure 13 shows PPFA source code to accomplish a skip to a new page when the department number in character positions 1 through 3 changes.

Figure 13. PPFA Code for Page Definition to Skip to New Page

```
SETUNITS 1 IN 1 IN ;
LINESP 8 LPI ;
PAGEDEF NEWPG REPLACE YES
        WIDTH 10.5 HEIGHT 8.1
        DIRECTION DOWN ;
  PAGEFORMAT NEWPG ;
    PRINTLINE REPEAT 40
              CHANNEL 1
              POSITION .5     TOP ;
    CONDITION TEST1 START 1 LENGTH 3
              WHEN CHANGE
              BEFORE SUBPAGE
              NEWSIDE ;
    ENDSUBPAGE ;
```

The example in Figure 14 is similar; but in this case the skip is to a new sheet, or form, where printing of the output is resumed.

Figure 14. PPFA Code for Page Definition to Skip to New Sheet

```
SETUNITS 1 IN 1 IN ;
LINESP 8 LPI ;
PAGEDEF NEWFM REPLACE YES
        WIDTH 10.5 HEIGHT 8.1
        DIRECTION DOWN ;
  PAGEFORMAT NEWFM ;
    PRINTLINE REPEAT 40
              CHANNEL 1
              POSITION .5     TOP ;
    CONDITION TEST1 START 1 LENGTH 3
              WHEN CHANGE
              BEFORE SUBPAGE
              NEWFORM ;
    ENDSUBPAGE ;
```

Processing Line Data with Shift-Out/Shift-In (SOSI) Controls

Shift-out (SO-X'0E') and shift-in (SI-X'0F') controls are used to signal the beginning and end, respectively, of a string of double-byte code points that are to be rendered using characters from a double-byte font or rendered as a QR Code bar code symbol. SOSI processing is specified in the print request and applies to both fixed text fields and the input line data.

SOSI processing for text output is supported by two modes of font selection in the PageDef. Both modes may be intermixed in the same Page Map.

- *Record-based or field-based font selection.* In this mode, the font to be used following an SO can be uniquely selected for each record or field by specifying a non-zero shift-out font local ID in byte 26 of the LND or byte 34 of the RCD that is used to process the line data. The font used following an explicit SI is then always the primary font specified in byte 10 of the LND or byte 23 of the RCD and use of this font must be enabled with flag bit 4 = B'1'. An error condition exists if flag bit 4 = B'0'. Note that an implicit SI is assumed at the start of every record. This selects the primary font specified in byte 10 of the LND or byte 23 of the RCD, if it is enabled with flag bit 4 = B'1'.

- *Page-based font selection.* In this mode, the font to be used following an SO is the same for all records and fields on the page. LND byte 26 or RCD byte 34 is set to X'00' and the font used following an SO is the font mapped to local ID X'02' in the AEG for the Data Map. The font used following an explicit SI is the font mapped to local ID X'01' in the AEG. Note that the font used following the implicit SI at the start of every record is still the primary font specified in byte 10 of the LND or byte 23 of the RCD, as long as it is enabled with flag bit 4 = B'1'. Both an SO font and an SI font must be mapped in the AEG with the proper local IDs. The presence of only one mapped font is an error condition. If no fonts are mapped in the AEG, a presentation system default may be used.

The SO and SI controls used to delimit the strings of double-byte code points are not valid printable characters nor are they valid QR Code bar code data characters. The line data processor must either remove or convert the SO and SI characters to blanks according to the selection of SOSI processing mode in the print request.

For text output, the SOSI processing modes are described as follows:

SOSI mode **Action**

SOSI1
Specifies that each shift-out, shift-in control is to be converted to a blank and a Set Coded Font Local text control.
- Each SO (X'0E') is replaced with a blank (X'40'), followed by a PTOCA structure that contains a Set Coded Font Local text control for the font mapped to local ID X'02'.
- Each SI (X'0F') is replaced with a PTOCA structure that contains a Set Coded Font Local text control for the font mapped to local ID X'01', followed by a blank (X'40').

SOSI2
Specifies that each shift-out, shift-in control is to be converted to a Set Coded Font Local text control.
- Each SO (X'0E') is replaced with a PTOCA structure that contains a Set Coded Font Local text control for the font mapped to local ID X'02'.
- Each SI (X'0F') is replaced with a PTOCA structure that contains a Set Coded Font Local text control for the font mapped to local ID X'01'.

SOSI3
Specifies that the shift-in control is to be converted to a Set Coded Font Local text control and two blanks. A shift-out control is to be converted to a Set Coded Font Local text control.
- Each SO (X'0E') is replaced with a PTOCA structure that contains a Set Coded Font Local text control for the font mapped to local ID X'02'.
- Each SI (X'0F') is replaced with a PTOCA structure that contains a Set Coded Font Local text control for the font mapped to local ID X'01', followed by two blanks (X'4040').

SOSI4
Specifies that each shift-out, shift-in control is to be skipped and not counted when calculating offsets for the print data. The conversion of the shift-out and shift-in controls for SOSI4 is the same as for SOSI2.

Note: SOSI4 is used when double-byte character set (DBCS) text is converted from ASCII to EBCDIC. When SOSI4 is specified, the page definition offsets are correct after conversion; therefore, the user does not need to account for SOSI characters when computing offsets to various fields within the data.

For QR Code bar code output, the data is converted to Shift/JIS ASCII data. The SO and SI control characters are removed and are *not* converted to blanks and Set Coded Font Local text controls as they are for text output. The converted data is then used as the QR Code bar code data. This processing is the same for all SOSI processing modes. For SOSI4, each shift-out and shift-in control is not counted when computing offsets to various fields within the data.

When processing data with SOSI controls, the processor assumes that each line or record starts with single-byte code points. This means that the data is scanned for SOSI controls one byte at a time. After processing a shift-out control, the data is scanned two bytes at a time. The first byte of each pair is checked to see if it is a shift-in control. If a line is to start with double-byte data, the first byte in the line must be a shift-out control. This is due to the fact that single-byte code points are assumed at the start of each line. The processor also assumes that each field (including fixed text fields) starts with a single-byte code point when processing in SOSI1, SOSI2, and SOSI3 modes. The processing of fields in SOSI4 mode is different than the other SOSI modes. SOSI4 processing requires that SO and SI controls not be counted as part of the field positioning. Therefore, the record is scanned to keep track of the last SO or SI prior to the field start position. The SO or SI control found prior to the field is used to determine if the field starts with a single-byte or double-byte code point.

Notes:

1. Since table reference characters (TRCs) also might use the fonts mapped to local IDs X'01' and X'02' in the AEG of the Data Map, it is recommended that the mixing of SOSI controls and TRCs be avoided when using page-based font selection.
2. Shift-out/Shift-in controls are not used in Unicode data to signify a shift into and out of DBCS processing. Therefore, it is not possible to switch processing between Unicode encoding and single-byte (SBCS) encoding within a line data field or record. That is, when a line data field is processed with a Page Definition, either the whole field is treated as Unicode-encoded or none of it is treated as Unicode-encoded.
3. When building bar codes from line data, SOSI input data is not appropriate for bar code symbologies other than QR Code. Refer to the *Bar Code Object Content Architecture Reference* for information about the valid encoding for each bar code.
4. When using Shift-out/Shift-in controls with Record Format data using delimited fields, if the field is to print using double-byte code points, the SO control must follow the delimiter for the field.
5. Shift-out/Shift-in controls are not supported when processing XML data.

Printing Bar Codes with a Page Definition

A Page Definition can be used to print a bar code symbol using data from one of the following places:

- Line data record
- XML element
- Field in the line data record
- Field in the XML element

This is done by specifying a Bar Code Symbol Descriptor (X'69') triplet on the LND, RCD, or XMD. The presence of this triplet indicates to the presentation services program that the selected field is to be presented as a bar code symbol. The position specified by the LND, RCD, or XMD indicates the position of the symbol origin and the text orientation specified by the LND, RCD, or XMD indicates the rotation of the symbol with respect to the page X_p-axis. The data used for the bar code can be obtained from multiple fields using the Concatenate Bar Code Data (X'93') triplet. This triplet contains a start-new-symbol flag to indicate the beginning of a series of concatenated fields; therefore, multiple bar code symbols can be created on a page by reusing RCD or XMD structured fields with this flag set.

Note that the text suppression function is not supported when the field is presented as a bar code. This function is supported only for text and is ignored for other data. Note also that the bar code function is not supported on LNDs, RCDs, or XMDs that specify conditional processing (flag bit 11 = B'1'); if the Bar Code Symbol Descriptor triplet is specified it is ignored.

For improved printer throughput, all bar code symbols on a page that use the same descriptor and that specify the same rotation are grouped into a single bar code object by the presentation services program before the page is presented. To align the object presentation space X_{bc}-axis with the X-axis of the bar code symbol, the origin of the object presentation space is selected as one of the four corners of the page based on the LND, RCD, or XMD text orientation. The bar code presentation space origin is therefore made coincident with the current text coordinate system (I,B) origin. For example, if an LND specifies a (90°,180°) text orientation, the symbol rotation is 90° and the origin of the bar code object presentation space is the top-right corner of the page. The extents of the bar code object presentation space are determined by the extents of the page presentation space. For example, if the origin of the object presentation space is the top-right corner of the page, the X-extent of the object presentation space is the Y_p-extent of the page and the Y-extent of the object presentation space is the X_p-extent of the page. The symbol origin offset from the object presentation space origin and from the current text (I,B) coordinate system origin is specified by the *IPos* and *BPos* parameters of the LND, RCD, or XMD. The units of measure for the bar code object presentation space, used for determining symbol origin offsets, are the same as those defined on the page (X_p,Y_p) presentation space in the PGD structured field of the Active Environment Group (AEG) for the Data Map.

The presentation services program also defines an object area presentation space for the object that is identical in size, position, and units of measure to the bar code presentation space. The rotation of the object area presentation space about the page X_p-axis is the same as the rotation of the bar code symbol about this axis, which is the same as the text orientation specified in the LND, RCD, or XMD.

Printing Graphics with a Page Definition

A Page Definition can be used to generate simple graphics primitives such as lines, boxes, circles, and ellipses when record-format line data is processed with RCDs or XML data is processed with XMDs. This is done by specifying a Graphics Descriptor (X'7E') triplet on the RCD or XMD. This triplet might specify a complete graphics primitive, as is always the case with a full arc, or it might specify the beginning or end of the primitive.

For improved printer throughput, all graphics primitives on a page with the same descriptor and the same orientation are grouped into a single graphics object by the presentation services program before the page is presented. The origin for the graphics object area is one of the four corners of the page as determined by the text orientation specified in the *TxtOrent* parameter of the RCD or XMD and is therefore coincident with the current (I,B) origin. The rotation of the graphics object area about the page X_p-axis matches the rotation of the current text (I,B) coordinate system. For example, with a (90°,180°) text orientation, the object area rotation is 90°. The extents of the object area match the extents specified in the Margin Definition (X'7F') triplet on the BDM. The position of the graphics primitive in the (I,B) coordinate system therefore maps to the same position in the object area (X_{oa},Y_{oa}) coordinate system. This in turn is mapped to a graphics window whose upper left corner is at the graphics presentation space (GPS) origin and whose extents match those of the object area. The upper left corner of the graphics presentation space window is therefore also coincident with the current (I,B) origin. The mapping between graphics window and object area is *position and trim*.

For example, if the RCD specifies a (90°,180°) text orientation, the upper left corner of the graphics window is at the top-right corner of the page and graphics primitives in this object are rotated 90° with respect to the page X_p axis. The X-extent of the graphics window is the Y-extent of the graphics object area and the Y-extent of the graphics window is the X-extent of the graphics object area. The units of measure for the graphics presentation space and for the graphics object area are the same as those defined on the page (X_p,Y_p) presentation space in the PGD structured field of the Active Environment Group (AEG) for the Data Map.

Relative Baseline Positioning—LND Processing

Records, fields, and objects are positioned using the print position specified in bytes 2–5 of the LND structured field. These bytes normally specify an absolute offset from the origin of the current text (I,B) coordinate system. With relative baseline positioning, LND bytes 4–5 may be used to specify a baseline position *relative* to an established baseline position. This allows records, fields, and objects to be positioned (in the baseline direction) relative to a previous record, field, or object.

Relative baseline positioning is used when LND flag byte bit 13 is set to B'1'. The relative offset may be positive or negative and is measured using the current I,B coordinate system. Note that the origin of the current I,B coordinate system depends on the current text orientation. The baseline position used as a reference for the relative offset depends on whether the LND that specifies relative positioning is a base LND and on whether a page or subpage boundary was crossed since the last LND was used to print. The baseline position used as a reference for the relative offset is determined as follows:

- For base LNDs, offsets are defined relative to the last base LND processed, either by printing or by spacing. However, if a page or subpage boundary was crossed after the last base LND was processed, offsets are defined relative to the first LND for the page or subpage.
- For reuse LNDs other than base LNDs, the offset is defined relative to the last LND used to print.
- If the first LND of a Data Map specifies relative positioning, the offset is defined relative to the current text coordinate system origin (I=0,B=0), using the current text (I,B) coordinate system.
- If the first LND of a subpage specifies relative positioning, the offset is defined relative to the last print position, using the current text (I,B) coordinate system. Note that when skipping into a subpage, if the skipped-to LND specifies relative positioning, the relative offset is measured with respect to the first LND of the subpage, which may specify a relative position as well. This function allows a subpage to “float” relative to the last print position.

The following restriction applies to relative baseline positioning:

- The text orientation of an LND that specifies relative baseline positioning must be the same as the text orientation of the LND that defines the baseline position from which the relative offset is measured.

Note that if the line data processed with relative baseline positioning LNDs contains carriage controls that specify double or triple spacing, the presentation system must accumulate the relative offsets of the skipped LNDs in order to achieve the proper line spacing. If an LND that specifies absolute positioning is skipped, the position is reset to the absolute position and the relative offsets of any additional skipped LNDs are accumulated with respect to the absolute position. When a page boundary is crossed, printing resumes at the first LND.

Application Note: When relative baseline positioning is used, the PageDef generator cannot check for off-page errors, since the data normally determines, with skip-to-channel carriage controls, when the relative baseline LNDs are invoked. AFP print servers will generate a page break if the active Data Map is about to position data past the page's y-extent. This will not cause the generation of an error message. Note that the page's y-extent is specified in the PGD of the Data Map.

Skip-to-Channel Processing for Relative Baseline Positioning

When a skip-to-channel carriage control is received, the remainder of the LNDs are searched sequentially for a matching channel code until the end of the subpage is reached. If no matching channel code is found, and if the skip is not to channel 1, a search is made for a *relative* LND with matching channel code starting at the top of that subpage; if found, processing continues with this LND. If no relative LND with matching channel code is found in the subpage or if the skip is to channel 1, the search for *any* LND with matching channel code continues in the next sequential subpage. If the end of the Data Map is reached, a new page is started, and the Data Map is searched, starting at the beginning, for *any* LND with matching channel code.

Relative Baseline Positioning—RCD Processing

Relative baseline positioning can also be used when record-format line data is processed with RCDs.

Relative baseline positioning is used when flag byte bit 13 is set to B'1'. The baseline position used as a reference for the relative offset depends on whether the RCD that specifies relative positioning is a record RCD and is determined as follows:

- For record RCDs, offsets are defined relative to the last record RCD processed. However, if a page boundary was crossed after the last record RCD was processed, offsets are defined relative to the top margin.
- For field RCDs, the offset is defined relative to the last RCD used to print.

- If the first RCD of a Data Map specifies relative positioning, the offset is defined relative to the top margin.

The following restriction applies to relative baseline positioning:

- The text orientation of an RCD that specifies relative baseline positioning must be the same as the text orientation of the RCD that defines the baseline position from which the relative offset is measured.

Relative Baseline Positioning—XMD Processing

Relative baseline positioning can also be used when XML data is processed with XMDs.

Relative baseline positioning is used when flag byte bit 13 is set to B'1'. The baseline position used as a reference for the relative offset depends on whether the XMD that specifies relative positioning is an element XMD and is determined as follows:

- For element XMDs, offsets are defined relative to the last element XMD processed. However, if a page boundary was crossed after the last element XMD was processed, offsets are defined relative to the top margin.
- For field XMDs, the offset is defined relative to the last XMD used to print.
- If the first XMD of a Data Map specifies relative positioning, the offset is defined relative to the top margin.

The following restriction applies to relative baseline positioning:

- The text orientation of an XMD that specifies relative baseline positioning must be the same as the text orientation of the XMD that defines the baseline position from which the relative offset is measured.

Relative Inline Positioning—XMD Processing

Data and objects are positioned using the print position specified in IPos and BPos parameters of the XMD structured field. The IPos normally specifies an absolute offset from the origin of the current text (I,B) coordinate system. With relative inline positioning, the IPos parameter may be used to specify an inline position relative to an established inline position. This allows data and objects to be positioned (in the inline direction) relative to data placed previously on the page. If no data were placed on the page prior to the current data, the relative inline position is relative to the left margin. Note that the actual location of the left margin on a page is affected by the text orientation; see “Margin Definition (X'7F') Triplet” on page 73.

Relative inline positioning is used when XMD flag byte bit 12 is set to B'1'. The relative offset may be positive or negative and is measured using the current I,B coordinate system. Note that the origin of the current I,B coordinate system depends on the current text orientation.

The following restriction applies to relative inline positioning:

- The text orientation of an XMD that specifies relative inline positioning must be the same as the text orientation of the XMD that defines the inline position from which the relative offset is measured.

Note: Data must not exceed the boundaries of the page, which are defined in the Page Descriptor (PGD) structured field. If the new print position is outside these boundaries, printing of the page stops.

The Function of the Form Definition

A Form Definition is a MO:DCA print control object that is used to place pages on sheets and is always required for printing with presentation services programs. Form Definitions contain information about the physical environment in which the output is to be printed, such as the paper drawer to be used and whether printing is to be done in simplex or duplex mode. The Form Definition might also specify overlays to be used with the data. Two types of overlays may be specified in a Form Definition: medium overlays and PMC overlays. Medium overlays are positioned at the medium origin, while PMC overlays are positioned with respect to the page origin. Contrast these with overlays that are mapped in a Page Definition, which are invoked for a page using an Include Page Overlay (IPO) structured field and are called page overlays. The overlays themselves must be generated with a separate program designed to build overlay objects. The format for medium overlays and for PMC overlays (invoked in Form Definitions) is the same as the format for page overlays (invoked with an IPO structured field and mapped in Page Definitions).

Form Definitions are like Page Definitions in that only one Form Definition can be associated with a given print file and also in that each Form Definition includes one or more components. While the components of a Page Definition are called Data Maps or Page Formats, the components of a Form Definition are called Medium Maps or Copy Groups. An application program can switch between Medium Maps by using conditional processing, as in the example in Figure 12 on page 31. Control for presentation starts with the first Medium Map in the Form Definition. Control for presentation can be changed to a different Medium Map by using an Invoke Medium Map (IMM) structured field. If the Form Definition is used to present multiple documents in a print file, control for presentation is returned to the first Medium Map whenever a new document is encountered.

A file can be printed multiple times, each with a different Form Definition. The example in Figure 5 on page 16 can be modified to add Form Definition names in addition to Page Definition names.

Details on Form Definitions and overlay objects can be found in the *Mixed Object Document Content Architecture (MO:DCA) Reference*. A set of Form Definitions that address standard requirements is provided with presentation services program software, but users can also create customized Form Definitions.

Chapter 4. Mixed Documents: Adding MO:DCA Structured Fields to Line Data

Chapter 3, "Using a Page Definition to Print Data" describes how Page Definitions can be used to format traditional application line data without the need to make any application programming changes. Under certain circumstances, however, functions are needed that can only be accomplished by changing the application. These functions can be invoked by using one of a small set of MO:DCA structured fields, any of which can be intermixed with line data to obtain specific results. A document of this type, in which structured fields are intermixed with line data, is called a *mixed document*.

Note: MO:DCA structured fields cannot be combined with XML data.

MO:DCA structured fields cannot be interspersed with line data records indiscriminately. Data object structured fields and resource structured fields can appear only within their respective objects and resources, and only in the sequence allowed by the architecture. For example, the Map Coded Font (MCF) structured field is part of the Active Environment Group that in turn can appear in a presentation page, an overlay, or a Page Definition. However, it is not permitted to include an MCF between line-mode data records in an output file or to bracket line-mode records with Begin Page and End Page structured fields. Refer to "Page Definition Structure" on page 16 for the structure of the Page Definition object and refer to the *Mixed Object Document Content Architecture (MO:DCA) Reference* for the structure of other data and resource objects.

This chapter discusses how data and resource objects can be intermixed with line data and provides examples of structured fields that can be included individually with line-data records. These structured fields are:

- Invoke Data Map
- Invoke Medium Map
- Include Page Segment
- Include Page Overlay
- Include Object
- Presentation Text

Note: The No Operation (NOP) structured field may appear anywhere in a mixed document and thus is not listed in the structured field groupings.

This chapter contains coding examples for some of these structured fields. Chapter 5, "Structured Fields in a Page Definition and in Line Data" contains additional information on the format of these structured fields. See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for the formal definition of all MO:DCA structured fields.

The presence of structured fields in line data does not change the fact that the Page Definition is the controlling resource that determines how the data appears on the page. Structured fields other than those that change Data Maps or Medium Maps do not affect the placement of line-data records, nor do they affect the text orientation or font selection used to print line-data records. These characteristics of line-data records are defined in the Page Definition. Only when the application generates fully composed documents is a Page Definition not used.

Copyright © AFP Consortium 1994, 2018

X'5A' Carriage Control Character

When printing in a z/OS system environment, if MO:DCA structured fields are used either in a fully composed document or intermixed with line data, each MO:DCA structured field must be prefixed with a X'5A' character. The X'5A' appears in the first byte position and provides a signal to a presentation services program that the record is a structured field, not a data record.

The X'5A' character precedes the MO:DCA structured field and is not formally part of the structured field, so it is *not* counted in the structured field length value that immediately follows it. The examples in this chapter all contain a X'5A' character in the first byte position.

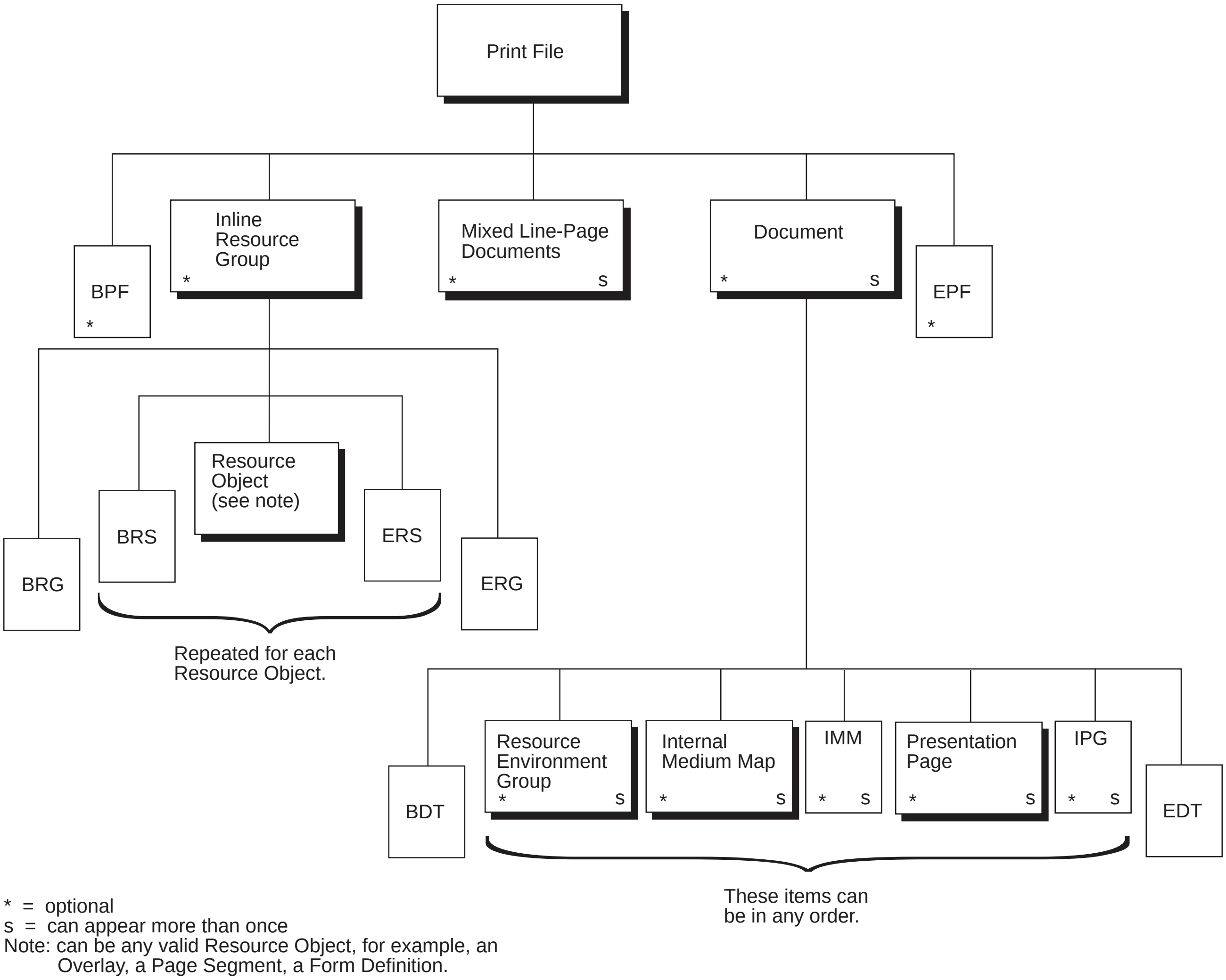
In a z/OS system environment, each MO:DCA structured field must occupy one record. The requirement to prefix MO:DCA structured fields with X'5A' means that all other records in the data set must begin with a carriage control character, even if it is only a “print-and-space-one-line” carriage control. Either ANSI or machine code carriage control can be used for these records.

In an AIX environment, the carriage control character is optional. The New Line control, also called Linefeed (X'25' in EBCDIC, X'0A' in ASCII), is used to determine end-of-record in AIX. The use of the Linefeed carriage control to determine end-of-record allows variable-length records to be easily created in AIX environments.

Print File Structure

An AFP print file consists of an optional inline resource group followed by one or more documents. Each document may, in turn, be preceded by an optional document index. All resources in an inline resource group must precede all other data in the print file. The group of resources is delimited by the Begin Resource Group (BRG) and End Resource Group (ERG) structured fields. Each resource object in the group is delimited by the Begin Resource (BRS) and End Resource (ERS) structured fields. If multiple fully composed documents are present in the print file, they must be delimited by Begin Document (BDT) and End Document (EDT) structured fields. Note that mixed line-page documents and composed documents can occur in any order following the inline resource group. Figure 15 on page 41 shows the structure of an AFP print file.

Figure 15. Structure of a Print File


Print File
BPF *
Inline Resource Group *
Mixed Line-Page Documents * s
Document * s
EPF *
BRG
BRS
Resource Object (see note)
ERS
ERG
Repeated for each Resource Object.
BDT
Resource Environment Group * s
Internal Medium Map * s
IMM * s
Presentation Page * s
IPG * s
EDT
These items can be in any order.
* = optional
s = can appear more than once
Note: can be any valid Resource Object, for example, an Overlay, a Page Segment, a Form Definition.

Notes:

1. The BPF/EPF structured fields are optional as a pair; if one is specified, the other must be specified as well.
2. The mixed line-page documents and composed documents can occur in any order following the inline resource group.
3. Each AFP (MO:DCA) document may optionally be preceded by a single document index that is implicitly tied to the document and that indexes the document. For the formal definition of the MO:DCA document index, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.
4. An AFP (MO:DCA) document may contain Link Logical Element (LLE) structured fields following the BDT. It may also group presentation pages into named page groups. MO:DCA page groups may in turn contain Tag Logical Element (TLE) structured fields following the BNG. These structures do not affect the presentation of the document. For the formal definition of these structures, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.
5. If a Medium Map is included internal (inline) to the document, it is activated by immediately following it with an IMM that explicitly invokes it, otherwise the internal Medium Map is ignored. An IMM that does not follow an internal Medium Map may not invoke an internal Medium Map elsewhere in the document and is assumed to reference a Medium Map in the current Form Definition.

Mixed Documents

The objects that comprise an AFP print file are as follows:

Inline Resource Group

Contains one or more resource objects to be associated with printing this file. See "Inline Resource Group Structure" on page 43 for a detailed description of the structure of the resource group and the objects it can contain.

Note: In the MO:DCA architecture, these resource groups are called external resource groups because they occur outside a document.

The Inline Resource Group is an optional component of the Print File. If no Inline Resource Group is defined, the resources stored in the AFP resource library of the system are used. (In MVS/ESA™ with USERLIB support, resources might be stored in private libraries that are used at print time for individual data sets. Up to eight private libraries may be used with a single data set. The libraries are named in the USERLIB parameter of the OUTPUT JCL statement.)

The scope of an inline resource group is the print file. Once the last document in the print file has been processed, the resources in the resource group are no longer available to the presentation system for use with another print file.

Documents

The print file may contain one or more documents to be printed. These may be fully composed-page documents, line-mode documents, or mixed-mode documents, in any order. If multiple composed-page documents appear, each one must be delimited by a BDT and an EDT structured field. For the complete definitions of document structure, see Appendix A, "Document and Resource Object Diagrams", on page 169.

Finishing Operations for a Print File

A Form Definition may be used to specify finishing operations to be applied to the documents in a print file. The scope of the finishing operations as well as the type of operation is specified with a Medium Finishing Control (MFC) structured field in the Document Environment Group (DEG) of the Form Definition. For a definition of the finishing operations and parameters that may be specified, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*. The following rules specify how the scope of the finishing operations applies to a print file when the file contains line-data and mixed-data documents, with or without BDT/EDT, as well as composed documents.

- If the MFC specifies print-file level finishing, all media in the print file is collected for finishing in a print-file level media collection and the finishing operations are applied to the complete collection; that is, the complete print file.
- If the MFC specifies document-level finishing and selects all documents, the print file is processed as a set of documents as follows:
  - Any document bounded by BDT/EDT is processed as a single document regardless of whether the data between BDT/EDT is line data, mixed data, or composed data.
  - Line data and mixed data that is not bounded explicitly by BDT/EDT is processed as an implied document with implied BDT/EDT. When such data follows the resource group or an EDT, a BDT is implied and the implied document lasts until a BDT is encountered or until the end of the print file is reached. In either case, the implied document is terminated with an implied EDT.

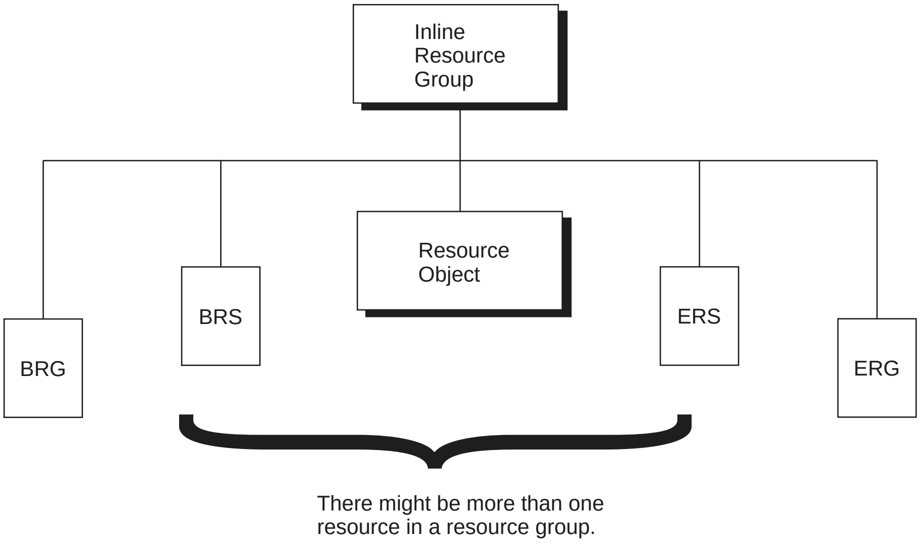
  The media in each document, whether implied or explicit, is collected for finishing in a document-level media collection and the finishing operations are applied to each collection, that is each document, individually. Note that, in this case, the same finishing operations are applied to each document.
- If the MFC specifies document-level finishing and selects a single document, the print file is processed as a set of documents in the same manner as when all documents are selected. The offset of the selected document is calculated by counting all documents, whether implied or explicit, and the selected document might itself be an implied document. The media in the selected document are collected for finishing and the

finishing operations are applied to the single collection; that is, the single document. If the same document is selected multiple times, finishing operations are applied in the order specified. Note that, using this type of MFC, unique finishing operations may be specified for each document in the print file.

Inline Resource Group Structure

A resource group begins with the Begin Resource Group (BRG) structured field and ends with the End Resource Group (ERG) structured field. Inline resources are included in the inline resource group and can be referred to by name within the print file. They override objects of the same name stored in resource libraries accessed by the print server. Each individual resource begins with the Begin Resource (BRS) structured field and ends with the End Resource (ERS) structured field. When a resource object is stored in a library, the BRG, BRS, ERG, and ERS structured fields are not present. When using AFP with a z/OS system, all structured fields of resource objects included in inline resource groups must be preceded by the X'5A' character. Figure 16 shows the structure of an inline resource group.

Figure 16. Structure of an Inline Resource Group



The structured fields and objects in an inline resource group are as follows. (Chapter 5, “Structured Fields in a Page Definition and in Line Data” describes the structured fields.)

BRG (Begin Resource Group)
Begins an inline resource group in the Print File.

BRS (Begin Resource)
Begins a resource object, specifies the resource type, and specifies the name used to select the object for printing.

Resource Object
A resource object can be one of the following:

- A page segment
- An overlay
- A data object
- An object container
- A document
- A Form Definition
- A Page Definition
- A font object (a code page, a font character set, or a coded font)

See the description of the BRS structured field in the *Mixed Object Document Content Architecture (MO:DCA) Reference* for the hexadecimal codes used to identify each type of resource object.

ERS (End Resource)
Ends the resource object. Any name specified in the ERS must match the name specified in the BRS.

ERG (End Resource Group)
Ends the inline resource group. Any name specified in the ERG must match the name specified in the BRG.

Note: Not all presentation services programs support all resource objects in a Resource Group.

Programming Considerations for Inline Resources

Because most resource objects consist of variable-length records, any print file that includes these resources inline must be in variable-length-record format and must use data records beginning with a carriage control byte.

Invoke Data Map

The Invoke Data Map (IDM) structured field selects a new Data Map for printing line data and ends the current line-format page.

Note: When using machine carriage control characters, care must be taken to prevent a blank page from being printed at the start of a document. If the application inserts IDM structured fields following records that have a “skip to channel *nn* immediate” carriage control (X'8B') without making an exception for the start of the document, a blank page will be generated. When the first line data record contains a skip immediate carriage control, a line-format page is started even though there is no data to be printed. When the IDM follows the initial skip immediate carriage control at the start of the document to be printed, the IDM ends the current page, causing a blank page to be printed. When the skip immediate carriage control is used later in the document to end the page and it is followed by the IDM structured field, a blank page does not occur since the skip immediate carriage control has already ended the current line-format page.

- For traditional line data, processing begins with the first Line Descriptor (LND) structured field of the invoked Data Map for the next line-format page.
- For record-format line data, processing begins with the first Record Descriptor (RCD) structured field that matches the record ID of the first record processed following the IDM.

The IDM structured field can be used to change formatting based on some change in the application data, such as the start of output for a different department or branch office.

The IDM structured field always contains sixteen bytes of information. The Data Map name in the data portion of this structured field must be eight bytes long. If the name of the actual Data Map to be invoked is shorter than eight bytes, trailing blanks must be added.

Sample IDM Structured Field

The Invoke Data Map structured field shown in Figure 17 causes a presentation services program to select Data Map SUMMARY.

The IDM is 16 (X'10') bytes long and has the structured field identifier X'D3ABCA'. In the example, the flag byte is set to X'00' and bytes 6 and 7 contain a sequence number of X'0000'. It is not necessary to number MO:DCA structured fields sequentially or even to place a meaningful value in the sequence number field. However, for some errors detected by presentation services programs, the sequence number of the structured field in error is printed as part of the error information in the PIMSG data set. This field is reserved in MO:DCA data streams and should be set to zero.

When a presentation services program processes the IDM structured field, the current page is ended. The next record read by the presentation services program begins on a new page and the information contained in Data Map SUMMARY is used to format subsequent data lines. Use of this structured field assumes the currently active Page Definition contains a Data Map with the name SUMMARY in its Begin Data Map structured field. If no such Data Map exists, an error is generated.

Figure 17. Sample Invoke Data Map Structured Field

X'5A'	X'0010'	X'D3ABCA'	X'00'	X'0000'	X'E2E4D4D4C1D9E840'

Invoke Medium Map

The Invoke Medium Map (IMM) structured field is similar to the IDM structured field except that it causes a presentation services program to select a new Medium Map, or Copy Group, in the current Form Definition at the point where the IMM structured field appears in the print file. The presentation services program ends printing on the current sheet when an IMM is encountered. Note that if the Medium Map specifies the N-up function, the IMM might cause the presentation services program to end printing on the current N-up partition instead of on the current sheet.

The IMM structured field can appear in line-mode, mixed-mode, or fully composed documents. For line-mode or mixed-mode data, processing resumes with the first Line Descriptor (LND) structured field in the Data Map that is active for the next line-format page. When the Data Map contains RCDs, processing resumes with the first RCD whose Record ID matches the current data record. The IMM structured field is sixteen bytes long and must be coded as shown below.

The IMM structured field can be written by the application when some physical control of the output is required. By using the IMM, the application can offset pages in the data from the medium origin, select paper from the primary or alternate bin, or change between simplex and duplex printing, simply by selecting a Medium Map that contains the desired function.

The functions provided by the IDM and IMM structured fields are the same as those provided by changing Data Maps and Medium Maps with conditional processing in a Page Definition. It is possible to use conditional processing to make the Data Map and Medium Map change without modifying the application to add the IDM and IMM structured fields.

Note that at the beginning of a new composed document and at the beginning of a new set of line-data records, control for presentation is returned to the first Medium Map in the Form Definition. This is shown in Figure 18 on page 46.

Figure 18. Returning Control to First Medium Map in Form Definition

```
Form Definition
   Medium Map M1
   Medium Map M2

Line data records       <presentation controlled by M1>
  ⋮
IMM, Medium Map M2
Line data records       <presentation controlled by M2>
  ⋮
BDT                     <presentation control reverts to M1>
  Composed Pages
  ⋮
  IMM, Medium Map M2    <presentation controlled by M2>
  ⋮
  Composed Pages
  ⋮
EDT
Line data records       <presentation control reverts to M1>
  ⋮
```

Sample IMM Structured Field

The Invoke Medium Map structured field shown in Figure 19 causes the presentation services program to select Medium Map `BIN2`. (Note that `BIN2` contains four trailing blanks to fill out the eight-byte data field.) When a presentation services program processes this structured field, the current page is ended. The next record read by the presentation services program is placed on a new sheet and the information contained in medium map `BIN2` is used. Note that if the Medium Map specifies the N-up function, the next record may be placed on a new partition of the same sheet. If the currently active Form Definition does not contain a Medium Map with that name in the Begin Medium Map structured field, an error is generated.

Figure 19. Sample Invoke Medium Map Structured Field

X'5A'	X'0010'	X'D3ABCC'	X'00'	X'0000'	X'C2C9D5F240404040'

Using Structured Fields to Skip to a New Page or Sheet

Chapter 3, "Using a Page Definition to Print Data" described the use of conditional processing in a Page Definition to perform a skip-to-new-page or skip-to-new-sheet operation based on a change in the value of a control field in an application data stream. The conditional processing function was added to the Page Definition to provide another way of producing the same output as by embedding IDM or IMM structured fields in a line-data file to force a new page or sheet. When an IDM or IMM structured field appears in an application data stream, the presentation services program ends the current page and resumes printing at the start of a new page, using the first Line Descriptor in the current Data Map. When the Data Map contains RCDs, printing resumes at the start of a new page using the first RCD whose Record ID matches the current data record.

The data stream shown in Figure 20 on page 47 provides the same result as the Page Definition shown in Figure 13 on page 32 and the data stream shown in Figure 21 on page 47 provides the same result as the Page Definition shown in Figure 14 on page 32.

Figure 20. Using an IDM Structured Field to Skip to a New Page

```
Line data records (with carriage control)
  ⋮
X'5A0010D3ABCA000001D5C5E6D7C7404040'
More line data records
  ⋮
```

Figure 21. Using an IMM Structured Field to Skip to a New Sheet

```
Line data records (with carriage control)
  ⋮
X'5A0010D3ABCC000001D5C5E6C6D4404040'
More line data records
  ⋮
```

The name of the Data Map invoked in Figure 20 is `NEWPG`. This is the name on the PAGEFORMAT statement in the PPFA example in Figure 13 on page 32. Re-invoking the same Data Map causes a skip to a new page. It is not necessary to have multiple Data Maps in the Page Definition to achieve this result. Consequently, standard Page Definitions supplied with the print services software can be used with this method.

The same is true of skipping to a new physical sheet. Figure 21 invokes a Medium Map named `NEWFM`. Even if `NEWFM` is the current and only Medium Map in the Form Definition, the presence of this structured field causes a skip to a new sheet of paper, or, in the case of N-up presentation, possibly a skip to a new N-up partition.

IMM Structured Fields to Insert a Blank Sheet

Occasionally an application requires that a blank sheet appear between groups of output within a single data set. This blank sheet might be selected from different-color paper loaded in the alternate bin, or it might just be another sheet from the primary bin. The blank sheet is generated by using a Form Definition that specifies the constant data function, which allows a sheet to be produced without any variable data on it. To generate the blank sheet, code two consecutive IMM structured fields, as shown in Figure 22.

Figure 22. Using Two IMM Structured Fields to Force a Blank Sheet

```
Line data records (with carriage control)
  ⋮
X'5A0010D3ABCC000000C1D3E3C2C9D54040'
X'5A0010D3ABCC000000D7D9C9C2C9D54040'
More line data records (with carriage control)
  ⋮
```

This example assumes a Form Definition with two Medium Maps, as could be built using the PPFA code shown in Figure 23 on page 48. The first Medium Map coded in the example will be used for the initial pages. They will contain user data (the CONSTANT parameter does not appear in this Medium Map) and are printed on paper selected from the primary bin. When the point is reached where a blank sheet is to be inserted, the application writes out an Invoke Medium Map that selects the second Medium Map. This Medium Map selects a sheet of paper from the alternate bin. No user data is placed on the pages coming from the alternate bin, because CONSTANT FRONT and DUPLEX NO are coded. If the output were to be printed in duplex, CONSTANT BOTH and DUPLEX YES can be coded instead.

Immediately following the IMM structured field to select the second Medium Map (`ALTBIN`) is a second IMM to return to the original Medium Map (`PRIBIN`) for the next portion of the data. This set of two consecutive IMM structured fields can be included in the output data stream as often as necessary.

Figure 23. Form Definition With Two IMMs to Force a Blank Sheet

```
FORMDEF BLANKT
 OFFSET 0 0
 REPLACE YES;
COPYGROUP PRIBIN
 DUPLEX NO BIN 1;
COPYGROUP ALTBIN
 CONSTANT FRONT
 DUPLEX NO BIN 2;
```

Variable-Length and Fixed-Length Records

MO:DCA structured fields are variable in length so their lengths can differ. Line data records intermixed with MO:DCA structured fields might also have different lengths. Fully composed MO:DCA documents may consist of records up to 32K bytes long. However, variable-length data is not always desirable. Programming requirements might make it preferable to use fixed-length records in some circumstances. A presentation services program can process a mixed document of fixed-length records even though some of the records contain structured fields with significant information that is much shorter than the data records to be printed. So long as the information in the length portion of the structured field is correct and the structured field is padded with blanks to the length of the other records in the data set, no errors are generated. The structured fields shown in Figure 24 are all considered valid by the presentation services program in a z/OS system environment. The third form, however, might not be supported in a multi-system environment.

Figure 24. Three Versions of the Invoke Data Map Structured Field

X'0010'	X'D3ABCA'	X'00'	X'0000'	PFORMAT1 (Data Map [Page Format] name—8 bytes EBCDIC)

X'0050'	X'D3ABCA' (Identifier)	X'08' (Flag byte)	X'0000'	PFORMAT1 (Data Map [Page Format] name—8 bytes EBCDIC)	63 bytes of X'00' followed by one byte of X'40'

X'0010'	X'D3ABCA' (Identifier)	X'00' (Flag byte)	X'0000'	PFORMAT1 (Data Map [Page Format] name—8 bytes EBCDIC)	64 bytes of any information to fill the record out to 80 bytes

The first structured field at the top of Figure 24 is the most common form of Invoke Data Map. The IDM structured field is 16 bytes (X'10') long, so the value X'10' appears in the length field of the introducer. Next is the X'D3ABCA' identifier for Invoke Data Map. The flag byte is zero. The syntax rules for Invoke Data Map indicate that the eight-byte name of the requested Data Map be coded as the data portion of the structured field. This is the rightmost information in the figure.

The second structured field in Figure 24 is 80 bytes long, but here the formal MO:DCA conventions for using padding bytes have been followed. In this example, the flag byte is coded as X'08', which signals that padding bytes appear in the structured field. The padding bytes follow the variable data for the IDM structured field and the final padding byte is coded as X'40' to signal that 64 padding bytes are present. The length field has been changed from 16 (X'10') to 80 (X'50') to reflect the increased length of the structured field.

The third structured field in Figure 24 is identical to the first, except that the actual MO:DCA data appears as the first 16 bytes in an 80-byte record. This format allows the IDM structured field to be included in a data set of fixed-length 80-byte records and no errors would result in a z/OS system environment.

Of course, fixed-length records that are longer than the number of bytes actually used to contain the MO:DCA structured field information will result in a data set that is larger than one containing variable-length records, each one no longer than necessary. This might be a consideration if the resulting data set is to be sent across a network.

Position and Orientation of Objects

Two coordinate systems are used to position and rotate objects in line data: the page (X_p,Y_p) coordinate system and the text (I,B) coordinate system. The page coordinate system is based on the fourth quadrant of a standard Cartesian coordinate system with the origin in the top-left corner, the X axis increasing from left to right, and the Y axis increasing from top to bottom. The text (I,B) coordinate system is defined, relative to the page coordinate system, by the text orientation as follows:

Text Orientation	(I,B) Coordinate System
0°,90°	Origin at top-left corner, I increases left to right, B increases top to bottom.
90°,180°	Origin at top-right corner, I increases top to bottom, B increases right to left.
180°,270°	Origin at bottom-right corner, I increases right to left, B increases bottom to top.
270°,0°	Origin at bottom-left corner, I increases bottom to top, B increases left to right.

The coordinate system used depends on the object and how it is included in line data. Table 10 summarizes how objects are positioned and rotated in line data. The table also summarizes how objects are positioned and rotated in MO:DCA data that has been transformed from line data, using the Line Data Object Position Migration (X'27') triplet to capture the text orientation that was active when the line data was presented with a Page Definition. More details on how objects are positioned and rotated are given in the sections that follow the table.

Table 10. Position and Rotation of Objects in Line Data and MO:DCA Data

OBJECTS IN LINE DATA	OBJECTS WITH X'27' TRIPLET IN MO:DCA DATA TRANSFORMED FROM LINE DATA
Page Segment Object	
Page Segment Origin	
(XpsOset,YpsOset) in IPS specify an offset from the current text coordinate system origin (I=0,B=0). The offset is measured using the current text (I,B) coordinate system.	(XpsOset,YpsOset) in IPS specify an offset from the page origin (Xp=0,Yp=0). The offset is measured using the page (X_p,Y_p) coordinate system. The offset was adjusted to include the LND or RCD position.
IM—Image Object in Page Segment	
IM—Image Object Origin	
(XoaOset,YoaOset) in IOC specify an offset from the page segment origin. The offset is measured using the current text (I,B) coordinate system.	(XoaOset,YoaOset) in IOC specify an offset from the page segment origin. The offset is measured using the temporary (X,Y) coordinate system.
IM—Image Object Rotation	
(XoaOrent,YoaOrent) in IOC specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.	(XoaOrent,YoaOrent) in IOC specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.
IM—Image Cell Origin	
(XCOset,YCOset) in ICP specify an offset from the image object origin. The offset is measured using the current text (I,B) coordinate system.	(XCOset,YCOset) in ICP specify an offset from the image object origin. The offset is measured using the temporary (X,Y) coordinate system.

Table 10 Position and Rotation of Objects in Line Data and MO:DCA Data (cont'd.)

OBJECTS IN LINE DATA	OBJECTS WITH X'27' TRIPLET IN MO:DCA DATA TRANSFORMED FROM LINE DATA
OCA Object in Page Segment	
OCA Object Origin—OBP Byte 23=X'00'	
(XoaOset,YoaOset) in OBP specify an offset from the page segment origin. The offset is measured using the current text (I,B) coordinate system.	(XoaOset,YoaOset) in OBP specify an offset from the page segment origin. The offset is measured using the temporary (X,Y) coordinate system.
OCA Object Origin—OBP Byte 23=X'01'	
(XoaOset,YoaOset) in OBP specify an offset from the page origin (Xp=0,Yp=0). The offset is measured using the page (X_p,Y_p) coordinate system.	(XoaOset,YoaOset) in OBP specify an offset from the page origin (Xp=0,Yp=0). The offset is measured using the page (X_p,Y_p) coordinate system.
OCA Object Rotation—OBP Byte 23=X'00'	
(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the current text (I,B) coordinate system I-axis.	(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the temporary (X,Y) coordinate system X-axis.
OCA Object Rotation—OBP Byte 23=X'01'	
(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.	(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.
Stand-alone IM—Image Object	
IM—Image Object Origin	
(XoaOset,YoaOset) in IOC specify an offset from the current LND or RCD position. The offset is measured using the current text (I,B) coordinate system.	(XoaOset,YoaOset) in IOC specify an offset from the temporary coordinate system (X=0,Y=0) origin. The offset is measured using the temporary (X,Y) coordinate system. The offset was adjusted to include the LND or RCD position.
IM—Image Object Rotation	
(XoaOrent,YoaOrent) in IOC specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.	(XoaOrent,YoaOrent) in IOC specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.
IM—Image Cell Origin	
(XCOset,YCOset) in ICP specify an offset from the image object origin. The offset is measured using the current text (I,B) coordinate system.	(XCOset,YCOset) in ICP specify an offset from the image object origin. The offset is measured using the temporary (X,Y) coordinate system.
Stand-alone OCA Object	
OCA Object Origin—OBP Byte 23=X'00'	
(XoaOset,YoaOset) in OBP specify an offset from current LND or RCD position. The offset is measured using the current text (I,B) coordinate system.	(XoaOset,YoaOset) in OBP specify an offset from the temporary coordinate system (X=0,Y=0) origin. The offset is measured using the temporary (X,Y) coordinate system. The offset was adjusted to include the LND or RCD position.
OCA Object Origin—OBP Byte 23=X'01'	
(XoaOset,YoaOset) in OBP specify an offset from the page origin (Xp=0,Yp=0). The offset is measured using the page (X_p,Y_p) coordinate system.	(XoaOset,YoaOset) in OBP specify an offset from the page origin (Xp=0,Yp=0). The offset is measured using the page (X_p,Y_p) coordinate system.

Table 10 Position and Rotation of Objects in Line Data and MO:DCA Data (cont'd.)

OBJECTS IN LINE DATA	OBJECTS WITH X'27' TRIPLET IN MO:DCA DATA TRANSFORMED FROM LINE DATA
OCA Object Rotation—OBP Byte 23=X'00'	
(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the current text (I,B) coordinate system I-axis.	(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the temporary (X,Y) coordinate system X-axis.
OCA Object Rotation—OBP Byte 23=X'01'	
(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.	(XoaOrent,YoaOrent) in OBP specify a rotation that is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.

Positioning With Respect to Current Descriptor

When objects are included in line data, they occur between line-data records and can be positioned with respect to the inline/baseline position specified by the LNDs or RCDs used to process the records. More precisely, an included object can be positioned with respect to the current LND, or current RCD. This is also sometimes referred to as the *current line position*, which is defined as follows:

Current LND Position

If the line-data records use ANSI carriage controls, spacing or skipping is performed first and printing of the record is performed last, therefore the current LND is the LND used to process the last record. If the line-data records use machine carriage controls, printing of the record is performed first and spacing or skipping is performed last. In this case, the current LND is the LND that is spaced to or skipped to, that is, it is the LND that will be used to process the next record. Additionally, if the record is processed as a set of fields using a reuse chain, the current LND is the base LND, that is, the LND that is at the head of the reuse chain. If the current LND does not generate a position, the LND used is the last LND that did generate a position.

Current RCD Position

Because carriage controls are ignored in record-format line data, the current RCD is always the last record RCD that was used to process a data record.

Include Page Segment

The Include Page Segment (IPS) structured field is used to place a page segment resource anywhere on the page. It contains the full eight-character name of the page segment (with trailing blanks if necessary) and the position of the page segment, often referred to as the *page segment origin*. The page segment might be mapped in a Map Page Segment (MPS) structured field in the Active Environment Group (AEG) of the current Data Map, in which case the page segment is downloaded to the printer and may be used multiple times. If it is not mapped, the page segment data is loaded as part of the page.

Objects within the page segment might be positioned with respect to the page segment origin. The page segment inherits the Active Environment Group definition of the including page.

AFP print servers initialize the following PTOCA control sequences as shown prior to processing a text object in an AFP page segment:

Control Sequence	Value
Set Baseline Increment	6 lines per inch
Set Inline Margin	0
Set Intercharacter Adjustment	0
Set Text Color	X'FFFF' (printer default color)
Set Text Orientation	0°,90°

The initial print position for text in the page segment is the reference point defined on the including page or overlay coordinate system by the IPS, that is, the page segment origin.

Positioning of Page Segments

Special care must be taken when including page segments in line data to ensure that the objects in the page segment are positioned and oriented properly.

Location of Page Segment Origin

The page segment origin is located on the page as follows:

- If one of the IPS offsets is specified as X'FFFFFF', the page segment origin along that axis is located at the position specified in the current LND or RCD.
- If the IPS offset is not X'FFFFFF', the page segment origin is located at the IPS offset measured with respect to the current text coordinate system origin (I=0,B=0), using the current text (I,B) coordinate system. For example, if the text orientation is (90°,180°), the page segment offsets are measured from the top-right corner of the page, with the I-axis running from top to bottom and the B-axis running from right to left.
- If the page segment is included with a Resource Object Include (X'6C') triplet on the LND or RCD, the page segment origin is located at the specified offset measured with respect to the position specified in the current LND or RCD, using the current text (I,B) coordinate system.

In summary, the origin of a page segment in line data is always positioned using the text (I,B) coordinate system specified in the current LND or RCD.

Position and Orientation of IM Image Objects in a Page Segment

The image object area offset, as specified in the IOC structured field is measured with respect to the page segment origin, using the text (I,B) coordinate system specified in the current LND or RCD. If the image is celled, the Image Cell Position (ICP) structured field specifies an offset from the image object origin that is measured using the current text (I,B) coordinate system.

The rotation of the IM image is specified in the IOC and is measured with respect to the page coordinate system X_p-axis (origin is top-left corner of page).

Note: For page segments in MO:DCA data, if the IM image is complex (celled), it is recommended that the rotation be set to (0°,90°). For page segments in mixed data, the rotation should be set to match the current text orientation.

Position and Orientation of Image, Graphics, and Bar Code Objects in a Page Segment

If the Object Area Position (OBP) structured field specifies byte 23 (RefCSys) = X'00' (current), the object area offset is measured with respect to the page segment origin, using the text (I,B) coordinate system specified in the current LND or RCD. The object area rotation is measured with respect to the I-axis of the current text (I,B) coordinate system.

If OBP byte 23 = X'01' (page or overlay), the object area offset is measured with respect to the page origin (top-left corner of page) using the page coordinate system. The object area rotation is measured with respect to the page coordinate system X_p-axis (origin is top-left corner of page).

Note: When line data that includes an IPS structured field is transformed into a MO:DCA document by an AFP application program, the text orientation that was set when the page segment and its objects were positioned must be captured and retained in order to properly position the page segment on the MO:DCA page. This can be done using a Line Data Object Position Migration (X'27') triplet on the IPS structured field in the MO:DCA document. For a description of this triplet, see the description of "Include Page Segment (IPS)" on page 95.

Sample IPS Structured Field

Figure 25 contains a sample IPS structured field. This example places the segment `SIGNAT` at the current print position. If the name of the segment were `S1SIGNAT`, then all eight characters would have to be coded in the IPS structured field.

See the programming tip below for information on how the current print position is affected by the IPS.

Figure 25. Include Page Segment Structured Field

X'5A'	X'0016'	X'D3AF5F'	X'00'	X'0000'	X'E2C9C7D5C1E34040'	X'FFFFFF'	X'FFFFFF'

Programming Tip

The current line position is unchanged after the page segment is printed. Additional logic might be needed in the application to place subsequent print lines so that they do not overprint the page segment.

Include Page Overlay

The Include Page Overlay (IPO) structured field functions in a manner similar to Include Page Segment. The IPO structured field specifies the full name of the overlay (any O1 prefix in the overlay name must be included) and the position of the overlay origin. The IPO references an overlay resource that is to be positioned on the page.

The overlay name must appear in the Map Page Overlay structured field of the Active Environment Group of the Data Map currently in effect. The overlay contains its own Active Environment Group definition that specifies the coordinate system for positioning and rotating objects, the size of the overlay, and the names of any fonts used in it. Considerations for the current line position are the same as those discussed in the previous programming tip. The current line position is unchanged after the overlay has been placed.

Note: The IBM 3800 printer does not support the IPO function.

Positioning Overlays

Because overlays define their own coordinate system and environment, the rules for positioning an overlay and its objects are somewhat different from those for positioning a page segment and its objects.

Location of Overlay Origin

The overlay origin is located as follows:

- If the IPO offset along either the page X_p-axis or the page Y_p-axis is specified as X'FFFFFF', the overlay origin along that same axis is located by translating the current LND or RCD (I,B) position to an offset along that X_p or Y_p axis.
- If the IPO offset is not X'FFFFFF', the overlay origin is positioned at the specified (X_p,Y_p) offset measured with respect to the page origin (top-left corner of page), using the page coordinate system.
- If the overlay is included with a Resource Object Include (X'6C') triplet on the LND or RCD, the overlay origin is located at the specified offset measured with respect to the position specified in the current LND or RCD, using the current text (I,B) coordinate system.

Orientation of Overlay

If the overlay is included either with an IPO or with a Resource Object Include (X'6C') triplet on the LND or RCD, the overlay rotation may be specified as 0°, 90°, 180°, or 270°, and is measured with respect to the page coordinate system X_p axis (origin is top-left corner of page). However, the 90°, 180°, and 270° rotations of a page overlay are not supported in all AFP environments. Consult the product documentation to see which rotations are supported. Note that the MO:DCA IS/1 and IS/2 interchange sets only support 0° rotation of a page overlay.

Position and Orientation of IM Image Object in an Overlay

The image object area offset, as specified in the IOC structured field, is measured using the overlay coordinate system (origin is top-left corner of overlay).

The rotation of the IM image is specified in the IOC and is measured with respect to the overlay coordinate system X-axis (origin is top-left corner of overlay).

Note: If the IM image is complex (celled), AFP print servers require the rotation set to 0°,90°.

Position and Orientation of IO Image, Graphics, and Bar Code Objects in an Overlay

If the Object Area Position (OBP) structured field specifies byte 23 (RefCSys) = X'00' (current) or X'01' (page or overlay), the object area offset is measured with respect to the overlay origin (top-left corner of overlay) using the overlay coordinate system.

The rotation of the OCA object is specified and measured using the overlay coordinate system X-axis (origin is top-left corner of overlay).

Sample IPO Structured Field

A sample IPO structured field appears in Figure 26. It places overlay `O1SIGNAT` at the current print position on the page.

Figure 26. Include Page Overlay Structured Field

X'5A'	X'0016'	X'D3AFD8'	X'00'	X'0000'	X'D6F1E2C9C7D5C1E3'	X'FFFFFF'	X'FFFFFF'

Include Object

The Include Object (IOB) structured field references an object that is to be positioned on the page. In general, the IOB may be used to include two classes of objects:

- OCA objects (BCOCA, GOCA, IOCA, PTOCA) that specify an Object Environment Group (OEG) or MO:DCA page segments that contain such objects
- Non-OCA paginated presentation objects, such as TIFF images, that are supported by the presentation system

The current AFP support for the IOB in line data is limited to the first class, OCA objects. When referencing an OCA object, the IOB may be used to override position, size, orientation, mapping, and default color parameters that are specified in the OEG. When referencing a non-OCA object, the IOB is used to specify the position, size, orientation, and mapping parameters for the object.

The RefCSys parameter in the IOB is used to select the coordinate system for positioning and rotating the object area into which the object is mapped:

Value Description

X'00' The object area offset in the IOB is measured with respect to the current LND or RCD position, using the current text (I,B) coordinate system. The object area rotation in the IOB is measured with respect to the current text (I,B) coordinate system I-axis.

X'01' The object area offset in the IOB is measured with respect to the page origin (X_p=0,Y_p=0), using the page (X_p,Y_p) coordinate system. The object area rotation in the IOB is measured with respect to page (X_p,Y_p) coordinate system X_p-axis.

Including Data Objects Directly in Line Data

Previously it was described how complete AFP resources can be included in the resource group of a print file rather than having to be stored in an external resource library. This is one approach that can be used with applications where many different resources must be included in the print stream and where it might not be feasible to store these resources externally to the application. However, another approach is possible for applications that require large numbers of graphics, images, or bar codes.

One example of such an application is label printing, where many different labels are printed in multiple-up format, each one requiring a unique bar code. Another example is a financial statement application that includes a chart of specific investment performance for each customer. The programming logic required for applications such as these is simpler if each bar code or page segment can be included in the output at the same point as the other data for a given label or statement. Grouping all resources (which can number in the thousands) in an external library or in a resource group at the beginning of the print file might not be practical. In addition, it might be preferable to keep the bar codes, images, or graphics as part of the actual line data for archive purposes. Finally, including them directly in the line data can eliminate the problem of devising unique names for thousands of objects that change each time the program is run.

Graphics, images, and bar codes included with other print data in this manner are not true inline resources, because they do not follow the rules for inline resources described previously. When structured fields that make up graphics, images, or bar codes are included directly in the line data, they provide yet another example of an AFP mixed-mode document.

Including Bar Code, Graphics, IO Image, and Presentation Text Objects with OEG

Objects that include an Object Environment Group (BCOCA, GOCA, IOCA, and PTOCA objects) can be included directly in a mixed-mode document intermixed with line data so long as the following rules are observed:

- The reference coordinate system (byte 23 of the data field of the Object Area Position [OBP] structured field) must be coded to provide the desired position and rotation of the object on the page:
  - If OBP byte 23 (RefCSys) = X'00' (current), the object area offset is measured with respect to the position specified in the current LND or RCD, using the current text (I,B) coordinate system. The object area rotation is measured with respect to the I-axis of the current text (I,B) coordinate system.
  - If OBP byte 23 (RefCSys) = X'01' (page or overlay), the object area offset is measured with respect to the page origin, using the page coordinate system (origin is top-left corner of page). The object area rotation is specified in the OBP and is measured with respect to the page coordinate system X_p-axis (origin is top-left corner of page).
- If the image or graphic has been built as a page segment, delete the Begin Page Segment and End Page Segment structured fields from the object. The remaining structured fields can be placed in the print stream at the point where the image or graphic should appear.

Including IM Image Objects

Page segments containing IM image data do not have an Object Environment Group, so somewhat different considerations apply to them. Between the BPS and EPS structured fields are the records that provide positioning information for the bits that define the image and the actual bits themselves in uncompressed form.

Just as for BCOCA, GOCA, IOCA, and PTOCA objects, the positioning information contained in the IOC structured field should be coded to provide the desired placement of the image. Bytes 0 through 5 in the IOC specify the image object area origin for IM images. The offset is measured with respect to the I,B position specified in the current LND or RCD, using the current text (I,B) coordinate system. The image object area offset should be coded as X'000000000000' to position the image at the current LND or RCD. If the image is celled, the Image Cell Position (ICP) structured field specifies an offset from the image object origin that is measured using the current text (I,B) coordinate system.

The rotation of the IM image is specified in the IOC and is measured with respect to the page coordinate system X_p-axis (origin is top-left corner of page).

Note: For page segments in MO:DCA data, if the IM image is complex (celled), it is recommended that the rotation be set to (0°,90°). For page segments in mixed data, the rotation should be set to match the current text orientation.

The Begin Page Segment (X'D3A85F') and End Page Segment (X'D3A95F') structured fields should be deleted. The remaining structured fields can then be placed in the print stream at the point where the image is to appear.

Including Standalone Presentation Text (PTX) Structured Fields

The Presentation Text (PTX) structured field is used to specify text data and the position, rotation, and fonts to be used when presenting text data. The PTX structured field was previously known as Composed Text (CTX), but its identifier of X'D3EE9B' and all its components remain the same. PTX structured fields are made up of control sequences and data. The PTX structured field is described in *Presentation Text Object Content Architecture Reference* and provides different functions in the form of control sequences. PTX is probably the most frequently used structured field in fully composed MO:DCA documents. PTX structured fields can be intermixed with line data records so long as a few rules are followed:

- Each PTX structured field should be coded as a self-contained environment. While PTX control sequences can be used to set the line spacing, page margin, data position, font, etc., these settings remain in effect only for the current PTX structured field. Processing of follow-on line data records or structured fields might change the settings. If a line data record follows a PTX, settings such as its placement and font is determined by the information in the current LND or RCD of the active Page Definition. A PTX can affect the printing of line-data records if it contains text control sequences that change inter-character and inter-word spacing, because these characteristics are not controlled by a Page Definition. If another PTX structured field follows the PTX, the text environment established by the last-used LND or RCD is re-issued before the new PTX is processed. Some presentation systems that convert the mixed-mode data to MO:DCA might also place Begin Presentation Text (BPT) and End Presentation Text (EPT) structured fields around each embedded PTX. Subsequent processing of the BPT will cause initial text conditions to be set prior to the processing of the PTX. See *Mixed Object Document Content Architecture (MO:DCA) Reference* for more information on initial text conditions set when the BPT is processed.
- Because the presentation services software considers line-data files to be mapped totally with a Page Definition, the presentation services software generates IPDS commands containing positioning and font information for every record in the file. If a record turns out to be a PTX structured field, the information in the PTX is used to create a subsequent IPDS Write Text command. If a large number of PTX structured fields are included in a line-mode data set, the additional IPDS commands generated by the presentation services software could add an unacceptable amount of processing overhead when the data set is printed.
- Page Definition information, PTX information, and any additional information contained in objects such as bar code and image placed on the page interact, so the programmer must keep careful track of the page

position and fonts in effect as records are written. For example, if the text position, text orientation, or font is not defined in a structured field or object, the values specified in the Page Definition for the current line-data record will be used. Depending on the complexity of the application, it might be easier to write fully composed output rather than using a Page Definition to set up the environment.

Figure 27. Presentation Text Structured Field

Length	X'D3EE9B'	Flag byte	Sequence number	Data

Record Format When Using PTX Structured Fields

When creating a mixed-mode data set that includes PTX structured fields, it is generally easier to use variable-length records. The PTX structured field length ranges up to 32,759 bytes. Much spool space is wasted if every record is padded out to this length, regardless of whether or not the entire 32K bytes contain valid data.

Using the PTX Structured Field

The PTX structured field contains PTOCA data, as defined in the *Mixed Object Document Content Architecture (MO:DCA) Reference*. The general format of the PTX structured field is shown in Figure 27. Either of two types of data can follow the PTX structured field introducer:

- The X'2BD3' escape sequence, followed by one or more text control sequences
- “Free-standing text”, which is a series of code points representing data to be printed

The first alternative is by far the most common use of PTX. A table of the control sequences that can be used with the PTX structured field appears in Table 11 on page 60.

The PTOCA Architecture groups control sequences into *function sets* or *subsets*. PT1 is the base subset that is supported by all AFP page printers. PT2 is a superset of PT1 that contains three additional control sequences: Underscore (USC), Overstrike (OVS), and Temporary Baseline Move (TBM). PT3 is a superset of PT2 that contains the Set Extended Text Color (SEC) control sequence for supporting spot colors and process colors in text. PT4 is a superset of PT3 that contains controls for glyph runs (used with complex Unicode text). See *Advanced Function Presentation Printer Information*, G544-3290 for information on which PTOCA subsets are supported by your printer.

In a PTX structured field, a control sequence immediately follows each X'2BD3' escape sequence. Each control sequence can be coded as *unchained* (even-numbered functions) or *chained* (odd-numbered functions). If unchained controls are used, each one must be preceded by the X'2BD3' escape sequence. In the chained format, each control sequence immediately follows the previous one with no intervening X'2BD3' escape sequence. The last control sequence in a chain must have the even-numbered (unchained) format to signal the end of the chain.

Each text control sequence is a minimum of two bytes long, where the X'2BD3' escape sequence, if present, is not counted as part of the length. The first byte indicates the length of the entire control sequence, including the length byte itself, the function byte, and any parameter bytes. The second byte contains the odd or even function code for the control sequence. A data field ranging from zero to 253 bytes follows.

One reason why free-standing text is seldom used is that one of the PTX control sequences available is *Transparent Data* (TRN), which has a string of code points as its data field, and thereby provides the actual text to be printed. Use of the TRN control sequence allows data whose encoding scheme uses the code points X'2B' or X'D3' to be included in a PTX without having these code points interpreted as an escape sequence.

The usual sequences for placing text on a page are as follows:

- Specify the beginning print position using Absolute Move Inline (AMI) and Absolute Move Baseline (AMB) control sequences
- Select the coded font to be used with the Set Coded Font Local (SCFL) control sequence
- Specify the code points of the text to be printed using a Transparent Data (TRN) control sequence

Here is an example:

```
     X'5A001BD3EE9B0000002BD304D300F004C700B403F10106DAC4C1E3C1'
```

This example begins with a X'5A' carriage control character, as would be required in the z/OS system environment. Following this byte in the example is a two-byte length field, which provides the length of the entire record (X'001B' = 27). The X'5A' character is not included in this count. The next three bytes are the Presentation Text identifier (X'D3EE9B'). Following that is the X'00' flag byte and the two-byte sequence number (X'0000'). The first two bytes of the data are the escape sequence (X'2BD3'), followed by a text control sequence that indicates chaining.

The first control sequence is an Absolute Move Baseline that specifies a baseline offset of X'00F0' logical units from the page origin. For a 240 units-per-inch coordinate system, this indicates an offset of one inch down the page.

The second control sequence is an Absolute Move Inline that specifies an Inline offset of X'B4' or 180 units from the left margin.

Following this is a Set Coded Font Local that selects the coded font that maps to font local ID 1 in the MCF structured field in the Active Environment Group for the Data Map.

The last control sequence is a six-byte-long Transparent Data, which simply contains the word `DATA` and ends the chaining sequence because it uses the X'DA' (even) function type.

Programming Tip

When deciding how to code Presentation Text structured fields, keep in mind that it is good programming practice to build as long a PTX structured field as possible, to reduce overhead in the print server associated with reading and processing many short records written by the application. Text control sequences should be chained wherever possible. While a string of unchained control sequence pairs will work also, the presence of the X'2BD3' escape sequences can use up many of the 32,759 bytes of the PTX structured field unnecessarily.

Within a fully composed document, the last control sequence in any text object must always indicate end of chaining. If PTX structured fields are intermixed with line data in a mixed-mode document, the last control sequence in the PTX must also indicate end of chaining. This can be accomplished either by specifying an even function type for the last control sequence or by ending every PTX with a No Operation control sequence with an even function type (X'02F8').

Use of Fonts

Either fixed-pitch or proportionally spaced fonts can be used to present text with the PTX structured field. Positioning of the first character in a string of data contained in the TRN control sequence can be accomplished by preceding the TRN with one of the absolute or relative move text controls, as shown in the example on page 59. If no *move* control sequences follow in the same PTX, data contained in any subsequent TRN controls will be placed immediately following the text in the preceding TRN. Font information stored in the printer is used to ensure that data does not overlap. As a result, it is possible to highlight one word in a string simply by using a Set Coded Font text control. If the PTX record shown on page 59 is extended to print the word `DATA` a second time in a different font, as in this example:

```
X'…2BD304D3010004C700B403F10106DBC4C1E3C103F10206DAC4C1E3C1'
```

then the resulting output will look like this:

DATA*DATA*

Table 11. Control Sequences Used in PTX Structured Field

PTOCA Control Sequence Function	Unchained (Even Function)	Chained (Odd Function)
PT1 Control Sequences		
Absolute Move Baseline	04D2	04D3
Absolute Move Inline	04C6	04C7
Begin Line	02D8	02D9
Begin Suppression	03F2	03F3
Draw Baseline Rule	07E6	07E7
Draw Inline Rule	07E4	07E5
End Suppression	03F4	03F5
No Operation	*xx*F8	*xx*F9
Relative Move Baseline	04D4	04D5
Relative Move Inline	04C8	04C9
Repeat String	*xx*EE	*xx*EF
Set Baseline Increment	04D0	04D1
Set Coded Font Local	03F0	03F1
Set Intercharacter Increment	04C2	04C3
Set Inline Margin	04C0	04C1
Set Text Color	0574	0575
Set Text Orientation	06F6	06F7
Set Variable Space Character Increment	04C4	04C5
Transparent Data	*xx*DA	*xx*DB
PT2 Control Sequences		
Overstrike	0572	0573
Temporary Baseline Move	*xx*78	*xx*79
Underscore	0376	0377
PT3 Control Sequences		
Set Extended Text Color	*xx*80	*xx*81

Table 11 Control Sequences Used in PTX Structured Field (cont'd.)

PTOCA Control Sequence Function	**Unchained (Even Function)**	**Chained (Odd Function)**
PT4 Control Sequences		
Glyph Advance Run	*xx*8C	*xx*8D
Glyph ID Run		*xx*8B
Glyph Layout Control		*xx*6D
Glyph Offset Run	*xx*8E	*xx*8F
Unicode Complex Text	106A	

Right-justification and centering of text cannot be done simply by using PTX control sequences. Calculations must be done in the program to place each character at the correct position on the page. This can become fairly complex if proportional fonts are used.

Boxes and Rules

The Draw Baseline Rule and Draw Inline Rule control sequences may be used to draw rules and boxes on the page to highlight information or to separate one area of the output from an adjacent area. The length and thickness of the rule must be specified in the control sequence and these values are expressed in the units of measure specified in the Presentation Text Descriptor (PTD) structured field. If the rule is to be drawn in the positive baseline or inline direction (that is, from top to bottom or from left to right), the positive number expressing the length and thickness is used. If the rules are to be drawn in the direction opposite the baseline direction (“up” relative to the data on the page) or the direction opposite the inline direction (“backward” relative to the data on the page), the line length or thickness must be coded in two's complement form.

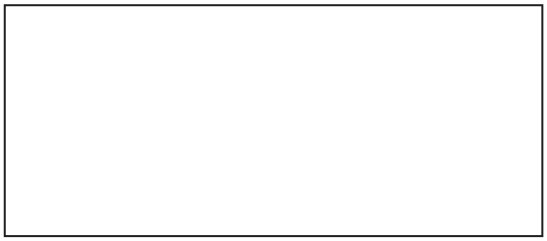
The two's complement of a two-byte hexadecimal number is obtained by inverting each bit of the number and adding a one to the low-order bit position. For example, a one-inch rule is 240 logical units long, or X'F0' L-units, when using 240 units per inch. This value can be placed directly in a Draw Inline Rule control sequence. To obtain the value to use when drawing this rule in the opposite direction, you calculate the two's complement of X'F0' by inverting to get X'FF0F' and then adding X'0001'. The result is X'FF10'. The full, chained control sequence that draws a 3-unit thick rule one inch long in the “backward” direction is X'07E5FF10000300'.

The third and fourth data bytes of the draw rule control sequence specify the thickness of the rule. To determine whether a positive number or the two's complement number is needed, you should decide in which direction to add pels, starting from the initial print position. For inline rules, a positive thickness value adds pels from top to bottom, while a two's complement value adds pels from bottom to top. For baseline rules, pels are added to the right if the thickness value is positive and to the left if the thickness is expressed as a two's complement (negative) number.

These details come into play when drawing boxes with mitered corners. To make the box outline complete and not have a gap between the end of a baseline rule and the start of an inline rule beneath it, you might have to change the origin point of the rule, the length of the rule, or the rule thickness from positive to negative. Gaps between inline and baseline rules become increasingly visible as the thickness of the rules increase.

Figure 28 on page 62 illustrates a text-control sequence to draw a box one inch high by two inches wide. The rules that generate the box are four pels thick, so the lengths of the rules in the Draw Rule control sequences have been extended by 4 pels where necessary to make sure the corners are complete.

Figure 28. Text Controls to Draw a Box



```
… 04C7000F04D300F007E501E000040007E700F0000400…
    < AMI >< AMB ><Inline Rule><Baseline Rule>
                 (bottom side)     (left side)
… 04C901E004D500F007E5FE20FFFC0007E6FF10FFFC00
    < RMI >< RMB ><Inline Rule><Baseline Rule>
                   (top side)      (right side)
```

Composed Documents

The discussion up to this point has described how line-mode data can be printed in any desired format by using an appropriate Page Definition. Information has also been provided on how formatting can be changed at selected points in the output by using conditional processing or embedded IDM or IMM structured fields to select a new Data Map or Medium Map for use with subsequent line-data records. This technique of switching among maps in the Page Definition or Form Definition requires advance knowledge of all the output formats that will be used and the appropriate Data Maps and Medium Maps must be coded. Using this technique also means that formatting can be changed only on a page basis. When a new Data Map or Medium Map is selected, processing of the current page is ended and the next line-data record appears on the following page. It is not possible to reach an intermediate point on a page and then select a new Data Map for use in processing the remaining records on the page.

These and other limitations often make it impossible to use external formatting objects to produce the exact kind of output desired for a particular application. If the positioning and formatting needed for each page of your application output is not known in advance, then the application data probably does not lend itself to outboard formatting using a Page Definition. In these cases, you should consider generating fully-composed documents, rather than line-mode or mixed-mode data.

Some examples of applications that have been developed using fully-composed output are:

- Utility bills containing line-by-line details and graphical representations of energy use for each customer compared to the average for all customers
- Insurance policies with clauses, supplements, and detailed client-specific information that vary from one policy to the next
- Financial statements containing sections that describe specific investments, payments, or accounts, each of which vary considerably from one statement to the next. Boxes and shading might be used to highlight certain items of information and the location of the boxes or shaded areas can be anywhere on the page.

In many cases the text in the output from these applications is also printed using proportional fonts, which can be more difficult to place using a Page Definition than fixed-pitch fonts. Note that in composed documents, the PTX structured fields must be bracketed by Begin Presentation Text (BPT) and End Presentation Text (EPT) structured fields. These structured fields are allowed only in fully composed documents. They cannot be used to bracket a set of PTX structured fields to be included in a line-data file.

For a definition of composed documents, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Programming Options

Software packages are available that can be used to generate fully composed MO:DCA documents. Another option is to develop a custom application that produces only the specific output desired.

The MO:DCA data stream generated by such an application will be transformed directly into IPDS by the print services program. No optimizing is performed on MO:DCA data. As a result, the application developer should be aware of throughput considerations associated with the MO:DCA structure. Such considerations are highlighted in this chapter in boxes titled “Programming Tip”.

Overall Document Structure

A fully composed document will conform to the structure shown on the right side of Figure 30 on page 170. Each document is composed of one or more pages that have the format shown in Figure 32 on page 171. Each page must begin with an Active Environment Group, but the actual objects that appear on the page (text, image, graphics, or bar code) follow the AEG and can appear in any order. The application programmer works with these objects, so an understanding of their format, use, and placement on the page can be helpful when developing an AFP program.

Document Indexing

Indexing and attribute tagging structured fields may be added to documents to permit the selective retrieval of specific pages and page groups for later viewing or printing. The MO:DCA architecture defines six structured fields specifically for this purpose:

- Begin Document Index
- Index Element
- Tag Logical Element
- End Document Index
- Begin Named Page Group
- End Named Page Group

An index is bracketed by Begin Document Index and End Document Index structured fields. It may contain Index Element (IEL) structured fields used to locate objects in a document and Tag Logical Element structured fields, used to tag pages and page groups with attribute names and their values. Pages in a document may be grouped for indexing using the Begin Named Page Group (BNG) and End Named Page Group (ENG) structured fields.

In AFP environments, the document index is located external to the document.

Programmers are free to include Begin Named Page Group (BNG), End Named Page Group (ENG), and Tag Logical Element (TLE) structured fields in the body of fully composed documents. However, the BNG, ENG, and TLE structured fields are not supported for indexing in a line-data or mixed-data environment.

Document Links

Fully composed MO:DCA documents may contain logical links between document components. An example is a hypertext link from an area on page N that contains a technical term to an area on page M that contains the term's definition. Such links are specified using Link Logical Element (LLE) structured fields. LLE structured fields are not supported in line-data or mixed-data documents.

Chapter 5. Structured Fields in a Page Definition and in Line Data

This chapter defines the structured fields used in a Page Definition print control object. It also describes special functions used with certain MO:DCA structured fields when they occur in line-mode or mixed-mode data. Refer to the *Mixed Object Document Content Architecture (MO:DCA) Reference* for definitions of structured fields not included in this manual. Any conflicts arising from definitions in this manual and the *Mixed Object Document Content Architecture (MO:DCA) Reference* are resolved by the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Structured Field Format

Structured fields used in Page Definitions are registered in the MO:DCA architecture and follow the MO:DCA structured field syntax rules. A summary of the syntax rules for Page Definition structured fields follows.

Structured Field Introducer				
Length (2B)	SF Identifier (3B)	Flags (1B)	Reserved X'0000'	Data

Length A two-byte count field that specifies the length of the structured field. The length value can range from 8 to 32,767. The count includes the length field itself, the other structured field introducer parameters, and the structured field data contents, including padding bytes when present.

SF Identifier A three-byte value that identifies the type of structured field.

Flag A one-byte field that specifies the value of optional structured-field indicators.

Bits 0–3 and 5–7 are not used and should be set to zero.

Bit 4 of this byte is a padding indicator. If this bit is on, the structured field data includes padding bytes.

Reserved; X'0000'

The two-byte reserved field is used by some applications to specify a structured field sequence number so that the structured field can be located more easily in a data stream. These applications define the sequence-numbering conventions used. Print-services products do not validate sequence numbering; therefore, applications can use any numbering convention. This field is reserved in MO:DCA data streams and should be set to zero.

Data Not all structured fields include a data portion. If present, this portion contains specific orders, parameters, and data appropriate for the given structured field. The length of the data field can range from 0 to 32,759.

Padding If bit 4 in the flag byte (the padding indicator) is set to B'1', padding bytes follow the data field. If padding is indicated, the length of the padding is specified in the following manner:

- For 1 or 2 bytes of padding, the length is specified in the last padding byte.
- For 256 to 32,759 bytes of padding, the length is contained in the last three bytes. The last byte is X'00', and the two preceding bytes specify the padding length.
- For 3 to 255 bytes of padding, the length can be specified by either method.

Note: The length count of the padding data includes the length field itself.

Structured Field Descriptions

The description for each structured field contains the following information:

- Purpose
- Meaning and allowed values of variable parameters
- Contents of constant parameters

Notation Conventions

The bold-faced heading for each structured field contains the following information:

- The three-letter abbreviation
- The three-byte hexadecimal code
- The full name of the structured field

The following conventions apply in the descriptions of the structured fields:

- The byte position of each structured field parameter is given. The first byte of the data field is byte 0.
- If the parameter is a variable, the name and function of the parameter is given.
- Each parameter is specified as either *optional* (O) or *mandatory* (M).
- Any valid triplets (see “Structured Field Triplets” on page 67 for a description) listed in the tables are specified as either optional or mandatory. The definition of each triplet follows the description of the structured field in which it appears.
- Any number not preceded by X (hexadecimal) or B (binary) is a decimal number.
- If the parameter is a constant or reserved parameter, only the parameter contents are given; for example, X'0960' or B'001'. A reserved parameter is one that has no meaning at present, but might in the future. Reserved bytes should be set to X'00' by data stream generators and should be ignored by data stream receivers.
- If the same parameter occurs more than once but contains different values, the values in the last parameter are used.
- The tables specify the type of parameter, where appropriate. The parameter type notations are:

  UBIN Unsigned binary integer: a positive integer where the high-order bit may be used as a data bit.

  SBIN Signed binary integer: an integer where the high-order bit is the sign (plus or minus) of the integer and cannot be used as a data bit.

  BITS Bit String: a string consisting solely of bits; binary flags where each binary bit can be assigned a specific meaning.

  CHAR Character encoding: a string of one or more code points for alphanumeric characters. For example, X'C1C2C3F1' in EBCDIC can represent ABC1.

  CODE Architected value: a code assigned to a specific item.

Structured Field Triplets

Several structured fields contain self-identifying parameters called *triplets* in their data field. A triplet contains three components: the triplet length, the triplet identifier, and the triplet values. See Table 12.

Table 12. Structured Field Triplet Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Triplet Length	3–254	Specifies the length of the triplet, including this byte	M
1	CODE	Triplet Identifier		Identifies the triplet	M
2–*n*		Triplet Data		Contains the data for this triplet	M

External Resource Object Naming Conventions

MO:DCA objects can be named using one of the following two formats:

- Token name. This name is specified using a fixed-length 8-byte parameter on Begin, Invoke, Map, and Include structured fields
- Fully qualified name. This name can be up to 250 bytes long and is specified using the Fully Qualified Name (FQN) X'02' triplet on Begin, Map, and Include structured fields, as well as on object-processing structured fields. For names, the FQNFmt parameter on this triplet is set to X'00' to specify a character string format and the FQNType parameter specifies how the name is used. When a fully qualified name is specified using FQNType X'01' on a Begin structured field, it overrides any token name that might have been specified on the structured field.

MO:DCA object names are encoded using the code page and character set specified in a Coded Graphic Character Set Global ID (X'01') triplet, except in those cases where the name defines a fixed encoding. The X'01' triplet can specify the encoding in two forms; use of the Coded Character Set Identifier (CCSID) form is recommended. For a definition of the X'01' triplet, see *Mixed Object Document Content Architecture (MO:DCA) Reference*.

The X'01' triplet may be specified on most MO:DCA structured fields that contain character data such as an object name. Careful specification of code page and character set is essential for interchange since the system defaults for code page and character set might vary from one system environment to another.

In AFP environments, print servers treat the object name—other than TrueType and OpenType full font name—as an external resource library member name and attempt to process a resource library member with the same name. This means that the external names are subject to the system imposed file naming rules.

To ensure portability across all AFP platforms, external object names other than TrueType and OpenType full font names must be composed according to the following conventions:

- Names consist only of the following characters: A–Z, 0–9, $, #, @. When the object name is specified using the fixed-length 8-byte token name parameter, a trailing space character (X'40' in the EBCDIC encoding) or a trailing null code point (X'00') is assumed to terminate the name.

- To ensure portability across older versions of print servers that do not support encoding definitions in the X'01' triplet, names should use only the recommended characters and be encoded in EBCDIC using code page 500 and a character set that includes the above-mentioned characters. The preferred character set is 961, which includes only those characters, however character sets such as 697, which contain additional characters, are also appropriate. With this encoding, the code points for the characters are:
  A–I (code points X'C1'–X'C9')
  J–R (code points X'D1'–X'D9')
  S–Z (code points X'E2'–X'E9')
  0–9 (code points X'F1'–X'F9')
  $, #, and @ (code points X'5B', X'7B', and X'7C', respectively)

  Note that such older print servers normally assume this EBCDIC encoding as the default encoding for the document. This EBCDIC encoding can be identified with CCSID 500, which represents the combination of code page 500 and character set 697.

TrueType and OpenType full font names specified in the MDR structured field are not restricted to these characters and might be encoded as required by the AFP-generating application. However, since these names are used to search inline font containers and Resource Access Tables (RATs) that use a fixed UTF-16BE encoding for full font names, efficiency is gained if the full font names in the MDR are also encoded in UTF-16BE. This avoids an encoding conversion. The UTF-16BE encoding can be identified with CCSID 1200. This encoding needs to be specified with a X'01' triplet on the MDR that specifies the full font name.

Begin and End Structured Fields

Begin structured fields identify the beginning of an object in a print data stream or of a data stream resource. All Begin structured fields are used in conjunction with corresponding End structured fields to delimit specific objects in a document, page, or resource.

When a Begin data object structured field (BPT, BGR, BIM, or BBC) is encountered in a page, an overlay, or a page segment, the initial data presentation conditions are set from values specified in the data descriptor structured field for the object, prior to processing the object data. If no initial conditions are specified in the object data descriptor or if null values are specified, then the initial conditions are set to the system default. If no system default is defined, the device default is used.

Any triplet parameter encountered on any of the Begin structured fields that is not explicitly defined to be valid in the structured field definition is ignored. The document presenter assumes that these fields are informational although they might have meaning in other applications.

Begin Data Map (BDM)

The Begin Data Map structured field begins a Data Map resource object.

BDM (X'D3A8CA') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A8CA'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	DMName		Name of the Data Map	M
8	CODE	DatFmt	 X'00' X'01' X'02'	Data formatting specified by this Data Map: Data Map formats traditional line data using LNDs Data Map formats line data containing record format IDs using RCDs Data Map formats XML data containing start and end tags using XMDs	O
9–*n*		Triplets		See BDM Semantics for triplet applicability.	M

BDM Semantics

DMName
Token name of the Data Map

This is a mandatory parameter because an Invoke Data Map (IDM) structured field selects a Data Map by specifying its token name.

DatFmt
An optional parameter that determines how the Data Map is used to format line data

If this parameter is not specified, the architected default is X'00'.

Value	Description
X'00'	The Data Map contains LNDs and is used to format traditional line data that may contain CCs and TRCs.
X'01'	The Data Map contains RCDs and is used to format line data that contains record format IDs and may also contain CCs.
X'02'	The Data Map contains XMDs and is used to format XML data that contains start and end tags.

Triplets
Optional triplets

See the following for detailed information about the triplets permitted on the BDM structured field:
“Encoding Scheme ID (X'50') Triplet” on page 70
“Page Count Control (X'7C') Triplet” on page 71
“Margin Definition (X'7F') Triplet” on page 73

Notes:

1. If a triplet is included on this structured field, the optional positional parameters become mandatory.

2. If one of the Data Maps in a PageDef contains LNDs, then all of the Data Maps in a PageDef must be LND based.
3. If one of the Data Maps in a PageDef contains RCDs, then all of the Data Maps in a PageDef must be RCD based and subpages are not used (the Data Map contains only one subpage).
4. If one of the Data Maps in a PageDef contains XMDs, then all of the Data Maps in a PageDef must be XMD based and subpages are not used (the Data Map contains only one subpage).

BDM Triplets

Encoding Scheme ID (X'50') Triplet

This triplet is an optional triplet when used with DatFmt X'00' (formatting with LNDs) and X'01' (formatting with RCDs) but it is mandatory when used with DatFmt = X'02' (formatting with XMDs). It is used to specify the encoding scheme associated with the user data, the XML Name (X'8A') triplet specified on XMDs, the Record Descriptor ID field specified on RCDs, the Field Delimiter specified on RCDs and XMDs, the Fixed text specified in the Fixed Data Text (FDX) structured field, and the Comparison String field of the Conditional Processing Control (CCP) structured fields.

The values supported in the ESidUD field of the Encoding Scheme ID triplet when formatting data with a Page Definition are:

- X'6100': EBCDIC Presentation SBCS
- X'2100': PC Data SBCS (ASCII)
- X'7807': UTF-8
- X'7200': UTF-16

When this triplet is specified for LND or RCD processing, it is used to determine if searching for the Byte Order Mark (BOM) is necessary. If the triplet specifies UTF-8 or UTF-16, the first bytes (following any CC or TRC) of the first line or record of the line data are examined to see if the BOM has been placed in the data. If the BOM is in the data, it is removed so it is not considered part of the user data.

For XMD processing, this triplet is mandatory and the first bytes of the first line or record of the print file are always examined to see if the BOM is in the data.

For more information about BOM processing, see "Unicode Line Data" on page 13.

The Encoding Scheme ID triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Notes:

1. ESidUD is required for Data Maps that are to print XML data.
2. This triplet may occur once. If this triplet is specified more than once, only the first is used.
3. Each Encoding Scheme ID triplet specified on each BDM structured field of a single Page Definition must specify the same encoding.
4. This Encoding Scheme ID triplet overrides the encoding specified in the XML data.
5. Except for certain situations in processing XML data with FOCA fonts (see "XML Data" on page 13), the font selected to print the data must match the encoding of the user data specified in this triplet.

Page Count Control (X'7C') Triplet

This is an optional triplet that may occur once. If this triplet is specified more than once, only the first is used. It is used only if DatFmt = X'01' (formatting with RCDs) or X'02' (formatting with XMDs). It is used to specify how the page count is initialized and maintained for the active Data Map. If this triplet is specified on a Data Map that contains LNDs, it is ignored.

Triplet X'7C' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	7	Length of the triplet, including Tlength	M
1	CODE	Tid	X'7C'	Identifies the Page Count Control triplet	M
2–3	CODE	PageNum	X'0001'–X'FFFF'	Initial page number	M
4			0	Reserved; should be zero	M
5	CODE	CountCtr	 X'00' X'01' X'02' X'03'	Page count control for Data Map: Stop Resume Continue Reset	M
6	BITS	CountFlgs		Bits that specify additional page count controls See Triplet X'7C' Semantics for bit definitions.	M

Triplet X'7C' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Page Count Control triplet

PageNum Initial page number

This field specifies the initial page number to be set into the page count when the page count control specifies X'01' (Resume) or X'03' (Reset).

CountCtr Page count control

This field specifies how the page count is initialized and maintained for the active Data Map.

Value Description

X'00' Stop

When this Data Map is invoked, the page count is initialized to the last page number used with the previous Data Map, which is the current page number. If there is no current page number, it is initialized to the value specified by PageNum. Once the page count is initialized, it is *not* incremented for the duration of this Data Map.

X'01' Resume

On the *first* invocation of this Data Map, the page count is initialized to the value specified by PageNum. On each successive invocation of this Data Map, the page count is initialized to the last page number used on the previous invocation of this Data Map. Once the page count is initialized, it is incremented for every page presented with this Data Map.

X'02' Continue

When this Data Map is invoked, the page count is initialized to the last page number used with the previous Data Map, which is the current page number. If there is no current page number, such as when this is the first Data Map invoked for the job, it is initialized to the value specified by PageNum. Once the page count is initialized, it is incremented for every page presented with this Data Map.

X'03' Reset

When this Data Map is invoked, the page count is initialized to the value specified by PageNum. Once the page count is initialized, it is incremented for every page presented with this Data Map.

All others
Reserved

CountFlgs This field specifies additional page count controls.

Bit 0 Count control for MO:DCA pages

Value	Description
B'0'	Do not count MO:DCA pages that occur in mixed-mode data.
B'1'	Count MO:DCA pages that occur in mixed-mode data.

Bit 1 Count control for constant pages; that is, pages that contain no variable data

Such pages are generated when:

- The Formdef specifies “constant form”.
- The Formdef specifies N-up and no variable data is allowed on the page.

Value	Description
B'0'	Do not count constant pages.
B'1'	Count constant pages.

Bits 2–7
Reserved; all bits should be B'0'

If this triplet is not specified, the defaults are CountCtr = X'02' (Continue) and CountFlgs = B'00' (do not count MO:DCA pages and constant pages that occur in mixed-mode data).

Margin Definition (X'7F') Triplet

This is an optional triplet that is used only if DatFmt = X'01' (formatting with RCDs) or X'02' (formatting with XMDs). This triplet may occur once. If this triplet is specified more than once, only the first is used. It is used to specify the page margins for the Data Map. These margins are used for logical page eject processing and for graphics processing. If this triplet is specified on a Data Map that contains LNDs, it is ignored.

Triplet X'7F' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	14	Length of the triplet, including Tlength	M
1	CODE	Tid	X'7F'	Identifies the Margin Definition triplet	M
2–5	CODE	TxtOrent	 X'0000 2D00' X'2D00 5A00' X'5A00 8700' X'8700 0000'	Text (I,B) Orientation: 0,90 degrees 90,180 degrees 180,270 degrees 270,360 degrees	M
6–7	UBIN	LeftMar	0 to page extent minus 1	Left Margin Offset from page edge	M
8–9	UBIN	TopMar	0 to page extent minus 1	Top Margin Offset from page edge	M
10–11	UBIN	RightMar	0 to page extent minus 1	Right Margin Offset from page edge	M
12–13	UBIN	BotMar	0 to page extent minus 1	Bottom Margin Offset from page edge	M

Triplet X'7F' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Margin Definition triplet

TxtOrent Text Orientation

This field specifies the text orientation that is used to fix the page margins for this Data Map. See Figure 29 on page 75.

The margins are fixed for the Data Map; that is, they do not change when the text orientation changes in the Data Map. For example, if this parameter specifies a (0,90) degree text orientation, the top-left diagram in Figure 29 on page 75 shows how the LeftMar, TopMar, RightMar, and BotMar parameters define the left, top, right, and bottom margins, respectively. Once specified, these margins define a bounding box for the Data Map that is indicated by the dashed lines.

Note that if the text orientation is changed within the Data Map, the same bounding box applies to the new orientation, but the name of the margins change in the new orientation. For example, if the new text orientation is (90,180) degrees, as shown in the top-right diagram of Figure 29 on page 75, the left margin in the new orientation is actually defined by the TopMar parameter, the bottom margin is defined by the LeftMar parameter, and so on. Therefore, when a possible baseline overflow is evaluated in the new orientation, the print position is actually checked against the LeftMar parameter, which is the bottom margin for that text orientation.

Similar processing occurs for graphics that are generated by an RCD or XMD. If the text orientation for the graphics is (90,180) degrees, the bottom margin is defined by the LeftMar parameter. Therefore, active lines or boxes that are ended at the bottom margin are actually ended at the margin defined by the LeftMar parameter, which is the bottom margin for that text orientation.

LeftMar Left Margin

This field specifies the offset of the left margin along the i axis from the left edge of the page. The left edge of the page is the zero position on the i axis.

TopMar Top Margin

This field specifies the offset of the top margin along the b axis from the top edge of the page. The top edge of the page is the zero position on the b axis.

RightMar Right Margin

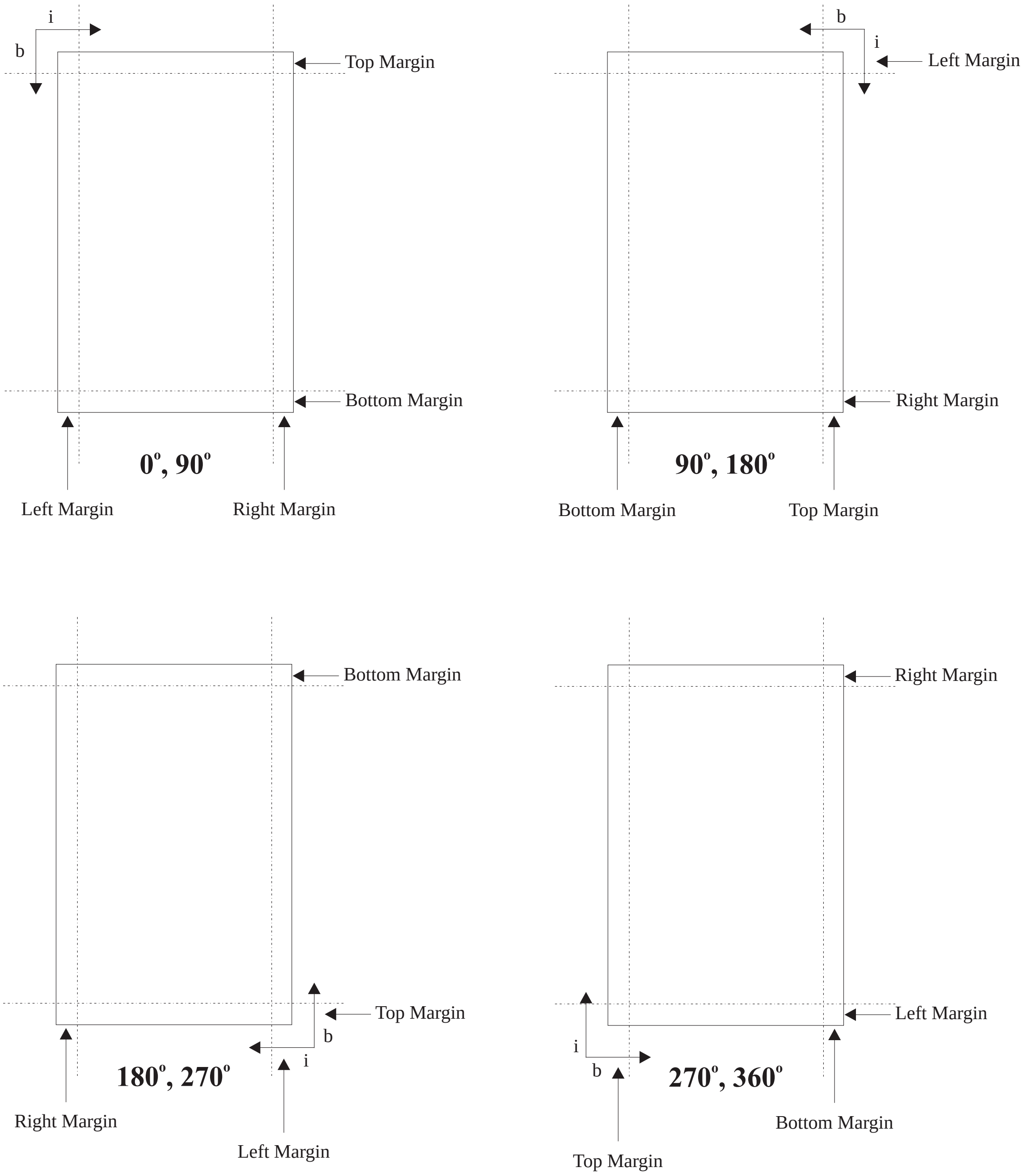
This field specifies the offset of the right margin along the i axis from the right edge of the page.

BotMar Bottom Margin

This field specifies the offset of the bottom margin along the b axis from the bottom edge of the page.

If this triplet is not specified, the default is a text orientation of (0,90) degrees and all margin offsets set to X'0000'.

Figure 29. Relationship of Margin Definition to Text Orientation



Begin Data Map Transmission Subcase (BDX)

The Begin Data Map Transmission Subcase structured field begins a Data Map Transmission Subcase object, which contains the structured fields used to map lines of data to the page.

BDX (X'D3A8E3') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A8E3'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	DMXName		Name of the Data Map Transmission Subcase	O

BDX Semantics

DMXame Token name of the Data Map Transmission Subcase

This is an optional parameter.

Begin Page Map (BPM)

The Begin Page Map structured field begins a Page Map resource object, also called a *Page Definition* or *PageDef*. A Page Definition is a print control resource object used to compose line data into pages for printing on page printers.

BPM (X'D3A8CB') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A8CB'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	PMName		Name of the Page Map	O

BPM Semantics

PMName Token name of the Page Map

This is an optional parameter.

Conditional Processing Control (CCP)

The Conditional Processing Control structured field defines tests to be performed on selected input records in line data and specifies the actions to take based on the test results. This optional structured field is selected with LND, RCD, or XMD structured fields in the Page Definition. An LND, RCD, or XMD can have a unique CCP associated with it or it can reference a CCP that has already been used. In either case, the CCP is referenced with the CCPID field of the LND, RCD, or XMD. If a CCP structured field is included in a Page Definition, it must appear before the Data Maps in the Page Definition.

CCP (X'D3A7CA') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A7CA'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–1	CODE	CCPid	X'0001'–X'FFFF'	CCP Identifier	M
2–3	CODE	NxtCCPid	X'0001'–X'FFFF'	Next CCP Identifier	M
4	BITS	CCPFlgs			M
Bit 0			B'0'–B'1'	Before subpage actions	
Bit 1			B'0'–B'1'	After subpage actions	
Bit 2			B'0'–B'1'	Spacing actions	
Bits 3–7			B'00000'	Reserved	
5			X'00'	Reserved	M
6–7	UBIN	NumRGs	X'0001'–X'FFFF'	Number of repeating groups	M
8–9	UBIN	RGLgth	X'0015'–X'FFFF'	Length of each repeating group	M
10–11	UBIN	CSLgth	X'0000'–X'FFFF'	Length of comparison string	M
12–*n*		Repeating groups		One or more repeating groups	M

CCP Semantics

CCPid CCP Identifier

CCPs can be chained to handle complex data within multiple CCP records. If this is the first or only CCP, this field matches the CCP Identifier in CCPID field of the LND, RCD, or XMD. Subsequent CCPs in a chain have unique identifiers.

NxtCCPid Next CCP Identifier

This field contains the identifier of the next CCP to be processed. A value of zero indicates that this is the last or only CCP to process.

CCPFlgs Conditional Processing Flags

Bit 0 If B'1', this CCP requires action before a subpage boundary.

Bit 1 If B'1', this CCP requires action after a subpage boundary.

Bit 2 This field provides spacing action at the top of a page when the following conditions are present:

- A new page is started by a true condition.
- Conditional processing is active.
- ANSI control characters are used.

If B'0', space according to the ANSI carriage control value. If B'1', suppress spacing; print only.

Bits 3–7
Reserved; should be B'00000'

Note that when CCPs are used with RCDs to process record-format line data or XMDs to process XML data, the whole page is the only sub-page, therefore timing actions that specify a subpage are processed against the complete page.

NumRGs Number of Repeating Groups

This field indicates the number of repeating groups for this CCP. The value must be greater than zero.

RGLgth Length of each repeating group

This field indicates the length of the repeating groups to follow. The length must be at least 21 bytes.

CSLgth Length of Comparison String

This field indicates the length of the text string in the Comparison String parameter within the repeating group.

Each repeating group of the CCP contains action information. See Table 13 for the definitions of the CCP repeating groups.

Table 13. CCP Repeating Group Structure

Bytes	Parameter Name	Type	Description and Values
0	Timing of Action	UBIN	**0** Take default action; the default is for action to be taken immediately before presenting current line **1** Take action immediately before presenting current line **2** Take action before presenting current subpage **129** Take action immediately after presenting current line **130** Take action after presenting current subpage
1	Medium Map Action	UBIN	**0** Ignore **1** Continue using current medium map with page eject **2** Invoke named Medium Map **3** Invoke first Medium Map **4** Invoke next Medium Map
2–9	Medium Map Name	CHAR	Any 8-byte value
10	Data Map Action	UBIN	**0** Ignore **1** Continue using current Data Map with page eject **2** Invoke named Data Map **3** Invoke first Data Map **4** Invoke next Data Map
11–18	Data Map Name	CHAR	Any 8-byte value

Table 13 CCP Repeating Group Structure (cont'd.)

Bytes	Parameter Name	Type	Description and Values
19	Comparison	UBIN	**0** Any change **1** Equal to **2** Less than **3** Equal to or less than **4** Greater than **5** Equal to or greater than **6** Not equal **7** Take the action without comparison
20–*nnn*	Comparison String	CHAR	1 to *nnn* bytes, where *nnn* plus the total length of the fixed-length fields in the CCP is less than, or equal to, 32,759 bytes

Byte 0 Timing of Action

This parameter indicates when the action specified for the CCP is to be taken. Only values of 0, 1, 2, 129 (X'81'), and 130 (X'82') are allowed.

Byte 1 Medium Map Action

This parameter indicates what action is to be taken for the Medium Map when the conditional processing test for a comparison field is true.

Bytes 2–9 Medium Map Name

If the Medium Map Action parameter is 2 (Invoke named Medium Map), these 8 bytes contain the name of the Medium Map to be used.

Byte 10 Data Map Action

This parameter indicates the action to be taken for the Data Map when the conditional processing test for a comparison field is true.

Bytes 11–18 Data Map Name

If the Data Map Action parameter is 2 (Invoke Named Data Map), these 8 bytes contain the name of the Data Map to be used.

Byte 19 Comparison

This one-byte parameter indicates the type of comparison between the input data and the comparison string (bytes 20–*nnn*).

Value 0 (“any change”) specifies that the contents of the comparison field are to be compared with the contents of the comparison field in the last preceding record that was checked using the current CCP structured field. An “any change” comparison is true when the contents of the comparison field have changed from one record being checked to the next. If the field lies outside the boundary of the current input record, which might occur with variable-length records or with truncated trailing blanks, the comparison is false, and the current records are not used in future comparisons.

An “any change” comparison is always false the first time the CCP structured field is used. Whenever a new Data Map is invoked, all comparisons are reset, and comparison field values in the previous Data Map are not retained.

Byte 20–*nnn* Comparison String

This variable-length parameter indicates the text string against which a comparison test is to be performed, if the Comparison parameter has a value between 1 and 6.

The length of the text string is determined by a value contained in bytes 10–11 of the CCP structured field.

Note: To be able to match this Comparison String to input data, the encoding of the text specified in this parameter must match the encoding of the input data.

Data Map Transmission Subcase Descriptor (DXD)

The Data Map Transmission Subcase Descriptor structured field is supported only for migration purposes.

DXD (X'D3A6E3') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A6E3'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–3		ConData		Constant data	M

DXD Semantics

ConData Constant data

This field must be set to X'0001 00FF', but is not checked.

End Data Map (EDM)

The End Data Map structured field terminates the Data Map object initiated by a Begin Data Map structured field.

EDM (X'D3A9CA') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A9CA'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	DMName		Name of the Data Map	O

EDM Semantics

DMName Token name of the Data Map being terminated

If a name is specified, it must match the name in the most recent Begin Data Map structured field. If the first two bytes of this parameter contain the value X'FFFF', the name matches any name specified on the corresponding Begin Data Map structured field.

End Data Map Transmission Subcase (EDX)

The End Data Map Transmission Subcase structured field terminates the Data Map Transmission Subcase initiated by a Begin Data Map Transmission Subcase structured field.

EDX (X'D3A9E3') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A9E3'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	DMXName		Name of the Data Map Transmission Subcase	O

EDX Semantics

DMXame Token name of the Data Map Transmission Subcase being terminated

If a name is specified, it must match the name in the most recent Begin Data Map Transmission Subcase structured field. If the first two bytes of this parameter contain the value X'FFFF', the name matches any name specified on the corresponding Begin Data Map Transmission Subcase structured field.

End Page Map (EPM)

The End Page Map structured field terminates the Page Map object initiated by a Begin Page Map structured field.

EPM (X'D3A9CB') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A9CB'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	PMName		Name of the Page Map	O

EPM Semantics

PMName Token name of the Page Map being terminated

If a name is specified, it must match the name in the most recent Begin Page Map structured field. If the first two bytes of this parameter contain the value X'FFFF', the name matches any name specified on the corresponding Begin Page Map structured field.

Fixed Data Size (FDS)

The Fixed Data Size structured field specifies the number of bytes of text found in the following Fixed Data Text (FDX) structured fields.

FDS (X'D3AAEC') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3AAEC'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–1	UBIN	TxtLgth	1–65,535	Number of data bytes in following FDX structured fields	M

FDS Semantics

TxtLgth Number of bytes of text in the FDX structured fields that immediately follow

If no fixed data text exists in this Data Map Transmission Subcase, the FDS and FDX structured fields should not be specified.

Fixed Data Text (FDX)

The Fixed Data Text structured field contains text that can be selected and presented with LND, RCD, or XMD structured fields in the Page Definition. This text is used when flag bit 7 of the LND, RCD, or XMD is set to B'1'. Any number of FDX structured fields can appear, but the total number of data bytes must match bytes 0–1 of the Fixed Data Size (FDS) structured field. The output should fit on the page and the fit can be affected by the size of the font used.

The DataStrt and DataLgth fields of the LND, RCD, or XMD specify the fixed data offset and length.

FDX (X'D3EEEC') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3EEEC'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–*n*	CHAR	Text		Fixed text to be added	O

FDX Semantics

Text Code points of the fixed text to be added to the page

From 0 to 32,743 bytes may be specified.

Invoke Data Map (IDM)

The Invoke Data Map structured field selects a new Data Map for printing line data and ends the current line-format page. With LND Data Maps, processing begins with the first Line Descriptor (LND) structured field of the invoked Data Map for the next line-format page. With RCD Data Maps, processing begins with the first Record Descriptor (RCD) structured field that matches the Record ID of the current line-data record. With XMD Data Maps, processing begins with the first XML Descriptor (XMD) structured field that matches the current Qualified Tag.

IDM (X'D3ABCA') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3ABCA'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	DMName		Name of the new Data Map in the Page Definition	M

IDM Semantics

DMName Token name of the new Data Map in the currently active Page Definition

This name must match the name on the Begin Data Map (BDM) structured field. If the name is shorter than eight bytes, trailing blanks must be added.

Include Object (IOB)

Notes:

1. The IOB is a MO:DCA structured field. The following description documents its use in line-mode and mixed mode applications and introduces parameter values that are only valid in these environments. For the formal definition of the IOB structured field, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.
2. When processing XML data, the IOB may only be used in a Page Definition resource.

An Include Object structured field references an object and optionally contains parameters that identify the object and that specify presentation parameters such as object position, size, orientation, mapping, and default color. Where the presentation parameters conflict with parameters specified in the object's environment group (OEG), the parameters in the Include Object structured field override. If the referenced object is a page segment, the IOB parameters override the corresponding environment group parameters on all data objects in the page segment.

IOB (X'D3AFC3') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3AFC3'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	ObjName		Name of the object	M
8				Reserved; should be zero	M
9	CODE	ObjType		See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for the supported object types.	M
10–12	SBIN	XoaOset	-32,768 – +32,767	X-axis origin of the object area	M
			X'FFFFFF'	Use the X axis origin defined in the object (not supported if RefCSys=X'00')	
13–15	SBIN	YoaOset	-32,768 – +32,767	Y-axis origin of the object area	M
			X'FFFFFF'	Use the Y-axis origin defined in the object (not supported if RefCSys=X'00')	
16–17	CODE	XoaOrent	X'0000' X'2D00' X'5A00' X'8700'	The object area's X-axis rotation from the X axis of the reference coordinate system: 0 degrees 90 degrees 180 degrees 270 degrees	M
			X'FFFF'	Use the X-axis rotation defined in the object (not supported if RefCSys=X'00')	

Offset	Type	Name	Range	Meaning	M/O
18–19	CODE	YoaOrent	 X'0000' X'2D00' X'5A00' X'8700'	The object area's Y-axis rotation from the X axis of the reference coordinate system: 0 degrees 90 degrees 180 degrees 270 degrees	M
			X'FFFF'	Use the Y axis rotation defined in the object (not supported if RefCSys=X'00')	
20–22	SBIN	XocaOset		See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for the supported range.	M
23–25	SBIN	YocaOset		See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for the supported range.	M
26	CODE	RefCSys	 X'00' X'01'	Reference coordinate system: Text (I,B) coordinate system defined by current LND, RCD, or XMD Page or overlay coordinate system	M
27–*n*		Triplets		See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for triplet applicability.	M

IOB Semantics

For a complete definition of the IOB semantics, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*. The following describes parameter values that are unique to the IOB when used in line-mode or mixed-mode environments.

XoaOset If RefCSys = X'01', this parameter specifies the offset along the X axis, X_{pg} or X_{ol}, of the including page or overlay coordinate system to the origin of the X axis, X_{oa}, of the object area coordinate system. The value for this parameter is expressed in terms of the number of page or overlay coordinate system X-axis measurement units. If the referenced object specifies an Object Environment Group (OEG), this parameter overrides the corresponding parameter in the Object Area Position (OBP) structured field of the OEG. If the object is a page segment, this parameter overrides the corresponding OBP parameters in the environment groups of all objects that comprise the page segment and specifies the object area offsets from the page or overlay origin for all data objects in the page segment. A value of X'FFFFFF' indicates that the X-axis offset specified in the object's OEG is to be used; therefore the offset value (-1) is excluded from the allowed range. If the object does not specify the X-axis offset in an OEG, the architected default is X'000000'.

If RefCSys = X'00', this parameter specifies the offset along the I axis to the I-position of the current LND, RCD, or XMD. The value for the parameter in this case is expressed in terms of the number of I-axis measurement units.

YoaOset If RefCSys = X'01', this parameter specifies the offset along the Y axis, Y_{pg} or Y_{ol}, of the including page or overlay coordinate system to the origin of the Y axis, Y_{oa}, of the object area coordinate system. The value for this parameter is expressed in terms of the number of page or overlay coordinate system Y-axis measurement units. If the referenced object specifies an Object Environment Group (OEG), this parameter overrides the corresponding parameter in the Object Area Position (OBP) structured field of the OEG. If the object is a page segment, this parameter overrides the corresponding OBP parameters in the environment groups of all objects that comprise the page segment and specifies the object area offsets from the page or overlay origin for all data objects in the page segment. A value of X'FFFFFF' indicates that the

Y-axis offset specified in the object's OEG is to be used; therefore the offset value (-1) is excluded from the allowed range. If the object does not specify the Y-axis offset in an OEG, the architected default is X'000000'.

If RefCSys = X'00', specifies the offset along the B axis to the B-position of the current LND, RCD, or XMD. The value for the parameter in this case is expressed in terms of the number of B-axis measurement units.

XoaOrent If RefCSys = X'01', this parameter specifies the amount of clockwise rotation of the object area's X axis, X_{oa}, about its defined origin relative to the X axis of the page or overlay coordinate system. If the referenced object specifies an Object Environment Group (OEG), this parameter overrides the corresponding parameter in the Object Area Position (OBP) structured field of the OEG. If the object is a page segment, this parameter overrides the corresponding OBP parameters in the environment groups of all objects that comprise the page segment. A value of X'FFFF' indicates that the X-axis rotation specified in the object's OEG is to be used. If the object does not specify the X-axis rotation in an OEG, the architected default is X'0000' (0 degrees).

If RefCSys = X'00', this parameter specifies the amount of clockwise rotation of the object area's X axis, X_{oa}, about its defined origin relative to the I axis of the current LND, RCD, or XMD (I,B) coordinate system.

YoaOrent If RefCSys = X'01', this parameter specifies the amount of clockwise rotation of the object area's Y axis, Y_{oa}, about its defined origin relative to the X axis of the page or overlay coordinate system. The YoaOrent value must be 90 degrees greater than the XoaOrent value or a X'01' exception condition exists. If the referenced object specifies an Object Environment Group (OEG), this parameter overrides the corresponding parameter in the Object Area Position (OBP) structured field of the OEG. If the object is a page segment, this parameter overrides the corresponding OBP parameters in the environment groups of all objects that comprise the page segment. A value of X'FFFF' indicates that the Y-axis rotation specified in the object's OEG is to be used. If the object does not specify the Y-axis rotation in an OEG, the architected default is X'2D00' (90 degrees).

If RefCSys = X'00', specifies the amount of clockwise rotation of the object area's Y axis, Y_{oa}, about its defined origin relative to the I axis of the current LND, RCD, or XMD (I,B) coordinate system.

Note: The following combinations of values are the only ones valid for the XoaOrent and YoaOrent parameters:

XoaOrent	YoaOrent	Description
X'0000'	X'2D00'	0 and 90 degrees respectively
X'2D00'	X'5A00'	90 and 180 degrees respectively
X'5A00'	X'8700'	180 and 270 degrees respectively
X'8700'	X'0000'	270 and 0 degrees respectively

RefCSys Determines how the object is positioned and rotated on the page:

Value	Description
X'00'	The object area offset in the IOB is measured with respect to the current LND, RCD, or XMD position using the current text (I,B) coordinate system. The object area rotation in the IOB is measured with respect to the current text (I,B) coordinate system I-axis. In this case, the object area offset and rotation parameters must be specified explicitly, that is, a value of X'FF...FF', which indicates that the value in the object's OEG is to be used, is not supported for {XoaOset,YoaOset} and {XoaOrent,YoaOrent} when RefCSys = X'00'.

X'01' The object area offset in the IOB is measured with respect to the page origin (X_p=0, Y_p=0) using the page (X_p,Y_p) coordinate system. The object area rotation in the IOB is measured with respect to the page (X_p,Y_p) coordinate system X_p-axis.

When line data with IOBs is transformed into MO:DCA data, the IOBs are generated in the MO:DCA data as well. If an IOB specifies RefCSys=X'00', the position and orientation must be modified for the MO:DCA IOB to specify the equivalent position and orientation based on the page (X_p,Y_p) coordinate system.

IOB Triplets

Extended Resource Local Identifier (X'22') Triplet

The Extended Resource Local Identifier (X'22') triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

This triplet is mandatory when the IOB structured field is specified in a PageDef, in which case it must occur once. If this triplet is specified more than once, only the first is used. It specifies a local identifier for the IOB that is used to reference the IOB from one or more LND, RCD, or XMD structured fields. The ID specified for each IOB must be unique within the PageDef.

Triplet X'22' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	7	Length of the triplet, including Tlength	M
1	CODE	Tid	X'22'	Identifies the Extended Resource Local Identifier triplet	M
2	CODE	ResType	 X'30'	Specifies the resource type: IOB Reference	M
3–6	CODE	ResLID	X'00000000'– X'FFFFFFFF'	Specifies the extended resource local ID	M

Triplet X'22' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Extended Resource Local Identifier triplet

ResType Specifies the resource type associated with the extended local ID

Value	Description
X'30'	IOB Reference The local identifier (LID) specified by this triplet is assigned to an IOB structured field and is used by an LND, RCD, or XMD to reference the IOB that specifies this ID. This value is only used when the triplet occurs in a Page Definition in AFP line-data environments.
All others	Reserved

ResLID Specifies a unique resource object Local ID

It may be in the range of X'00000000' to X'FFFFFFFF'.

Include Page Overlay (IPO)

Note: The IPO is a MO:DCA structured field. The following description documents its use in line-mode and mixed mode applications and introduces parameter values that are only valid in these environments. For the formal definition of the IPO structured field, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

The Include Page Overlay structured field references an overlay resource object that is to be positioned on the page. The overlay contains its own Active Environment Group. For line-mode and mixed-mode applications only, a value of X'FFFFFF' may be used for either the X axis offset (bytes 8–10), the Y axis offset (bytes 11–13), or both.

IPO (X'D3AFD8') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3AFD8'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	OvlyName		Name of the overlay resource	M
8–10	SBIN	XolOset	-32,768 – +32,767	X-axis origin for the page overlay	M
			X'FFFFFF'	Use X-axis position specified by current LND or RCD	
11–13	SBIN	YolOset	-32,768 – +32,767	Y-axis origin for the page overlay	M
			X'FFFFFF'	Use Y-axis position specified by current LND or RCD	
14–15	CODE	OvlyOrent	 X'0000' X'2D00' X'5A00' X'8700'	The overlay's X-axis rotation from the X_p axis of the page: 0 degrees 90 degrees 180 degrees 270 degrees	O
16–*n*		Triplets		See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for triplet applicability.	O

IPO Semantics

OvlyName
Token name of the overlay being referenced

If the first two characters of the overlay name are `O1` (capital letter O followed by the number 1), then bytes 0 and 1 must contain the characters O and 1, respectively.

XolOset
Offset along the X_p axis from the page origin where the origin of the overlay is placed

The value for this offset is expressed in terms of the measurement units currently in effect for the active Data Map. A value of X'FFFFFF' indicates that the overlay is to be placed at the X_p axis point specified by the current LND or RCD; therefore the offset value (-1) is excluded from the allowed range.

YolOset
Offset along the Y_p axis from the page origin where the origin of the overlay is placed

The value for this offset is expressed in terms of the measurement units currently in effect for the active Data Map. A value of X'FFFFFF' indicates that the overlay is to be placed at the Y_p axis point specified by the current LND or RCD; therefore the offset value (-1) is excluded from the allowed range.

OvlyOrent Overlay orientation

Specifies the amount of rotation of the page overlay's X axis about the page overlay origin relative to the X_p axis of the page. The page overlay X axis rotation is limited to 0, 90, 180, and 270 degrees. The page overlay Y axis rotation is always 90 degrees greater than the page overlay X axis rotation.

If no value is specified for this parameter, the architected default is 0 degrees. Note that the 90°, 180°, 270° rotations of a page overlay are not supported in all AFP environments. Consult the product documentation to see which rotations are supported. Also note that the MO:DCA IS/1 and IS/2 interchange sets only support 0° rotation of a page overlay.

Include Page Segment (IPS)

Note: The IPS is a MO:DCA structured field. The following description documents its use in line-mode and mixed mode applications and introduces parameter values that are only valid in these environments. For the formal definition of the IPS structured field, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

The Include Page Segment structured field references a page segment resource object that is to be positioned on the page or overlay. For line-mode or mixed-mode applications only, a value of X'FFFFFF' may be used for either the I-axis offset (bytes 8–10), the B-axis offset (bytes 11–13), or both.

IPS (X'D3AF5F') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3AF5F'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–7	CHAR	PsegName		Name of the page segment resource	M
8–10	SBIN	IpsOset	-32,768 – +32,767	I-axis origin for the page segment	M
			X'FFFFFF'	Use I-axis position specified by current LND or RCD	
11–13	SBIN	BpsOset	-32,768 – +32,767	B-axis origin for the page segment	M
			X'FFFFFF'	Use B-axis position specified by current LND or RCD	
14–*n*		Triplets		See the *Mixed Object Document Content Architecture (MO:DCA) Reference* for triplet applicability.	O

IPS Semantics

PsegName Token name of the page segment being referenced

All eight bytes of the name must be specified.

IpsOset Offset along the I axis from the current text coordinate system origin (I=0, B=0) to the origin of the page segment

The value for this offset is expressed in terms of the measurement units currently in effect for the active Data Map and is measured using the current text (I,B) coordinate system. A value of X'FFFFFF' indicates that the page segment origin is to be placed at the I-axis point specified by the current LND or RCD; therefore the offset value (-1) is excluded from the allowed range.

BpsOset Offset along the B axis from the current text coordinate system origin (I=0, B=0) to the origin of the page segment

The value for this offset is expressed in terms of the measurement units currently in effect for the active Data Map and is measured using the current text (I,B) coordinate system. A value of X'FFFFFF' indicates that the page segment origin is to be placed at the B-axis point specified by the current LND or RCD; therefore the offset value (-1) is excluded from the allowed range.

Note: The MO:DCA Line Data Object Position Migration (X'27') triplet may be specified on the IPS in MO:DCA documents to capture the text orientation that was specified when the page segment referenced by the IPS was included in line data. The information in this triplet allows the page segment and its objects to be positioned and oriented correctly on the MO:DCA page.

Line Descriptor Count (LNC)

The Line Descriptor Count structured field specifies the number of Line Descriptor (LND), Record Descriptor (RCD) or XML Descriptor (XMD) structured fields in the Data Map Transmission Subcase.

LNC (X'D3AAE7') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3AAE7'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–1	UBIN	NumDSC	1–65,535	Number of LND, RCD, or XMD structured fields in the Data Map Transmission Subcase	M

LNC Semantics

NumDSC Number of LND, RCD, or XMD structured fields in the Data Map Transmission Subcase

Note: The LND, RCD, or XMD in a Data Map are numbered sequentially, starting with 1. Values from 1 through the number of LND, RCD, or XMD are allowed.

Line Descriptor (LND)

The Line Descriptor structured field contains information, such as line position, text orientation, font selection, field selection, and conditional processing identification, used to format line data.

Note: The LNDs in a Data Map are numbered sequentially, starting with 1. Values from 1 through the number of LNDs are allowed.

LND (X'D3A6E7') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A6E7'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–1	BITS	LNDFlgs			M
Bit 0			B'0'–B'1'	End Page if Skipping	
Bit 1			B'0'–B'1'	End Page if Spacing	
Bit 2			B'0'–B'1'	Generate Inline Position	
Bit 3			B'0'–B'1'	Generate Baseline Position	
Bit 4			B'0'–B'1'	Generate Font Change	
Bit 5			B'0'–B'1'	Generate Suppression	
Bit 6			B'0'–B'1'	Reuse Record	
Bit 7			B'0'–B'1'	Use Fixed Data	
Bit 8			B'0'	Reserved; should be zero	
Bit 9			B'0'–B'1'	Use Compatibility TRC	
Bit 10			B'0'–B'1'	Set Text Color	
Bit 11			B'0'–B'1'	Conditional Processing	
Bit 12			B'0'–B'1'	Resource Object Include	
Bit 13			B'0'–B'1'	Relative Baseline Position	
Bits 14–15			B'00'	Reserved; should be zero	
2–3	UBIN	IPos	0 to page extent minus 1	Inline Position	M
4–5	UBIN	BPos	0 to page extent minus 1	Absolute baseline position	M
	SBIN		X'8000'–X'7FFF'	Relative baseline position	
6–9	CODE	TxtOrent		Text (I,B) Orientation:	M
			X'0000 2D00'	0,90 degrees	
			X'2D00 5A00'	90,180 degrees	
			X'5A00 8700'	180,270 degrees	
			X'8700 0000'	270,360 degrees	
10	CODE	FntLID	X'01'–X'7F'	Primary font local ID	M
			X'FF'	Presentation system default font	
11	CODE	ChnlCde	X'00'–X'0C'	Channel code	M
12–13	UBIN	NLNDskp	X'0001'–X'FFFF'	Next LND if skipping	M
14–15	UBIN	NLNDsp	X'0001'–X'FFFF'	Next LND if spacing	M
16–17	UBIN	NLNDreu	X'0001'–X'FFFF'	Next LND if reusing data	M
18–25	CHAR	SupName		Suppression token name A value of X'FF...FF' (null value) is not valid.	M

Offset	Type	Name	Range	Meaning	M/O
26	CODE	SOLid	X'01'–X'7F'	Shift-out font local ID	M
			X'00'	Not specified	
27–30	UBIN	DataStrt	X'00000000'–X'00007FFF'	Data start position	M
31–32	UBIN	DataLgth	X'0000'–X'FFFE'	Data length	M
			X'FFFF'	Place remaining bytes	
33–34	CODE	TxtColor	See LND Semantics.	Text color value	M
35–36	UBIN	NLNDccp	X'0000'–X'FFFF'	Next LND if conditional processing	M
37	CODE	SubpgID	X'00'–X'FF'	Subpage ID	M
38–39	CODE	CCPID	X'0000'–X'FFFF'	CCP Identifier	M
40–*n*		Triplets		See LND Semantics for triplet applicability.	O

Architecture Note: Prior to the addition of color and conditional processing to the page definition, bytes 33 through 39 were not defined as part of the LND structured field. Older page definition generators continued to generate the shorter version of the LND for some time after conditional processing and color were added to the architecture. Some of these page definitions still exist and should be handled by page definition processors. The valid length of the LND without the color and conditional processing fields was 33 bytes. Processors of page definitions may safely assume the value of each missing field is zero. LND triplets are not allowed on the shorter version of the LND.

LND Semantics

LNDFlgs LND flags

Bit 0 End Page if Skipping

This bit shows whether the page ends if the control character is set for skipping. This flag is processed if the channel code of the LND does not match the skip-to channel code.

Value	**Description**
B'0'	The current page does not end.
B'1'	The next data should be on a new page at the print position specified by the first LND with the indicated channel code. If the channel code is not found but the End Page if Skipping is set, the search for the channel code continues with the first LND in the Data Map.

Bit 1 End Page if Spacing

This bit shows whether the page ends if the control character is set for spacing.

Value	**Description**
B'0'	The current page does not end.
B'1'	The next data should be on a new page at the print position indicated by the first LND in the Data Map.

Bit 2 Generate Inline Position

This bit shows whether the data processed using this LND is placed on the page at the inline position specified in bytes 2–3. This position becomes the new print position.

Value **Description**

B'0' The current inline position is used.

B'1' Bytes 2–3 are used for the new inline position.

Bit 3 Generate Baseline Position

This bit shows whether the data processed using this LND is placed on the page at the baseline position specified in bytes 4–5. The baseline position may be an absolute position or a relative position, as specified by bit 13. This position becomes the new print position.

Value **Description**

B'0' The current baseline position is used.

B'1' Bytes 4–5 and bit 13 are used to generate either an absolute baseline position with respect to the current text (I,B) coordinate system origin or a relative baseline position with respect to a previously defined baseline position.

Bit 4 Generate Font Change

This bit shows whether the data processed using this LND is printed using the Font Local Identifier specified in byte 10.

Value **Description**

B'0' The following rules apply:

- If the record to be processed contains a TRC, use the font corresponding to the TRC.
- If the current Data Map maps fonts with MCF or MDR structured fields, use the first font in the font list.
- If the current Data Map does not map fonts, use the hardware default font.

B'1' The font specified in byte 10 is used.

Bit 5 Generate Suppression

This bit shows whether the data processed using this LND is suppressible text.

Value **Description**

B'0' The data is not suppressible text.

B'1' The data is suppressible text.

If this LND is used to present the selected data as a bar code, this flag is ignored. Suppression is only supported for text.

Note: Refer to the *Presentation Text Object Content Architecture Reference* for more information on text suppression.

Bit 6 Reuse Record

This bit shows whether the line data processed by this LND should be reused and processed by the LND specified in bytes 16–17 (NLNDreu).

Value **Description**

B'0' The record is not reused.

B'1' The record is reused.

Bit 7 Use Fixed Data

This bit shows whether to present text from the Fixed Data Text (FDX) structured fields.

Value	Description
B'0'	Fixed data is not presented.
B'1'	Fixed data is presented.

Bytes 27–30 (DataStrt) specify the start of the text to be added, and bytes 31–32 (DataLgth) specify the number of data bytes.

Note: No data from the current record is placed within the page under the control of this LND.

Bit 8 Reserved; should be zero

Bit 9 Use Compatibility TRC

This bit indicates whether the compatibility TRC should be used. The compatibility TRC uses only the last 4 bits, but the noncompatibility TRC uses all 8 bits.

Value	Description
B'0'	The compatibility TRC is not used. Valid TRC values are from 0 through 126. If a TRC is specified that is beyond the number of fonts mapped, use the first font in the MCF.
B'1'	The compatibility TRC is used. Only the first four fonts specified are used when this bit is set. If the TRC value is greater than the number of fonts specified, use the first font.

Bit 10 Set Text Color

If this bit is B'1', it specifies that the line data processed by this LND is to be presented in the color specified by the Color Specification (X'4E') triplet or by the value specified in bytes 33–34. The X'4E' triplet, if specified, takes precedence over the value in bytes 33–34; however, the value in bytes 33–34 may be used instead of the X'4E' color if the presentation device does not support the PTOCA PT3 subset. If this bit is B'0', the line data processed by this LND is presented in the presentation process default color.

Bit 11 Conditional Processing

This bit indicates whether conditional processing should be performed on the current line data record.

Value	Description
B'0'	Conditional processing is not performed.
B'1'	Conditional processing is performed.

If this bit is B'1', the LND is referred to as a *conditional processing LND* and is only used to specify data to be compared, not to place data on the page.

Bit 12 Resource Object Include

This bit indicates whether this LND specifies by name resource objects to be included on the page at a specified position.

Value	Description
B'0'	Named resource objects are not included.
B'1'	Named resource objects identified in the Resource Object Include (X'6C') triplet on this LND are included.

Bit 13 Relative Baseline Position

This bit indicates whether the baseline position specified in bytes 4–5 of this LND is an absolute position or a relative position. If an absolute baseline position is specified, it is measured as a positive offset in the baseline direction from the current text (I,B) coordinate system origin. If a relative position is specified, it is measured as a positive or negative offset from a previous baseline position using the current text (I,B) coordinate system, which is defined by the text orientation specified in bytes 6–9.

Value	Description
B'0'	The baseline position specified in bytes 4–5 is an absolute position.
B'1'	The baseline position specified in bytes 4–5 is a relative position.

The following restriction applies to relative baseline positioning:

- The text orientation of an LND that specifies relative baseline positioning must be the same as the text orientation of the LND that defines the baseline position from which the relative offset is measured.

Bits 14–15
Reserved; should be zero

IPos Inline Position

This field specifies the inline position of data, specified as an offset from the current text (I,B) coordinate system origin, which is defined by the text orientation specified in bytes 6–9 (TxtOrent). The offset is measured using the measurement units specified for the page in the PGD. If bit 2 of byte 0 is B'1', it is used. This position becomes the new print position.

Note: Data must not exceed the boundaries of the page, which are defined in the Page Descriptor (PGD) structured field. If the new print position is outside these boundaries, printing of the page stops.

BPos Baseline Position

This field specifies the baseline position of data. If bit 13 specifies absolute baseline position, these bytes specify a positive offset in the baseline direction from the current text (I,B) coordinate system origin. If bit 13 specifies a relative baseline position, these bytes specify a positive or negative offset from a previous baseline position using the current text (I,B) coordinate system, which is defined by the text orientation in bytes 6–9. The baseline position used as a reference for the relative offset depends on whether the LND that specifies relative positioning is a base LND and on whether a page or subpage boundary was crossed since the last LND was used to print. For a definition of base LNDs see “Field Formatting—LND Processing” on page 29. The baseline position used as a reference for the relative offset is determined as follows:

- For base LNDs, offsets are defined relative to the last base LND processed, that is, the last LND used to print or the last LND processed for spacing. However, if a page or subpage boundary was crossed after the last base LND was processed, offsets are defined relative to the first LND for the page or subpage.
- For reuse LNDs other than base LNDs, the offset is defined relative to the last LND that was processed for print (whether or not data prints).
- If the first LND of a Data Map specifies relative positioning, its offset is defined relative to the current text coordinate system origin (I=0,B=0), using the current text (I,B) coordinate system.
- If the first LND of a subpage specifies relative positioning, its offset is defined relative to the last base LND used to print, using the current text (I,B) coordinate system. Note that when skipping into a subpage, if the skipped-to LND specifies relative positioning, the relative offset is measured with respect to the first LND of the subpage, which might specify a relative position as well. This function allows a subpage to “float” relative to the last print position.

The offset is measured using the measurement units specified for the page in the PGD. If bit 3 of byte 0 is B'1', this position is used and becomes the new print position.

Application Note: When relative baseline positioning is used, the PageDef generator cannot check for off-page errors, since the data normally determines, with skip-to-channel carriage controls, when the relative baseline LNDs are invoked. AFP print servers generate a page break if the active Data Map is about to position data past the page's y-extent. This does not cause the generation of an error message. Note that the page's y-extent is specified in the PGD of the Data Map.

TxtOrent Text (I,B) Orientation

The four valid text orientations are:

Orientation	Value
0°,90°	X'0000 2D00'
90°,180°	X'2D00 5A00'
180°,270°	X'5A00 8700'
270°,0°	X'8700 0000'

Note that a change in text orientation means a change in the origin of the text (I,B) coordinate system:

Orientation	Value
0°,90°	Top-left corner of page
90°,180°	Top-right corner of page
180°,270°	Bottom-right corner of page
270°,0°	Bottom-left corner of page

If relative Baseline positioning is specified in bit 13, this text orientation must be the same as the text orientation of the LND that defines the baseline position from which the relative offset is measured.

FntLID Primary Font Local Identifier

If bit 4 of byte 0 is B'1', this is the local identifier of the font for text processed by this LND. When in shift-out/shift-in (SOSI) processing mode, this is also the local identifier of the font to be used following the implicit shift-in at the start of a record and, when using record-based SOSI font selection, it is also the font Local identifier to be used following an explicit shift-in in a record. A null value (X'FF') indicates that a presentation system default font is to be used.

Note: The Map Coded Font or Map Data Resource structured field in the AEG for the Data Map must have mapped the local identifier to the name of a font.

ChnlCde Channel Code

This field specifies the channel code of this LND. The value X'00' indicates that this LND has no channel code.

NLNDskp Next Line Descriptor if Skipping

If the record indicates skipping, this parameter points to the LND to use in scanning for the channel code indicated by the control character of the record. If in following the Next Line Descriptor if Skipping chain, the channel code is not found but the End Page if Skipping bit is set, the search for the channel code continues with the first LND in the Data Map.

The LNDs in a Data Map are numbered sequentially, starting with 1. Values from 1 through the number of LNDs are allowed.

NLNDsp Next Line Descriptor if Spacing

If the record indicates spacing, this parameter points to the LND to use next. A chain of Next Line Descriptor if Spacing values is followed until the number of entries followed equals the number of spaces desired.

The LNDs in a Data Map are numbered sequentially, starting with 1. Values from 1 through the number of LNDs are allowed.

NLNDreu Next Line Descriptor if Reusing Data

This parameter points to the LND used to continue processing the record when reusing data.

If bit 6 of byte 0 = B'1', this parameter points to the next LND in a chain to process the same data record. The end of the chain is indicated by bit 6 of byte 0 = B'0' and a value of X'0000' in this parameter. At the end of the chain, control returns to the base LND. For example, if single spacing is specified and the data does not contain a skip-to-channel carriage control, processing resumes with the LND following the base LND.

When machine carriage controls or no carriage controls are specified, the LND that started the reuse chain is used with the current data record to determine the next LND to use. When ANSI carriage controls are specified, the LND that started the reuse chain is used with the next data record to determine the next LND to use.

The LNDs in a Data Map are numbered sequentially, starting with 1. Values from 1 through the number of LNDs are allowed. The chain of LNDs that is traversed when re-using a record does not have to follow any numerical order.

SupName Suppression token name

This field specifies the suppression to be used with this LND. This value must match bytes 0–7 in one of the repeating groups in the Map Suppression (MSU) structured field of the Form Definition. The value can be any 8-byte value except null.

This parameter is ignored if bit 5, byte 0 is B'0'.

Note that suppression is only supported for text. If the data selected by this LND is presented as a bar code, this parameter is ignored. Note also that if text suppression is activated, only the field or record processed by this LND is suppressed, not text data that may be included using a Resource Object Include (X'6C') triplet.

SOLid Shift-out Font Local Identifier

If this byte is non-zero, it specifies the local identifier of the font to be used when a shift-out control is encountered in the record processed by this LND. Use of this parameter signals record-based font selection for SOSI processing. If this byte is zero, the parameter is not specified.

Note: When processing line data in shift-out/shift-in (SOSI) mode, an implicit shift-in is assumed at the start of every record.

DataStrt Data Start Position

This field specifies the offset of the first data byte to be processed by this LND. If bit 7 of byte 0 is B'1', the data from the Fixed Data Text (FDX) structured field is used. Otherwise, the data from the current record is used.

A value of 0 indicates that the first byte is to be used. No data is placed if this value is greater than the length of the data source.

If conditional processing is to be performed (bit 11 of byte 1 is B'1'), this parameter defines the start of the data to be compared, known as the Comparison field. The comparison is determined by the Conditional Processing Control (CCP) structured field identified in bytes 38–39 (CCPID).

DataLgth Data Length

This field specifies the number of bytes of data to be processed by this LND. If bit 7 of byte 0 is B'1', the data from the Fixed Data Text (FDX) structured field is used. Otherwise, the data from the current record is used.

If this value is X'FFFF', all the remaining data bytes are processed.

If this parameter causes data to be positioned outside the boundaries of the page, which are defined in the Page Descriptor (PGD) structured field, the printing of the page stops.

If conditional processing is to be performed (bit 11 of byte 1 is B'1'), this parameter defines the length of the comparison field. The comparison is determined by the Conditional Processing Control (CCP) structured field identified in bytes 38–39 (CCPID).

TxtColor Text Color Value

The specified color is used to present text processed by this LND when LND byte 0, bit 10= B'1' and a Color Specification (X'4E') triplet is not specified for this LND. This color may also be used if the X'4E' triplet is specified but the PTOCA PT3 subset is not supported by the presentation device. Color values are defined as shown in Table 14 on page 105. They reflect the range of values defined in the Standard OCA Color Value Table. For a definition of the Standard OCA Color Value Table, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*. RGB values are also defined for each color, assuming that the intensity range for each component is 0–255.

Table 14. Color-Value Table

Value	Color	Red (R)	Green (G)	Blue (B)
X'0000' or X'FF00'	Device default			
X'0001' or X'FF01'	Blue	0	0	255
X'0002' or X'FF02'	Red	255	0	0
X'0003' or X'FF03'	Pink/magenta	255	0	255
X'0004' or X'FF04'	Green	0	255	0
X'0005' or X'FF05'	Turquoise/cyan	0	255	255
X'0006' or X'FF06'	Yellow	255	255	0
X'0007'	White; see note 1	255	255	255
X'0008'	Black	0	0	0
X'0009'	Dark blue	0	0	170
X'000A'	Orange	255	128	0
X'000B'	Purple	170	0	170
X'000C'	Dark green	0	146	0
X'000D'	Dark turquoise	0	146	170
X'000E'	Mustard	196	160	32
X'000F'	Gray	131	131	131
X'0010'	Brown	144	48	0
X'FF07'	Device default	—	—	—
X'FF08'	Color of medium	—	—	—
All others	Reserved	—	—	—

Notes:

1. The color rendered on presentation devices that do not support white is device-dependent. For example, some printers simulate white with the color of the medium, which results in white if a white medium is used.
2. The value X'FFFF' is supported for migration purposes only and specifies the presentation process default.

NLNDccp Next Line Descriptor if Conditional Processing

A non-zero value in this parameter means that conditional processing is to be performed on the current line data record. This parameter points to the LND used to perform conditional processing on the current input line data record. The target LND, called a *conditional processing LND*, must specify flag bit 11=B'1' and must point to a CCP structured field that defines the conditional processing. The conditional processing LND defines the data to be compared and does not place data on the page. Note that a conditional processing LND may point to another conditional processing LND.

SubpgID Subpage Identifier

For conditional processing, a page can be divided into subpages to specify boundaries on which conditional processing actions can be taken. All LND structured fields of one subpage should have the same value in their Subpage Identifier byte, differing from the value in the LND structured fields of neighboring subpages. See Chapter 3, “Using a Page Definition to Print Data”, on page 15 for more information on conditional processing and subpages.

The presentation services program detects the beginning and end of subpages by checking for a change in the value of byte 37 (SubpgID). If conditional processing is to be performed (bit 11 of byte 1 is B'1'), this parameter is used to identify subpages when the timing of the conditional action is relative to the subpage.

CCPID CCP Identifier

If bit 11 of byte 1 is B'1' this parameter is used to locate the associated Conditional Processing Control (CCP) structured field that describes the conditional processing to be performed for this Page Definition.

Triplets Optional triplets

See the following for detailed information about the triplets permitted on the LND structured field:

“Fully Qualified Name (X'02') Triplet” on page 107
“Extended Resource Local Identifier (X'22') Triplet” on page 108
“Color Specification (X'4E') Triplet” on page 109
“Bar Code Symbol Descriptor (X'69') Triplet” on page 110
“Resource Object Include (X'6C') Triplet” on page 116
“Additional Bar Code Parameters (X'7B') Triplet” on page 118
“Object Reference Qualifier (X'89') Triplet” on page 119
“Color Management Resource Descriptor (X'91') Triplet” on page 121
“Concatenate Bar Code Data (X'93') Triplet” on page 122

LND Triplets

Fully Qualified Name (X'02') Triplet

The Fully Qualified Name (X'02') triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

This triplet is optional and may occur one or more times when a Bar Code Symbol Descriptor (X'69') triplet is specified on the LND, RCD, or XMD. The Fully Qualified Name type that may appear is X'DE'- *Data Object External Resource Reference*. The FQN triplet specifies the external identifier of a Color Management Resource (CMR) object that is used for the Bar Code object being generated. The identifier is used by the presentation system to locate the resource object in the resource hierarchy. The identifier is a character-encoded name that must be specified using FQNFmt = X'00'. The encoding for the external identifier of the CMR must be UTF-16BE.

Triplet X'02' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	150	Length of the triplet, including Tlength	M
1	CODE	Tid	X'02'	Identifies the Fully Qualified Name triplet	M
2	CODE	FQNType	X'DE'	Data Object External Resource Reference	M
3	CODE	FQNFmt	X'00'	GID format is character string	M
4–149		FQName		GID of the CMR Must be 146 bytes in length and encoded using UTF-16BE	M

Triplet X'02' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Fully Qualified Name triplet

FQNType Specifies how the fully qualified name is to be used

Value	Description
X'DE'	This GID specifies the name of a Color Management Resource to use when generating the Bar Code object.
All others	Reserved

FQNFmt Specifies the format of the Global Identifier

Value	Description
X'00'	The GID is a character-encoded name that means the data type is CHAR.
All others	Reserved

FQName Contains the Global Identifier (GID) to be used as the name of the CMR

The encoding for the external identifier of the CMR must be UTF-16BE.

Extended Resource Local Identifier (X'22') Triplet

The Extended Resource Local Identifier (X'22') triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

This triplet is optional and may occur one or more times to reference an IOB structured field in the PageDef. The reference consists of a local identifier that must match the local identifier on an IOB in the PageDef. When an IOB with matching ID is found, the IOB is processed to present the object on the page. If an IOB with matching ID is not found, an exception is generated.

Triplet X'22' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	7	Length of the triplet, including Tlength	M
1	CODE	Tid	X'22'	Identifies the Extended Resource Local Identifier triplet	M
2	CODE	ResType	 X'30'	Resource type: IOB Reference	M
3–6	CODE	ResLID	X'00000000'– X'FFFFFFFF'	Specifies the extended resource local ID	M

Triplet X'22' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Extended Resource Local Identifier triplet

ResType Specifies the resource type associated with the extended local ID

Value	Description
X'30'	IOB Reference The local identifier (LID) specified by this triplet is assigned to an IOB structured field and is used by an LND, RCD, or XMD to reference the IOB that specifies this ID. This value is only used when the triplet occurs in a Page Definition in AFP line-data environments.
All others	Reserved

ResLID Specifies a unique resource object Local ID

The range for the Local ID is X'00000000' to X'FFFFFFFF'.

Note: RefCSys parameter in IOB byte 26 determines how the object included with the IOB is positioned and rotated on the page:

Value	Description
X'00'	The object area offset in the IOB is measured with respect to the current LND, RCD, or XMD position using the current text (I,B) coordinate system. The object area rotation in the IOB is measured with respect to the current text (I,B) coordinate system I-axis.
X'01'	The object area offset in the IOB is measured with respect to the page origin (X_p=0,Y_p=0) using the page (X_p,Y_p) coordinate system. The object area rotation in the IOB is measured with respect to page (X_p,Y_p) coordinate system X_p-axis.

Color Specification (X'4E') Triplet

The Color Specification triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*. Support for this triplet is tied to PTOCA PT3 support.

This is an optional triplet that specifies a color for text processed by this LND when LND byte 0 bit 10=B'1' or when the selected data is to be presented as a bar code symbol. One Color Specification triplet may be specified on an LND. If this triplet is specified more than once, only the first is used. If this triplet is specified for text when LND byte 0 bit 10=B'0', it is ignored.

With this triplet a color is specified by selecting a color space, an encoding for the components of the color value in that space, and a color value. The color spaces supported are:

- RGB
- CMYK
- Highlight
- CIELAB
- Standard OCA color space

The selected encoding defines the number of bits used to specify each component. For example, with the RGB color space, one supported encoding is 8 bits per component, which maps the component intensity range 0 to 1 to the binary values 0 to 255. The color value specifies the color. With the RGB color space and an 8 bit per component encoding, the color value (255,255,255) specifies full intensity for each component, which defines the color white.

Bar Code Symbol Descriptor (X'69') Triplet

Architecture Note: The Bar Code Symbol Descriptor triplet is registered in the MO:DCA architecture as a private-use triplet since it is used only in the PageDef object, which is not a MO:DCA object.

This is an optional triplet and may occur once. If this triplet is specified more than once, only the first is used. When present, it specifies that the data selected by LND, RCD, or XMD parameters DataStrt and DataLgth or the data selected by the RCD or XMD parameter Fldno is to be presented as a bar code symbol. The data can be from the current record or it can be fixed data. The origin of the bar code symbol or the character reference point (see byte 4, bit 5, for details) on the page presentation space is specified by the LND, RCD, or XMD position parameters IPos and BPos, and the orientation of the bar code symbol with respect to the page presentation space X_p-axis is specified by the LND, RCD, or XMD text orientation parameter TxtOrent. Note that the suppression function that may be specified on an LND, RCD, or XMD is not supported when the record or field is presented as a bar code. Note also that when an LND that specifies bar code presentation is re-used because the carriage control in the line data record specifies suppress spacing, only the last bar code generated by this LND is presented. If this triplet is specified, LND, RCD, or XMD flag bit 5 — Generate Suppression and the SupName parameter are ignored.

This triplet is not supported on conditional processing LNDs, RCDs, or XMDs that is LNDs, RCDs, or XMDs that specify flag bit 11=B'1'; if present, it is ignored.

This triplet carries parameters defined by the BCOCA architecture, for more information see the *Bar Code Object Content Architecture Reference*. Not all presentation services programs support this triplet.

The data used for the bar code can be obtained from multiple fields using the Concatenate Bar Code Data (X'93') triplet.

Triplet X'69' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	4 or 18	Length of the triplet, including Tlength	M
1	CODE	Tid	X'69'	Identifies the Bar Code Symbol Descriptor triplet	M
2–3	CODE	DescID	X'0001'–X'FFFE'	Identifies a bar code symbol descriptor	M
4	BITS	SymbFlgs			O
Bit 0		HRI	B'0' B'1'	HRI is presented HRI is not presented	
Bits 1–2		HRIPos	B'00' B'01' B'10'	Default HRI below HRI above	
Bit 3		Astrsk	B'0' B'1'	Asterisk not presented Asterisk presented	
Bit 4			B'0'	Reserved; should be zero	
Bit 5		SuppSym	B'0' B'1'	Present bar code symbol Suppress presentation of symbol	
Bit 6		TrlBlk	B'0' B'1'	Do not suppress trailing blanks in data Suppress trailing blanks in data and select type and modifier	
Bit 7			B'0'	Reserved; should be zero	
5–6	UBIN	BCWdth	See BCOCA Reference.	Desired symbol width	O

Offset	Type	Name	Range	Meaning	M/O
7	CODE	BCType	See BCOCA Reference.	Bar code type	O
8	CODE	BCMod	See BCOCA Reference.	Bar code modifier	O
9	CODE	FntLID	X'00'–X'FE'	Font local identifier	O
			X'FF'	BCOCA default font	
10–11	CODE	Color	See BCOCA Reference.	Bar code color	O
12	UBIN	ModWdth	See BCOCA Reference.	Module width in mils	O
13–14	UBIN	ElmtHt	See BCOCA Reference.	Element Height in L-units	O
15	UBIN	Mult	See BCOCA Reference.	Height multiplier	O
16–17	UBIN	WE:NE	See BCOCA Reference.	Wide-to-narrow ratio	O

Triplet X'69' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Bar Code Symbol Descriptor triplet

DescID Specifies the ID of a bar code symbol descriptor

The descriptor is defined by bytes 4–17 of this triplet. If the ID matches the ID of a bar code symbol descriptor defined previously on an LND for this page, the previous descriptor is used regardless of whether bytes 4–17 are specified in the current descriptor. If the ID does not match a previously-defined ID, bytes 4–17 must be specified and define the bar code symbol descriptor that is to be identified with this ID and that is to be used to generate this symbol. The valid range for the ID is X'0001'–X'FFFE'.

For a given page, the presentation services program collects all bar code symbols that have the same bar code symbol descriptor ID and the same bar code symbol orientation and groups them into a single bar code object.

The origin for the bar code object presentation space that contains the symbols is one of the four corners of the page as determined by the text orientation specified in LND bytes 6–9 (TxtOrent). The bar code presentation space origin is therefore coincident with the current text coordinate system (I,B) origin.

For example, if the LND specifies a (90°,180°) text orientation, the origin of the bar code object presentation space is the top-right corner of the page and bar code symbols in this object are rotated 90° with respect to the page X_p axis.

The extents of the object presentation space are determined by the extents of the page presentation space. For example, if the origin of the object presentation space is the top-right corner of the page, the X-extent of the object presentation space is the Y-extent of the page and the Y-extent of the object presentation space is the X-extent of the page. The symbol origin offset from the object presentation space origin and from the current text (I,B) coordinate system origin is specified by LND bytes 2–5 (IPOS,BPOS).

The units of measure for the bar code object presentation space, used for determining symbol origin offsets, are the same as those defined on the page (X_p,Y_p) presentation space in the PGD structured field of the Active Environment Group (AEG) for the Data Map. The presentation services program also defines an object area presentation space for the object that is identical in size, position, and units of measure to the bar code presentation space, and that has the same rotation about the page X_p-axis as the bar code symbols in the object.

SymbFlgs These flags specify additional controls.

Bits 0–3 These bits have the same syntax and semantics as the corresponding bits in byte 0 of the Bar Code Symbol Data (BSA) structure defined by the BCOCA architecture.

Bit 4 Reserved; should be zero

Bit 5 Bar code symbol suppression

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

Value **Description**

B'0' The bar code symbol is presented.

B'1' Presentation of the bar code symbol is suppressed. This can be used to print just the HRI. If both bit 0 and bit 5 are B'1' or the bar code does not support HRI, nothing is presented for this bar code object.

When bit 5 = B'1', LND, RCD, or XMD parameters IPos and BPos specify the character reference point for the first character of the HRI.

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

Bit 6 This flag is defined for AFP Line Data. It identifies the desired method of handling trailing blanks in the bar code data; for some symbologies, the resulting data length is used to adjust the bar code type and modifier to match the resulting data length. The PageDef supports fixed-length fields for data that is to be bar encoded. Since some bar codes allow variable-length data, these fixed-length fields often are padded on the right with blanks; these blanks are often not intended to be included in the BCOCA object, particularly for a bar code type that does not allow blanks. This flag identifies how these trailing blanks should be handled when a BCOCA bar code object is built from the line data and PageDef information.

It is used as follows:

Value **Description**

B'0' Trailing blanks in the bar code data are not suppressed.

B'1' All trailing blanks in the bar code data are suppressed and the bar code type and modifier are adjusted to match the resulting data length.

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

- If the user specified an EAN bar code type (X'08', X'09', X'16', or X'17'), truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
2	X'16'—two-digit supplemental	X'00'
5	X'17'—five-digit supplemental	X'00'
7	X'08'—EAN-8	X'00'
12	X'09'—EAN-13	X'00'

Resulting Data Length	Bar Code Type	Bar Code Modifier
14	X'16'—two-digit supplemental	X'01'
17	X'17'—five-digit supplemental	X'01'
Any other value	Error	

- If the user specified a UPC bar code type (X'03', X'05', X'06', or X'07'), truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
2	X'06'—two-digit supplemental	X'00'
5	X'07'—five-digit supplemental	X'00'
10	X'05'—UPC version E	X'00'
11	X'03'—UPC version A	X'00'
12	X'06'—two-digit supplemental	X'02'
13	X'06'—two-digit supplemental	X'01'
15	X'07'—five-digit supplemental	X'02'
16	X'07'—five-digit supplemental	X'01'
Any other value	Error	

- If the user specified a POSTNET bar code type (X'18'), truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
5	X'18'—POSTNET	X'00'
9	X'18'—POSTNET	X'01'
11	X'18'—POSTNET	X'02' or X'04'
Any other value	X'18'—POSTNET	X'03'

Note: Since both the X'02' and X'04' modifiers have the same length, the processor of the PageDef cannot always determine which modifier is desired if the data is truncated to match. Therefore, the following rule is used to set the modifier:

– If the original modifier requested in the triplet is not X'04' and the resultant length of the field is 11 bytes, modifier X'02' is used as the modifier for the generated bar code.

- If the user specified an Intelligent Mail® Barcode type (X'22'), truncate the data and set the bar code type and modifier based on the resulting data length:

Resulting Data Length	Bar Code Type	Bar Code Modifier
20	X'22'—Intelligent Mail Barcode	X'00'
25	X'22'—Intelligent Mail Barcode	X'01'

Resulting Data Length	Bar Code Type	Bar Code Modifier
29	X'22'—Intelligent Mail Barcode	X'02'
31	X'22'—Intelligent Mail Barcode	X'03'
Any other value	error	

- If the user specified any other bar code type, use the user-specified bar code type and modifier.

Bit 7 Reserved; should be zero

BCWdth Specifies the desired symbol width for the entire bar code symbol in L-units (not including the quiet zone)

This parameter has the same syntax and semantics as bytes 10–11 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

Note: This is an optional parameter that is not supported by all BCOCA receivers; this parameter is ignored by products that do not support this function.

BCtype Indicates the type of bar code symbol to be generated

This parameter has the same syntax and semantics as byte 12 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

BCMod Gives additional processing information about the bar code symbol to be generated

This parameter has the same syntax and semantics as byte 13 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

FntLID Specifies the local ID of a font used to render the HRI and to provide the code point to bar code character mappings

This parameter has the same syntax and semantics as byte 14 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture. The value X'FF' specifies the BCOCA default font. Any other value needs to be mapped with a Map Coded Font (MCF) or Map Data Resource (MDR) structured field in the AEG of the Data Map.

Color Specifies the color in which the bars of the bar code symbol and the HRI are to be presented when a Color Specification (X'4E') triplet is not specified

This color may also be used if the X'4E' triplet is specified but extended colors for BCOCA is not supported by the presentation device. This parameter has the same syntax and semantics as bytes 15–16 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

ModWdth Specifies the width in mils (thousandths of an inch) of the smallest defined bar code element

This parameter has the same syntax and semantics as byte 17 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

ElmtHt Specifies the height in L-units along the Y_{bc} axis of the bar code presentation space

The units of measure for the L-units are defined in the PGD structured field of the Active Environment Group of the Data Map. This parameter has the same syntax and semantics as bytes 18–19 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

Mult Specifies a value that, when multiplied by the element height, yields the total bar and space height presented

This parameter has the same syntax and semantics as byte 20 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

WE:NE Specifies the ratio of the wide-element dimension to the narrow-element dimension when only two different size elements exist, that is, for a two-level bar code

This parameter has the same syntax and semantics as bytes 21–22 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture.

Note: The last 14 bytes (bytes 4–17) in this triplet are optional as a group. That is, either they are all specified or none are specified. If the descriptor ID is intended to match a previously-defined descriptor ID, these bytes should not be specified. When present, byte 4 is identical in syntax and semantics to byte 0 of the Bar Code Symbol Data (BSA) structure defined by the BCOCA architecture. Bytes 5–17 are identical in syntax and semantics to bytes 10–22 of the Bar Code Symbol Descriptor (BSD) defined by the BCOCA architecture, except for the font local ID parameter, which must be set to X'FF' in the triplet to specify the device default font.

Resource Object Include (X'6C') Triplet

This is an optional triplet that identifies an overlay or page segment object to be presented on the page at a specified position. Multiple Resource Object Include triplets may be specified on the same LND.

If the triplet identifies an overlay, the overlay name must be mapped with an MPO structured field in the AEG of the Data Map. If the triplet identifies a page segment, the page segment may be mapped in the AEG with an MPS structured field. If mapped, the page segment is downloaded and can be used multiple times (this is called a *hard* page segment). If it is not mapped, the page segment data is downloaded as part of the page (this is called a *soft* page segment).

This triplet is not supported on a conditional processing LND, that is, an LND that specifies flag bit 11=B'1'. If present, it is ignored.

Note: The Resource Object Include is a MO:DCA triplet. The following description documents its use in a Page Definition and introduces parameter values that are only valid in this environment. For the formal definition of this triplet in the MO:DCA architecture, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*. Note that when used on an LND, this triplet supports a negative range for the object origin offset. Only a positive range is supported when the triplet is used in MO:DCA data streams. Note also that when used on an LND, this triplet supports ObjType=X'5F' (page segment). This value is not supported in MO:DCA data streams. Not all presentation services programs support this triplet.

Triplet X'6C' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	17 or 19	Length of the triplet, including Tlength	M
1	CODE	Tid	X'6C'	Identifies the Resource Object Include triplet	M
2	CODE	ObjType	 X'DF' X'5F'	Object type: Overlay object Page Segment object	M
3–10	CHAR	ObjName		Name of the object	M
11–13	SBIN	IobjOset	-32,768 – +32,767	Relative I-axis offset	M
14–16	SBIN	BobjOset	-32,768 – +32,767	Relative B-axis offset	M
17–18	CODE	ObOrent	 X'0000' X'2D00' X'5A00' X'8700'	The overlay's X axis rotation relative to the X_p axis of the page: 0 degrees 90 degrees 180 degrees 270 degrees	O

Triplet X'6C' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Resource Object Include triplet

ObjType Specifies the object type

Value	Description
X'DF'	Overlay object
X'5F'	Page segment object

ObjName Specifies the object name

IobjOset Relative I-axis offset

Relative offset of the object origin along the I axis of the current text (I,B) coordinate system, measured from the current LND position using the page measurement units specified in the PGD.

BobjOset Relative B-axis offset

Relative offset of the object origin along the B axis of the current text (I,B) coordinate system, measured from the current LND position using the page measurement units specified in the PGD.

ObOrent Only supported for ObjType X'DF' = Overlay object

Specifies the amount of rotation, about the overlay origin, of the overlay's X_{ol} axis relative to the X_{pg} axis of the page. Valid values are the following:

Value	Description
X'0000'	0 degrees
X'2D00'	90 degrees
X'5A00'	180 degrees
X'8700'	270 degrees
All others	Reserved

The overlay Y axis rotation is always 90 degrees greater than the overlay X axis rotation.

Notes:

1. If this parameter is omitted, the architected default value for the overlay rotation is X'0000', zero degrees.
2. When a page segment is included with this triplet, the ObOrent parameter is ignored and the rotation of objects in the page segment is summarized in Table 10 on page 49.

Additional Bar Code Parameters (X'7B') Triplet

This is an optional triplet that specifies additional parameters for non-linear bar code symbologies (for example, 2D bar codes). This triplet may occur one or more times when a Bar Code Symbol Descriptor (X'69') triplet is specified. If this triplet is specified more than once, the data from each triplet is concatenated in the order it is received. If a X'69' triplet is not specified, the X'7B' triplet is ignored. If a X'7B' triplet is specified and the X'69' triplet selects a linear bar code symbol, the results are unpredictable.

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	4–254	Length of the triplet, including Tlength	M
1	CODE	Tid	X'7B'	Identifies the Additional Bar Code Parameters triplet	M
2				Reserved; should be zero	M
3–*n*	CODE	AddParm		Additional parameters for non-linear bar code symbols	M

Triplet X'7B' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Additional Bar Code Parameters triplet

AddParm Specifies additional parameters for non-linear bar code symbols

These parameters are specific to the particular symbology and may include parameters like symbol size (rows/columns) and processing mode.

Note: The data carried by a Bar Code Symbol Descriptor (X'69') triplet, with the exception of the SymbFlgs parameter, is used to build the Bar Code Data Descriptor (BDD) structured field for the resulting bar code object. The data carried by the Additional Bar Code Parameters triplet, along with the SymbFlgs parameter, the LND, RCD, or XMD position, and the LND, RCD, or XMD data, is used to build a Bar Code Data (BDA) structured field for the resulting bar code object. For a description of the contents of the Bar Code Data structured field, see the *Bar Code Object Content Architecture Reference*.

Object Reference Qualifier (X'89') Triplet

The Object Reference Qualifier (X'89') triplet is used to specify whether the name of an object is retrieved from the input data or retrieved using normal methods. If the name is to be retrieved from the input data, that name overrides any ObjName field and any Fully Qualified Name (type X'01') triplet that would normally be used to select an object. This triplet may occur once on an LND structured field.

When this triplet is specified, the following rules apply:

- If this triplet is specified more than once, only the first is used.
- This triplet applies to the first Resource Object Include (X'6C') triplet or Extended Resource Local Identifier (X'22') triplet that follows on the LND or RCD.
- If this triplet is not followed by either the Resource Object Include triplet or the Extended Resource Local Identifier triplet, then this triplet is ignored.
- The LND or RCD DataStrt/DataLgth fields or the RCD Fldno is used to select the name of the object. The object name retrieved from the input is not presented as data on the page.
  - If this triplet is followed by the Resource Object Include triplet, the ObjName parameter of the Resource Object Include triplet is ignored.
  - If this triplet is followed by the Extended Resource Local Identifier triplet, the ObjName parameter and the Fully Qualified Name (FQN) triplet using FQNType X'01' of the Include Object (IOB) structured field pointed to by the Extended Resource Local Identifier triplet are ignored.

Note: If the field of data specified by the DataStrt/DataLgth or Fldno parameters does not exist in the input record, then this Object Reference Qualifier (X'89') triplet and the Resource Object Include triplet or the Extended Resource Local Identifier triplet that it applies to are ignored.

This triplet is not supported on conditional processing LNDs or RCDs, that is LNDs or RCDs that specify flag bit 11=B'1' if present, it is ignored.

Triplet X'89' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	4	Length of the triplet, including Tlength	M
1	CODE	Tid	X'89'	Identifies the Object Reference Qualifier triplet	M
2				Reserved; should be zero	M
3	BITS	QualFlg	See Triplet X'89' Semantics for bit definitions.	Object reference qualifier flags	M

Triplet X'89' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Object Reference Qualifier triplet

QualFlg Specifies object reference qualifier flags, as follows:

Bit	Description
0	Object reference qualifier flags B'0' Do not use selected input data to override object name B'1' Use selected data to override object name
1–7	Reserved

Notes:

1. If this triplet is omitted, the architected default for QualFlg bit 0 is B'0'.
2. When the QualFlag bit 0 is B'1', the encoding of the object name obtained from the input data is platform dependent. It is the responsibility of the user to ensure that the encoding of the input data is correct for the platform being used.

Implementation Note: For interchange in AFP environments, the name retrieved from the input data by the LND or RCD:

- Must be no more than 8 characters long; if longer, some AFP print servers use only the first 8 characters.
- Must follow the naming conventions used in AFP environments; see **External Resource Naming Conventions** in the *Mixed Object Document Content Architecture (MO:DCA) Reference* for a description.
- Must not contain platform-dependent library names or path names.
- Must be encoded using EBCDIC if the resource is mapped in the active environment group.
- Must not use double-byte encoding.
- Must not contain any shift-in or shift-out characters.

Color Management Resource Descriptor (X'91') Triplet

The Color Management Resource Descriptor (X'91') triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

The Color Management Resource Descriptor triplet specifies the processing mode and scope for a Color Management Resource (CMR). This triplet is mandatory when the LND references a Color Management Resource (CMR) with the FQN type X'DE' triplet, in which case this triplet specifies the processing mode for the CMR and must occur once for each FQN type X'DE' specified. It is ignored in all other cases. This triplet must immediately follow the FQN type X'DE' triplet that specifies the CMR name.

Triplet X'91' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	5	Length of the triplet, including Tlength	M
1	CODE	Tid	X'91'	Identifies the Color Management Descriptor triplet	M
2				Reserved; should be zero	M
3	CODE	ProcMode	 X'01' X'02' X'03'	Specifies the processing mode for the CMR: Process the CMR as an audit CMR Process the CMR as an instruction CMR Process the CMR as a link CMR; valid only for Link DL CMRs	M
4	CODE	CMRScpe	X'01'	Scope of CMR is a data object in this page	M

Triplet X'91' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Color Management Descriptor triplet

ProcMode Specifies the processing mode for the CMR

Valid values are the following:

Value	Description
X'01'	This CMR describes processing that has been done to a document component; process the CMR as an audit CMR.
X'02'	This CMR describes processing that is to be done to a document component; process the CMR as an instruction CMR.
X'03'	This CMR defines a direct color conversion from an input color space to a device output color space; process the CMR as a link CMR. This processing mode is only valid for Link DL CMRs.
All others	Reserved

CMRScpe Specifies the scope of the CMR when used inside a document

Valid values are the following:

Value	Description
X'01'	The scope of the CMR is a data object in this page.
All others	Reserved

Concatenate Bar Code Data (X'93') Triplet

Architecture Note: The Concatenate Bar Code Data triplet is registered in the MO:DCA architecture as a private-use triplet since it is used only in the PageDef object, which is not a MO:DCA object.

This is an optional triplet and may occur once. If this triplet is specified more than once, only the first will be used. This triplet is only valid if the LND, RCD, or XMD also specifies a Bar Code Symbol Descriptor (X'69') triplet. If the X'69' triplet is not specified, this triplet is ignored. When this triplet is present, it specifies that the field of data selected by the LND, RCD, or XMD and the Bar Code Symbol Descriptor triplet is to be collected as part of a concatenation of fields. When complete, this concatenation will be used to generate one bar code symbol. Each field of a concatenation must specify this triplet. For RCD and XMD, completion of a given concatenated bar code symbol is defined as the end of the page or an RCD or XMD with the Start New Symbol flag (bit 0 of CBCFlgs) set in the X'93' triplet being reused. For LND, completion of a given concatenated bar code symbol is defined as the end of the page (the Start New Symbol flag (bit 0 of CBCFlgs) has no effect since an LND cannot be reused on a page).

When this triplet is used, the position parameters IPos and BPos of the LND, RCD, or XMD that selects the first segment of the concatenation will be used to position the concatenated symbol. The first segment is the segment selected with the lowest non-zero SegOrder value, or it is the first segment received when SegOrder is not specified. All other LND, RCD, or XMD position parameters will be ignored for the remaining segments.

This triplet is not supported on conditional processing LNDs, RCDs or XMDs (LNDs, RCDs or XMDs that specify flag bit 11=B'1'). If the triplet is present, it is ignored.

Triplet X'93' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	7	Length of the triplet, including Tlength	M
1	CODE	Tid	X'93'	Identifies the Concatenate Bar Code Data triplet	M
2	BITS	CBCFlgs	See Triplet X'93' Semantics for bit definitions	Flags that control the concatenation of bar code data fields	M
3–4	UBIN	SymbID	X'0001'–X'FFFF'	Identifies a bar code symbol concatenation	M
5–6	UBIN	SegOrder	X'0001'–X'FFFF'	Order of concatenation for this segment	M
			X'0000'	Not specified	

Triplet X'93' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Concatenate Bar Code Data triplet

CBCFlgs Specifies additional controls

Bit 0 Start New Symbol if reused (RCD and XMD only)

Value	Description
B'0'	Continue adding to symbol already started
B'1'	Start a new symbol

Bits 1–7 Reserved; should be B'0000000'

Note: An RCD or XMD is reused when the same RCD or XMD is selected more than once on a single page.

SymbID Specifies the ID of a bar code symbol

The symbol ID is used to identify multiple pieces of data to be collected, concatenated, and presented as a single bar code symbol. The SymbID of this triplet, combined with the DescID field of the Bar Code Symbol Descriptor (X'69') triplet and the orientation specified in the LND, RCD, or XMD are used to determine which fields of bar code data are to be concatenated. If the DescID, orientation, color specification triplets, CMRs (order must match as well as names and processing modes), and SymbID matches a previously selected DescID, orientation, color specification triplets, CMRs, and SymbID combination, then the data selected with this LND, RCD, or XMD is to be concatenated with the data selected previously. If a match is not found, a new symbol will be started. The valid range for the ID is X'0001'–X'FFFF'.

Note: When the Bar Code Symbol Descriptor (X'69') triplet requests that trailing blanks are to be suppressed (byte 4 bit 6), the trailing blanks of each piece of data collected will have its trailing blanks suppressed. The remaining data will then be added to the concatenation. The bar code type and modifier will then be adjusted (if necessary) according to the length of the entire concatenation of data.

SegOrder Specifies the order in the concatenation to place this segment of the bar code data

The segments will be concatenated in ascending order using the value specified in this parameter. If more than one bar code data segment specifies the same SegOrder value, the data will be concatenated in the order it is received.

A value of X'0000' in this parameter means the order is not specified. The segments will be concatenated in the order they are received.

All segments in a concatenation must specify the same type of order. If one specifies X'0000', all segments must specify X'0000'. If one specifies a value from X'0001' through X'FFFF', all segments must specify a value from X'0001' through X'FFFF'. It is an error to mix segment order types.

Record Descriptor (RCD)

The Record Descriptor structured field contains information, such as record position, text orientation, font selection, field selection, and conditional processing identification, used to format line data that consists of records tagged with record identifiers.

Note: The RCDs in a Data Map are numbered sequentially, starting with 1.

RCD (X'D3A68D') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A68D'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0–9	CHAR	RecID		Record descriptor ID	M
10	CODE	RecType	X'00'–X'03'	Record Type	M
11–13	BITS	RCDFlgs			M
Bit 0			B'0'	Reserved; should be zero	
Bit 1			B'0'	Reserved; should be zero	
Bit 2			B'0'–B'1'	Generate Inline Position	
Bit 3			B'0'–B'1'	Generate Baseline Position	
Bit 4			B'0'–B'1'	Generate Font Change	
Bit 5			B'0'–B'1'	Generate Suppression	
Bit 6			B'0'–B'1'	Field RCD	
Bit 7			B'0'–B'1'	Use Fixed Data	
Bit 8			B'0'	Reserved; should be zero	
Bit 9			B'0'	Reserved; should be zero	
Bit 10			B'0'	Reserved; should be zero	
Bit 11			B'0'–B'1'	Conditional Processing RCD	
Bit 12			B'0'	Reserved; should be zero	
Bit 13			B'0'–B'1'	Relative Baseline Position	
Bit 14–15			B'00'	Reserved; should be zero	
Bit 16			B'0'–B'1'	New Page	
Bit 17			B'0'–B'1'	Print Page Number	
Bit 18			B'0'–B'1'	Reset Page Number	
Bit 19			B'0'–B'1'	Group Indicator	
Bit 20			B'0'–B'1'	Field Delimiter Size	
Bit 21			B'0'–B'1'	Use Record ID	
Bit 22–23			B'00'	Reserved; should be zero	
14				Reserved; should be zero	M
15–16	UBIN	IPos	0 to page extent minus 1	Inline Position	M
17–18	UBIN	BPos	0 to page extent minus 1	Absolute baseline position	M
	SBIN		X'8000'–X'7FFF'	Relative position	
19–22	CODE	TxtOrent		Text (I,B) Orientation	M
			X'0000 2D00'	0,90 degrees	
			X'2D00 5A00'	90,180 degrees	
			X'5A00 8700'	180,270 degrees	
			X'8700 0000'	270,360 degrees	

Offset	Type	Name	Range	Meaning	M/O
23	CODE	FntLID	X'01'–X'FE'	Primary font local ID	M
			X'FF'	Presentation system default font	
24–25	UBIN	FLDrcd	X'0000'–X'FFFF'	Field RCD Pointer	M
26–33	CHAR	SupName		Suppression token name A value of X'FF...FF' (null value) is not valid.	M
34	CODE	SOLid	X'01'–X'FE'	Shift-out font local ID	M
			X'00'	Not specified	
35–38	UBIN	DataStrt	X'00000000'–X'00007FFF'	Data start position	M
39–40	UBIN	DataLgth	X'0000'–X'FFFE'	Data length	M
			X'FFFF'	Place remaining bytes	
41–42	UBIN	CONDrcd	X'0000'–X'FFFF'	Conditional Processing RCD Pointer	M
43	CODE	SubpgID	X'00'	Subpage ID (always X'00' for RCDs)	M
44–45	CODE	CCPID	X'0000'–X'FFFF'	CCP Identifier	M
46–47	UBIN	Pgno	X'0001'–X'FFFF'	Starting page number	M
			X'0000'	Not specified	
48–49	UBIN	ESpac	0 to page extent minus 1	End Space	M
50	CODE	Align	X'00'–X'01'	Field Alignment	M
51–52	CODE	FldDelim	X'0000'–X'FFFF'	Field Delimiter	M
53–54	UBIN	Fldno	X'0001'–X'FFFF'	Field Number	M
			X'0000'	Not specified	
55–56	UBIN	AdBIncr	X'0000'–X'FFFF'	Additional baseline increment	M
57–69				Reserved; should be zero	M
70–*n*		Triplets		See RCD Semantics for triplet applicability.	O

RCD Semantics

The RCD uses many parameters that are defined for the LND. The definition of such parameters is deferred to the LND. When such definitions are applied to the RCD, the term “LND” should be read as “RCD” and the byte offsets of the parameters should be adjusted to the RCD.

There are three types of RCDs:

- Record RCDs, which define the processing for an input record or define a default page header or trailer
- Field RCDs, which define the processing for an input field or specify constant text or graphics
- Conditional Processing RCDs, which specify the Conditional Processing associated with an input record

The Field RCDs and Conditional Processing RCDs associated with a Record RCD are chained to that RCD using RCD number pointers. An RCD is assumed to be a Record RCD if neither the Field RCD nor the Conditional Processing RCD bits are on in the RCDFlgs byte.

RecID Record Descriptor ID

The identifier to be used with this RCD. This field in the RCD is used only if this RCD does not contain a Fully Qualified Name (FQN) X'02' triplet type X'01'. The FQN type X'01' triplet is used to extend the identifier to a range of 1 to 250 bytes instead of 10 bytes. When an input record is processed with a Data Map containing RCDs, the first 1 to 250 data byte positions in the input record are assumed to contain a Record Descriptor ID and the Data Map's Record RCDs are searched to find a matching Record Descriptor ID. (A CC byte, if used with the input data stream, is not considered part of the input record.) When a match is found, that RCD is used to format the input record. If a matching RCD is not found, an error message is generated. Except for default Page Headers, default Page Trailers, Field RCDs, and Conditional Processing RCDs, all of the RCDs in a Data Map have a unique Record Descriptor ID. The Record Descriptor ID is set to all zeros for default Page Headers (one per Data Map), default Page Trailers (one per Data Map), Field RCDs, and Conditional Processing RCDs. If this RCD contains a FQN type X'01' triplet, this 10 byte Record Descriptor ID field is set to X'FF...FF'.

Notes:

1. To be able to find a matching Record Descriptor ID, the encoding of the identifier specified in this parameter or in the FQN type X'01' triplet must match the encoding of the input data.
2. If the FQN type X'01' triplet is used, all record type RCD structured fields must use the FQN X'01' triplet.
3. If the FQN type X'01' triplet is used, the names specified for all the FQN triplets must be the same length. Blanks may be used to pad the name. The encoding used for the blanks must match the encoding of the data used for the name.

RecType Record Type

This parameter is ignored on RCDs other than Record RCDs.

Value **Description**

X'00' Body RCD

This RCD does not have any special header or trailer properties associated with it and is used to format any input records with a matching Record ID.

X'01' Page Header RCD

This RCD is used to automatically print a header (such as the current customer name) at the beginning of each new page. The baseline position of this RCD can be anywhere on a logical page and can be specified as relative. If an input record is received with a matching Record Descriptor ID, that input record is not presented on receipt but is saved as the active page header record. If a default Record Descriptor ID is specified in a Page Header RCD, it is assumed to be a default Page Header RCD (only one default Page Header RCD can be specified in a Data Map and no input record data is processed with a default RCD). See “Logical Page Eject Processing” on page 132 for page header and new page processing details.

Note: Once a Page Header RCD is processed, the header record is saved for the duration of the job by the presentation services program. Whenever the same Data Map is re-invoked for that job, this saved header record is always presented with each page generated with the Data Map.

X'02' Page Trailer RCD

This RCD is used to automatically print a trailer (for example, a footnote or page number) at the end of each page. The baseline position of this RCD can be anywhere on a logical page and can be specified as relative. If an input record is received with a matching Record Descriptor ID, that input record is not presented on receipt but is saved as the active page trailer record. If a default Record Descriptor ID is specified in a Page Trailer RCD, it is assumed to be a default Page Trailer RCD (only one default Page Trailer RCD can be specified in a Data Map, and no input record data is processed with a default RCD). See “Logical Page Eject Processing” on page 132 for page trailer and new page processing details.

Note: Once a Page Trailer RCD is processed, the trailer record is saved for the duration of the job by the presentation services program. Whenever the same Data Map is re-invoked for that job, this saved trailer record is always presented with each page generated with the Data Map.

X'03' Group Header RCD

This RCD is used to automatically print a group header (for example, column headings for a group of data records) on a page. Note that the group header is not actually printed and causes no action until a Body RCD with Group Indicator (RCDFlgs bit 19) set to B'1' is processed for the page. The baseline position of this RCD can be specified as relative. If an input record is received with a matching Record Descriptor ID, that input record is saved as the active group header record and then presented. If that input record or RCD causes a page eject, that input record is used as the active group header record for the new page. See “Logical Page Eject Processing” on page 132 for active group header and new page processing details.

Note: Once a Group Header RCD is processed and is still active when leaving the Data Map, the group header record is saved by the presentation services program. Whenever the same Data Map is re-invoked, this saved group header record is presented again if the first body record after re-invoking the Data Map selects a Body RCD that has the Group Indicator on.

RCDFlgs Flag bits

Bits 0–5 For a definition of these flag bits see “Line Descriptor (LND)” on page 98. LND flag bit 0–1 is reserved in the RCD.

Bit 6 Field RCD

Value	**Description**
B'0'	If this bit and bit 11 are both B'0', this is a Record RCD.
B'1'	This is a Field RCD. If both this bit and bit 11 are on, it is an error.

Bits 7–15 For a definition of these flag bits see “Line Descriptor (LND)” on page 98. LND flag bits 9–10 are reserved in the RCD. The color for the data presented by this RCD is always the color specified by the Color Specification (X'4E') triplet, if specified. If this triplet is not specified, the data is presented in the presentation process default color.

Bit 16 New Page

Value	**Description**
B'0'	This bit value has no effect on this RCD.
B'1'	A logical page eject should be executed before presenting any data for this RCD. If this is a header or trailer RCD, the print position is moved to the start of a new page before this header or trailer becomes the active header or trailer. (See “Logical Page Eject Processing” on page 132.) This bit is ignored on RCDs other than Record RCDs.

Bit 17 Print Page Number

This flag indicates whether a page number should be generated on the current page before the page eject is performed.

Value	**Description**
B'0'	This bit has no effect on this RCD.
B'1'	A page number is generated.

The (I,B) position of the RCD indicates the position of the page number, which is maintained by the presentation process. The page number is rendered with the font selected by the FntLID parameter. This bit is ignored on RCDs other than Field RCDs.

Note: The FntLID parameter must be mapped to a font global identifier with an MCF structured field in the AEG of the Data Map. For the page formatter to generate the page number code points without accessing the font object, it needs to know the encoding scheme (EBCDIC, ASCII, or Unicode) for the font. This information is specified on the MCF with an Encoding Scheme ID (X'50') triplet. Only single-byte EBCDIC, double-byte EBCDIC, single-byte ASCII, and Unicode (UTF-16) encodings are supported for printing page numbers. The code points used for the page numbers are X'F0'–X'F9', X'42F0'–X'42F9', X'30'–X'39', and X'0030'–X'0039', respectively. If the encoding scheme is not specified on the MCF, single-byte EBCDIC encoding is assumed.

Bit 18 Reset Page Number

Value	**Description**
B'0'	This bit has no effect on this RCD.
B'1'	The current page number is reset to the value specified in the Pgno parameter. This bit is ignored on RCDs other than Field RCDs.

Bit 19 Group Indicator

Value	**Description**
B'0'	The input data associated with this RCD is not part of a group. If a group header record is active, it is deactivated and any saved group header input record is discarded.
B'1'	This flag indicates that the existing group header should continue to be saved and used for subsequent pages. This bit is ignored on RCDs other than Record RCDs.

Bit 20 Field Delimiter Size

Value	**Description**
B'0'	Delimiter is 1 byte and is specified in the second byte of the field.
B'1'	Delimiter is 2 bytes.

Bit 21 Use Record ID

This bit selects the Record ID as the data field to be used for presentation.

Value	**Description**
B'0'	Do not select the Record ID.
B'1'	Select the Record ID.

This function is restricted to Field RCDs; it is ignored on all other RCDs.

Bits 22–23 Reserved; should be B'00'

IPos Inline Position

See “Line Descriptor (LND)” on page 98.

BPos Baseline Position

See “Line Descriptor (LND)” on page 98.

- *Relative Baseline Position for Record and Field RCDs*

  If the baseline position is relative, the offset is measured as follows:

  – For Page Header RCDs, the offset is relative to the top of the page.

  – For Group Header and Body RCDs, the offset is relative to the last Group Header or Body RCD processed; if there is none, it is relative to the top margin.

  – For Field RCDs, it is relative to the last Field or Body RCD that was processed for print (whether or not data is printed).

  – For Page Trailer RCDs, it is relative to the last Record RCD processed; if there is none, it is relative to the top margin.

  Note that the actual location of “top” and “top margin” on a page is affected by the text orientation; see “Margin Definition (X'7F') Triplet” on page 73.

- *Overflow Processing for a Record RCD*

  If the specified baseline position is relative, this Record RCD and any Field RCDs that are part of this Record RCD are scanned to determine the resulting absolute baseline position at the end of processing this Record RCD. This computed baseline position is checked to see if it overflows into the bottom margin area. If not, this Record RCD is processed with the current input record. If it overflows into the bottom margin, a logical page eject is executed. See “Logical Page Eject Processing” on page 132 for page eject processing details. If the data in a single Record RCD (with its chained Field RCDs) is too large to fit within the top and bottom margins on a page, an error is generated. Note that the actual location of “top margin” and “bottom margin” on a page is affected by the text orientation; see “Margin Definition (X'7F') Triplet” on page 73.

TxtOrent Text (I,B) Orientation

See “Line Descriptor (LND)” on page 98.

FntLID Primary Font Local Identifier

See “Line Descriptor (LND)” on page 98.

FLDrcd RCD number of a Field RCD

A non-zero value in this parameter on a Record RCD indicates that field processing is to be performed on the current input data record. This parameter specifies the RCD number of a Field RCD. Multiple Field RCDs can be chained to a Record RCD using this parameter. The last Field RCD in the chain has a value of X'0000' in this parameter.

SupName Suppression token name

See “Line Descriptor (LND)” on page 98.

SOLid Shift-out Font Local Identifier

See “Line Descriptor (LND)” on page 98.

DataStrt Data Start Position

See “Line Descriptor (LND)” on page 98.

DataLgth Data Length

See “Line Descriptor (LND)” on page 98.

CONDrcd RCD number of a Conditional Processing RCD

A non-zero value in this parameter on a Record RCD means that conditional processing is to be performed on the current input data record. This parameter specifies the relative RCD number of a Conditional Processing RCD. Multiple Conditional Processing RCDs can be chained to a Record RCD using this parameter. The last Conditional Processing RCD in the chain has a value of X'0000' in this parameter. A Field RCD has a value of X'0000' in this parameter.

See "Line Descriptor (LND)" on page 98.

SubpgID
Subpage Identifier

See "Line Descriptor (LND)" on page 98.

CCPID
CCP Identifier

See "Line Descriptor (LND)" on page 98.

Pgno
Page Number

This parameter specifies the starting page number that is used when RCDFlgs specifies reset page number. This parameter is ignored on RCDs other than Field RCDs.

ESpac
End Space

Value	Description
X'0000'	No check is made for End Space.
All others	A check is initiated to verify that the remaining body space on the page (distance between the starting print position of this RCD and the bottom margin area) is greater than or equal to the baseline value specified in this parameter. If the remaining body space is less than the value specified, a logical page eject is executed. This can be used, for example, on a Group Header RCD (specifying space for the first data record of the group) to ensure that a group header does not print at the end of a page without the first data record of the group. This parameter is ignored on a Page Header or Page Trailer RCD and on RCDs other than Record RCDs. Note that the actual location of the bottom margin area is affected by the text orientation; see "Margin Definition (X'7F') Triplet" on page 73.

Align
Field Alignment

Value	Description
X'00'	The field is left aligned to the position specified in IPos.
X'01'	The field is right aligned to the position specified in IPos. This parameter is ignored for Field RCDs that specify a Bar Code Symbol Descriptor (X'69') triplet or Graphics Descriptor (X'7E') triplet and on RCDs other than Field RCDs.

FldDelim
Field Delimiter

This is a 1-byte or a 2-byte parameter, depending on the encoding used for the field. RCDFlgs bit 20 indicates the size of this parameter. A 1-byte parameter is specified in the second byte of the field. A value of X'0000' indicates that this parameter is not specified. For Record RCDs, this parameter specifies a 1-byte or 2-byte code that delimits all of the input record fields (excluding the 10 byte Record Format ID) used with this RCD and any Field and Conditional Processing RCDs chained to this RCD. The delimiter is specified at the end of the field it delimits. A delimiter may be specified after the Record ID but is ignored. A Field Number parameter is used rather than DataStrt and DataLgth parameters to specify the location of input fields on Conditional Processing and Field RCDs that are chained to this Record RCD. DataStrt and DataLgth parameters are still used to select bytes in the field or in Fixed Data Text. This parameter is ignored on all non-Record RCDs. For comparisons, any input record fields used with a CCP are assumed to be padded on the right out to the CCP Comparison String Length.

Fldno Field Number

Specifies the number of the field to be processed. The first field (field number 1) in the record is followed by the first delimiter; the second field is followed by the second delimiter; and so on. Any delimiter specified after the Record ID is ignored. The DataStrt and DataLgth parameters are applied to the selected field to select specific bytes in the field for processing. Specifying DataStrt = X'00000000' and DataLgth = X'FFFF' selects the whole field for processing. This parameter is used only if the Record RCD that this Field or Conditional Processing RCD is chained to specifies a Field Delimiter (other than X'00').

AdBlncr Additional baseline increment

Specifies the additional baseline increment that is to be added to the current baseline position after a group header or data record is presented. This parameter is only processed for Group Header RCDs, and for Body RCDs that are record RCDs. It is ignored on all other RCDs. Note that this increment is not used when positioning MO:DCA objects with respect to the current RCD in mixed mode.

Triplets See the following:

“Fully Qualified Name (X'02') Triplet” on page 133 for FQN type X'01' (Replace First GID Name)
“Fully Qualified Name (X'02') Triplet” on page 134 for FQN type X'DE' (Data Object External Resource Reference)
“Extended Resource Local Identifier (X'22') Triplet” on page 135
“Color Specification (X'4E') Triplet” on page 136
“Bar Code Symbol Descriptor (X'69') Triplet” on page 137
“Resource Object Include (X'6C') Triplet” on page 138
“Additional Bar Code Parameters (X'7B') Triplet” on page 139
“Graphics Descriptor (X'7E') Triplet” on page 140
“Object Reference Qualifier (X'89') Triplet” on page 146
“Color Management Resource Descriptor (X'91') Triplet” on page 147
“Concatenate Bar Code Data (X'93') Triplet” on page 148
“Rendering Intent (X'95') Triplet” on page 149

Logical Page Eject Processing

A logical page eject can be caused by the following:

- Any Record RCD with a specification of New Page
- A relative baseline overflow

  The overflow is caused by a Body or Group Header RCD with a relative baseline position value that when processed against the current input record causes an overflow of the current print position into the bottom margin (see Baseline Position on page 129). Note that the actual location of “bottom margin” is affected by the text orientation; see “Margin Definition (X'7F') Triplet” on page 73.
- A Data Map change or Medium Map change, or, in mixed-mode, a Begin Document or Begin Page structured field

When a logical page eject occurs, the following actions are taken in the following order.

For the current page:

1. If this is the start of a line data document (no previous page ejects, group header records, or body records have been processed with this PageDef), no header or trailer processing is performed.
2. If an active page header record was in effect prior to this RCD, that record is presented on the current page using the matching RCD. Otherwise, if the active Data Map contains a default Page Header RCD, that RCD is used to present a page header.
3. If an active page trailer record was in effect prior to this RCD, that record is presented on the current page using the matching RCD. Otherwise, if the active Data Map contains a default Page Trailer RCD, that RCD is used to present a page trailer.

For the new page:

1. The current print position is moved to the top of the new page. If the Data Map is changed, the new Data Map's Margin Definition and RCDs are used for subsequent processing.
2. The baseline position is offset from the top of the new page by the top margin.
3. If the RCD causing the page eject is a Page Header, Group Header, or Page Trailer RCD, the input record causing the page eject is saved as the active page header, group header, or page trailer record.
4. If an active group header record exists for this Data Map, that record is presented on the new page using the matching Record RCD. Note that the group header is not actually printed and causes no action until a Body RCD with Group Indicator (RCDFlgs bit 19) set to B'1' is processed for the page. If the RCD specifies relative positioning, the baseline position of the RCD is offset from the top of the page by the top margin plus the RCD BPos value.
5. If the page eject was caused by a Body RCD, the input record causing the page eject is presented on the new page using the RCD referenced by the record. If the RCD specifies relative positioning and is preceded on the page by a group header, the baseline position is relative to the last printed line of the group header. If the RCD specifies relative positioning and is not preceded on the page by a group header, the baseline position of the RCD is offset from the top of the page (0 position on the B axis) by the top margin plus the RCD BPos value.

Note that the actual location of “top of page” and “top margin” is affected by the text orientation; see “Margin Definition (X'7F') Triplet” on page 73.

RCD Triplets

Fully Qualified Name (X'02') Triplet

The Fully Qualified Name (X'02') triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

This triplet is optional and may occur once. If this triplet is specified more than once, only the first is used. The Fully Qualified Name type that may appear is X'01' — *Replace First GID Name*. This GID overrides the Record Descriptor's RecID field and is used as the Record Descriptor ID.

Triplet X'02' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	5–254	Length of the triplet, including Tlength	M
1	CODE	Tid	X'02'	Identifies the Fully Qualified Name triplet	M
2	CODE	FQNType	X'01'	Replace First GID name	M
3	CODE	FQNFmt	X'00'	GID format is character string	M
4–n		FQName		GID of the Record Descriptor The GID can be up to 250 bytes in length.	M

Triplet X'02' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Fully Qualified Name triplet

FQNType Specifies how the fully qualified name is to be used

Value	Description
X'01'	This GID replaces the RecID parameter in the structured field. **Note:** The GID (Global Identifier) that overrides the ten-byte RecID field has the same semantics as the ten-byte RecID parameter.
All others	Reserved

FQNFmt Specifies the format of the Global Identifier

Value	Description
X'00'	The GID is a character-encoded name, which means the data type is CHAR.
All others	Reserved

FQName Contains the Global Identifier (GID) to be used to override the RecID parameter

Note: To be able to find a matching Record Descriptor ID, the encoding of the identifier specified in this parameter must match the encoding of the input data.

Fully Qualified Name (X'02') Triplet

This triplet is optional and may occur one or more times when a Bar Code Symbol Descriptor (X'69') triplet or a Graphics Descriptor (X'7E') triplet is specified on the RCD. The Fully Qualified Name type that may appear is X'DE'—*Data Object External Resource Reference*. The FQN triplet specifies the external identifier of a Color Management Resource (CMR) object that is used for the Bar Code object or Graphics object being generated. The identifier is used by the presentation system to locate the resource object in the resource hierarchy. The identifier is a character-encoded name that must be specified using FQNFmt = X'00'. The encoding for the external identifier of the CMR must be UTF-16BE.

See “Fully Qualified Name (X'02') Triplet” on page 107.

Extended Resource Local Identifier (X'22') Triplet

This triplet is optional. It may occur one or more times to reference an IOB structured field in the PageDef. This triplet is ignored if the Graphics Descriptor (X'7E') triplet is specified on the RCD.

See “Extended Resource Local Identifier (X'22') Triplet” on page 108.

Color Specification (X'4E') Triplet

This is an optional triplet that specifies the color for text processed by this RCD, for a bar code generated by this RCD, and for graphics generated by this RCD. If this triplet is not specified, the record is presented in the presentation process default color. On an RCD that processes text or generates a bar code, this triplet may occur once. If this triplet is specified more than once, only the first is used. On a Field RCD with a Graphics Descriptor (X'7E') triplet, this color triplet can be specified once or twice.

- If it is specified once, the color applies to all graphics constructs.
- If it is specified twice, then:
  - If the RCD generates a graphics area, such as a full arc or box, the first color applies to the fill pattern of the graphics area and the second color applies to the boundary lines of the graphics area (if drawn).
  - If the RCD generates a graphics line, the second color overrides the first color and applies to the graphics line.
- If it is specified more than twice, only the first two are used.

If this RCD generates the start of a graphic primitive that is completed by a succeeding RCD, the colors specified on the “start” RCD determine the color for the complete primitive.

With this triplet, a color is specified by selecting a color space, an encoding for the components of the color value in that space, and a color value. The color spaces supported are:

- RGB
- CMYK
- Highlight
- CIELAB
- Standard OCA color space

The selected encoding defines the number of bits used to specify each component. For example, with the RGB color space, one supported encoding is 8 bits per component, which maps the component intensity range 0 to 1 to the binary values 0 to 255. The color value specifies the color. With the RGB color space and an 8 bit per component encoding, the color value (255,255,255) specifies full intensity for each component, which defines the color white.

The Color Specification triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Bar Code Symbol Descriptor (X'69') Triplet

This is an optional triplet. It may occur once. If this triplet is specified more than once, only the first is used. This triplet is ignored if the Graphics Descriptor (X'7E') triplet is specified on the RCD.

See “Bar Code Symbol Descriptor (X'69') Triplet” on page 110. Note that the LND/RCD parameters used by this triplet may be at different offsets in the LND and RCD.

Resource Object Include (X'6C') Triplet

This is an optional triplet that identifies an overlay or page segment object to be presented on the page at a specified position. Multiple Resource Object Include triplets may be specified on the same RCD. This triplet is ignored if the Graphics Descriptor (X'7E') triplet is specified on the RCD.

See “Resource Object Include (X'6C') Triplet” on page 116.

Additional Bar Code Parameters (X'7B') Triplet

This is an optional triplet that may occur one or more times when a Bar Code Symbol Descriptor (X'69') triplet is specified. If this triplet is specified more than once, the data from each triplet is concatenated in the order it is received.

See “Additional Bar Code Parameters (X'7B') Triplet” on page 118.

Graphics Descriptor (X'7E') Triplet

Architecture Note: The Graphics Descriptor triplet is registered in the MO:DCA architecture as a private-use triplet because it is used only in the PageDef object, which is not a MO:DCA object.

This is an optional Field RCD triplet. It may occur once. If this triplet is specified more than once, only the first is used. Text input and fixed data text are ignored on a Field RCD that specifies a Graphics Descriptor triplet. When present, the Graphics Descriptor triplet specifies the generation of a graphics primitive. The triplet may specify the complete primitive, or the start of the primitive, or the end of the primitive.

This triplet is ignored on RCDs other than Field RCDs. This triplet specifies primitives and their parameters as defined by the AFP GOCA architecture. For more information, see the *Graphics Object Content Architecture for Advanced Function Presentation Reference*. Not all presentation services programs support this triplet.

Triplet X'7E' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	20 or 35	Length of the triplet, including Tlength	M
1	CODE	Tid	X'7E'	Identifies the Graphics Descriptor triplet	M
2	CODE	ParmSpc	 X'01' X'02' X'03'	Parameter specification: Triplet specifies all parameters for graphics primitive Triplet specifies start parameters for graphics primitive Triplet specifies end parameters for graphics primitive	M
3–4	UBIN	Graphid	X'0000'–X'FFFE'	ID for matching Start/End graphic pairs	M
5	CODE	GrPrim	 X'01' X'02' X'03' X'04' X'05'	Graphics primitive: Horizontal line at current position Vertical line at current position Diagonal line at current position Full Arc at current position Box at current position	M
6				Reserved; should be zero	M
7	BITS	GraFlgs			M
Bit 0			B'0'	Reserved; should be zero	
Bit 1		Fill	B'0' B'1'	Interior of primitive is not filled Interior of primitive is filled	
Bit 2		Boundary	B'0' B'1'	Boundary of primitive is not drawn Boundary of primitive is drawn	
Bit 3–7			B'00000'	Reserved; should be zero	
8–9	UBIN	Iend	0 to page extent minus 1	I-coordinate for primitive end point	M
10–11	UBIN	Bend	0 to page extent minus 1	B-coordinate for primitive end point	M
12–13	UBIN	HAXIS	0–32,767	Length of ellipse X-axis (parallel to I-axis) for rounded corner on box	M
14–15	UBIN	VAXIS	0–32,767	Length of ellipse Y-axis (parallel to B-axis) for rounded corner on box	M
16	UBIN	MH	X'00'–X'FF'	Integer multiplier for radius of full arc	M
17	UBIN	MFR	X'00'–X'FF'	Fractional multiplier for radius of full arc	M

Offset	Type	Name	Range	Meaning	M/O
18–19	CODE	DescID	X'0001'–X'FFFE'	Identifies a graphics descriptor	M
20	CODE	FGMix	 X'02'	Foreground mixing rule: Overpaint	O
21				Reserved; should be zero	O
22	CODE	LineTpe	See AFP GOCA.	Line type for primitive boundary	O
23	CODE	LineWMH	See AFP GOCA.	Line width for primitive boundary: integral multiplier	O
24	CODE	LineWMFR	See AFP GOCA.	Line width for primitive boundary: fractional multiplier	O
25	CODE	PattSet	X'00'	Pattern set for primitive fill	O
26	CODE	PattSymb	See AFP GOCA.	Pattern symbol for primitive fill	O
27–28	SBIN	XMAJ	See AFP GOCA.	I-coordinate of arc major axis end point	O
29–30	SBIN	YMIN	See AFP GOCA.	B-coordinate of arc minor axis end point	O
31–32	SBIN	XMIN	See AFP GOCA.	I-coordinate of arc minor axis end point	O
33–34	SBIN	YMAJ	See AFP GOCA.	B-coordinate of arc major axis end point	O

Triplet X'7E' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Graphics Descriptor triplet

ParmSpc Specifies whether the triplet defines a complete primitive or whether it defines only the starting parameters or only the ending parameters

Value **Description**

X'01' All parameters

The triplet specifies all the parameters required to generate the primitive. For horizontal lines, the RCD IPos/Bpos parameters specify the start point and the Iend parameter specifies the ending inline position. For vertical lines, the RCD IPos/Bpos parameters specify the start point and the Bend parameter specifies the ending baseline position. For diagonal lines, the RCD IPos/Bpos parameters specify the start point and the Iend/Bend parameters specify the end point. For boxes, the RCD IPos/Bpos parameters specify the left top corner of the box and the Iend/Bend parameters specify the right bottom corner. For horizontal lines, the ParmSpc value is always assumed to be X'01'.

X'02' Start parameters

For vertical lines, the RCD IPos/Bpos parameters specify the start point and the ending baseline position (BPos) is specified by a Graphics Descriptor triplet with ParmSpc = X'03' on an ensuing RCD with a matching Graphid. For boxes, the RCD IPos/Bpos parameters specify the left edge of the box and the Iend parameter specifies the right edge of the box. The ending BPos parameter is specified by a Graphics Descriptor triplet with ParmSpc = X'03' on an ensuing RCD with a matching Graphid.

For diagonal lines, both the start point and the ending inline position are specified in the start parameters. The RCD IPos/Bpos parameters specify the start point and the Iend value specifies the ending inline position. The ending baseline position is specified by a Graphics Descriptor triplet with ParmSpc = X'03' on an ensuing RCD with a matching Graphid.

If a logical page eject is processed while any lines or boxes are active (have been started but not ended), these lines or boxes are ended. The bottom margin is used as the ending baseline position. Note that the actual location of “bottom margin” on a page is affected by the text orientation; see “Margin Definition (X'7F') Triplet” on page 73.

X'03' End parameters

The triplet specifies the ending baseline position for lines or boxes started by Graphics Descriptor (X'7E') triplets on previous RCDs. The Graphid parameter specifies the lines or boxes ended by this triplet. The RCD BPos specifies the ending baseline position for lines and boxes.

All others Reserved

Graphid Specifies an identifier that is used to tie one or more start (ParmSpc = X'02') and end (ParmSpc = X'03') Graphics Descriptor (X'7E') triplets together

An ending triplet ends all of the active lines and boxes that were started with a matching Graphid. Note that the start and end triplets must have the same orientation.

GrPrim Specifies the graphics primitive that is to be generated

The primitives and their parameters are specified based on definitions in the AFP GOCA architecture. See the *Graphics Object Content Architecture for Advanced Function Presentation Reference*.

Value **Description**

X'01'–X'03' Line at current position

A straight line is generated between two points. The line is defined by the line type (LineTpe) and line width (LineWMH, LineWMFR) parameters in the descriptor. The color of the line is determined by the Color Specification (X'4E') triplet on the RCD. If two X'4E' triplets are specified, it is determined by the second triplet. If a X'4E' triplet is not specified, it is the presentation process default color.

If the GrPrim value is X'01' (horizontal), the line is parallel to the I axis and ParmSpc is assumed to be X'01'. If the GrPrim value is X'02' (vertical), the line is perpendicular to the I axis. If the GrPrim value is X'03' (diagonal), the line is diagonal to the I axis. See ParmSpc for a description of the parameters used to draw the line.

X'04' Full Arc at current position

A circle or ellipse is generated with center at the current RCD position. The ParmSpc parameter is ignored for this primitive.

The color of the boundary line for the arc, if drawn, is determined by the *second* Color Specification (X'4E') triplet on the RCD or by the first Color Specification triplet if only one is specified. If a X'4E' triplet is not specified, the color of the boundary line is the presentation process default color. The color of the fill pattern for the interior of the arc is determined by the *first* Color Specification (X'4E') triplet on the RCD; if a X'4E' triplet is not specified, the color of the fill pattern is the presentation process default color.

X'05' Box at current position

A box is either specified by a single RCD or by a begin/end pair of RCDs. The box is generated with square corners or rounded corners, depending on the value of the HAXIS and VAXIS parameters. If either parameter is zero, square corners are generated. If they are non-zero but equal, the corners are quadrants of a circle whose diameter is HAXIS. If they are non-zero and not equal, the corners are quadrants of an ellipse whose full axes are HAXIS and VAXIS.

The color of the boundary line for the box, if drawn, is determined by the *second* Color Specification (X'4E') triplet on the RCD or by the first Color Specification triplet if only one is specified. If a X'4E' triplet is not specified, the color of the boundary line is the presentation process default color. The color of the fill pattern for the interior of the box is determined by the *first* Color Specification (X'4E') triplet on the RCD; if a X'4E' triplet is not specified, the color of the fill pattern is the presentation process default color.

See ParmSpc for more detail on how the box is drawn.

All others Reserved

GraFlgs Flags that specify how the primitive is generated

GraFlgs bits have the following definitions:

Bit 0 Fill

This flag indicates whether the interior of the primitive (the inside of the circle, ellipse, or box) is to be filled with a colored pattern. The pattern is specified by the pattern set (PattSet) and pattern symbol (PattSymb) parameters in the descriptor and corresponds to one of the pattern symbols in the default pattern set defined in the AFP GOCA architecture. The color of the fill pattern is determined by the *first* Color Specification (X'4E') triplet on the RCD; if a Color Specification triplet is not specified, it is the presentation process default color.

This bit is ignored for the line primitive.

Value	Description
B'0'	Do not fill the interior of the primitive
B'1'	Fill the interior of the primitive

Bit 1 Boundary

This flag indicates whether the boundary of the primitive (circle, ellipse, or box) is to be drawn with a line. The line is specified by the line type (LineTpe) and line width (LineWMH, LineWMFR) parameters in the descriptor.

Value	Description
B'0'	Do not draw the boundary of the primitive
B'1'	Draw the boundary of the primitive

If drawn, the color of the boundary is determined by the *second* Color Specification (X'4E') triplet on the RCD or by the first Color Specification triplet if only one is specified. If a X'4E' triplet is not specified, the color of the boundary line is the presentation process default color.

This bit is ignored for the line primitive, which is always drawn.

Bits 2–7 Reserved; all bits should be B'0'

Iend Specifies the I position of the end point for the primitive

This parameter is ignored if ParmSpc does not equal X'01' or X'02'.

Bend Specifies the B position of the end point for the primitive

This parameter is ignored if ParmSpc does not equal X'01'. This parameter specifies a relative baseline position if the RCD specifies a relative baseline position.

HAXIS Used only for the Box primitive to specify the length of the ellipse X-axis (parallel to the I-axis)

VAXIS Used only for the Box primitive to specify the length of the ellipse Y-axis (parallel to the B-axis)

MH Specifies the integer portion of the scale factor that is applied to the circle or ellipse defined by the XMAJ, XMIN, YMAJ, YMIN parameters in the descriptor

MFR Specifies the fractional portion of the scale factor that is applied to the circle or ellipse defined by the XMAJ, XMIN, YMAJ, YMIN parameters in the descriptor

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

For a circle, the radius is (MH×R + MFR×R), where R is the radius of the circle defined by the current arc parameters. For an ellipse, the major and minor axes are (MH×MAJ + MFR×MAJ) and (MH×MIN + MFR×MIN), where MAJ and MIN are the major and minor axes of the ellipse.

DescID Specifies the ID of a graphics descriptor

The descriptor is defined by bytes 20–34 of this triplet. If the ID matches the ID of a graphics descriptor defined previously on an RCD for this page, the previous descriptor is used regardless of whether bytes 20–34 are specified in the current descriptor. If the ID does not match a previously-defined ID, bytes 20–34 must be specified and define the graphics descriptor that is to be identified with this ID and that is to be used to generate this primitive. If this Graphics Descriptor (X'7E') triplet specifies ParmSpc = X'03', this parameter and all bytes in the graphics descriptor are ignored. The valid range for the ID is X'0001'—X'FFFE'.

For a given page, the presentation services program collects all graphics primitives that have the same graphics descriptor ID and the same orientation and groups them into a single graphics object.

The origin for the graphics object area is one of the four corners of the page as determined by the text orientation specified in the RCD (TxtOrent) and is therefore coincident with the current (I,B) origin. The rotation of the graphics object area about the page X_p-axis matches the rotation of the current text (I,B) coordinate system. For example, with a (90°,180°) text orientation, the object area rotation is 90°. The extents of the object area are determined by the extents of the page.

The position of the graphics primitive in the (I,B) coordinate system therefore maps to the same position in the object area (X_{oa},Y_{oa}) coordinate system. This position in turn is projected to a graphics window whose upper left corner is at the graphics presentation space (GPS) origin and whose extents match those of the object area. The upper left corner of the graphics presentation space window is therefore also coincident with the current (I,B) origin. The mapping between graphics window and object area is *position and trim*.

For example, if the RCD specifies a (90°,180°) text orientation, the upper left corner of the graphics window is at the top-right corner of the page and graphics primitives in this object are rotated 90° with respect to the page X_p axis. The X-extent of the graphics window is the Y_p-extent of the page and the Y-extent of the graphics window is the X_p-extent of the page.

The units of measure for the graphics presentation space and for the graphics object area are the same as those defined on the page (X_p,Y_p) presentation space in the PGD structured field of the Active Environment Group (AEG) for the Data Map.

FGMix Specifies how the graphics primitive mixes with underlying data

The only mixing supported is X'02' (Overpaint). This parameter is specified in an AFP GOCA object with the GDD Set Current Defaults instruction and the Set Mix drawing order.

LineTpe Specifies the type of line that is drawn

For supported values, see the *Graphics Object Content Architecture for Advanced Function Presentation Reference*. This parameter is specified in an AFP GOCA object with the GDD Set Current Defaults instruction and the Set Line Type drawing order.

LineWMH Specifies the width of the line that is drawn by defining the integer portion of the normal line width multiplier

For supported values, see the *Graphics Object Content Architecture for Advanced Function Presentation Reference*. This parameter is specified in an AFP GOCA object with the GDD Set Current Defaults instruction and the Set Line Width drawing order.

LineWMFR Specifies the width of the line that is drawn by defining the fractional portion of the normal line width multiplier

For supported values, see the *Graphics Object Content Architecture for Advanced Function Presentation Reference*. This parameter is specified in an AFP GOCA object with the Set Fractional Line Width drawing order.

PattSet Specifies the pattern set that contains the pattern symbols used to fill the interior of a primitive

The only value supported is X'00'—default pattern set. This parameter is specified in an AFP GOCA object with the GDD Set Current Defaults instruction and the Set Pattern Set drawing order.

PattSymb Specifies the pattern symbol within the current pattern set that is used to fill the interior of a primitive

For supported values, see the *Graphics Object Content Architecture for Advanced Function Presentation Reference*. This parameter is specified in an AFP GOCA object with the GDD Set Current Defaults instruction and the Set Pattern Symbol drawing order.

XMAJ, XMIN, YMAJ, YMIN Together, these parameters define a circle or ellipse on the (I,B) coordinate system

The center of the arc is at the (I,B) origin. When this triplet is used to generate a circle or ellipse, the center is moved to the (I,B) position specified by the RCD and the arc is scaled by the MH.MFR scale factor. Specifically, the four parameters specify the following:

XMAJ	I coordinate of major axis endpoint
YMAJ	B coordinate of major axis endpoint
XMIN	I coordinate of minor axis endpoint
YMIN	B coordinate of minor axis endpoint

These parameters are specified in an AFP GOCA object with the GDD Set Current Defaults instruction and the Set Arc Parameters drawing order.

Notes:

1. The last 15 bytes (bytes 20–34) in this triplet are optional as a group. That is, either they are all specified or none are specified. If the descriptor ID is intended to match a previously-defined descriptor ID, these bytes should not be specified.
2. The X'22', X'69', and X'6C' triplets are ignored when this triplet is specified on an RCD.

Object Reference Qualifier (X'89') Triplet

The Object Reference Qualifier (X'89') triplet is used to specify whether the name of an object is retrieved from the input data or retrieved using normal methods. If the name is to be retrieved from the input data, that name overrides any ObjName field and any Fully Qualified Name (type X'01') triplet that would normally be used to select an object. This triplet may occur once on an RCD structured field.

See "Object Reference Qualifier (X'89') Triplet" on page 119.

Color Management Resource Descriptor (X'91') Triplet

The Color Management Resource Descriptor triplet specifies the processing mode and scope for a Color Management Resource (CMR). This triplet is mandatory when the RCD references a Color Management Resource (CMR) with the FQN type X'DE' triplet, in which case this triplet specifies the processing mode for the CMR and must occur once for each FQN type X'DE' specified. It is ignored in all other cases. This triplet must immediately follow the FQN type X'DE' triplet that specifies the CMR name.

See “Color Management Resource Descriptor (X'91') Triplet” on page 121.

Concatenate Bar Code Data (X'93') Triplet

This is an optional triplet and may occur once. If this triplet is specified more than once, only the first will be used.

In the RCD, completion of a given concatenated bar code symbol is defined as the end of the page or an RCD with the Start New Symbol flag (bit 0 of CBCFlgs) set in the X'93' triplet being reused.

See "Concatenate Bar Code Data (X'93') Triplet" on page 122.

Rendering Intent (X'95') Triplet

The Rendering Intent (X'95') triplet is a MO:DCA triplet. For the formal definition of this triplet, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

The Rendering Intent triplet specifies the rendering intent parameter, which is used to modify the final appearance of color data. This parameter is based on the rendering intents defined by the International Color Consortium (ICC). This triplet is optional and may occur once when a Graphics Descriptor (X'7E') triplet is specified on the RCD. If this triplet is specified more than once, only the first is used. This triplet specifies the rendering intent that is to be used when presenting the Graphics object that is generated with this structured field. Only the rendering intent that applies to the object type of the referenced object is used; the other rendering intents are ignored.

Triplet X'95' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	10	Length of the triplet, including Tlength	M
1	CODE	Tid	X'95'	Identifies the Rendering Intent triplet	M
2–3				Reserved; should be zero	M
4–6			X'FFFFFF'	Not used	M
7	CODE	GOCARI	 X'00' X'01' X'02' X'03' X'FF'	Rendering intent for AFP GOCA objects: Perceptual Media-relative colorimetric Saturation ICC-absolute colorimetric Not specified	M
8–9				Reserved; should be zero	M

Triplet X'95' Semantics

Tlength Contains the length of the triplet

Tid Identifies the Rendering Intent triplet

GOCARI Specifies the rendering intent for AFP GOCA objects

Valid values are the following.

Value	Description
X'00'	Perceptual Gamut mapping is vendor-specific and colors are adjusted to give a pleasing appearance. This intent is typically used to render continuous-tone images.
X'01'	Media-relative colorimetric In-gamut colors are rendered accurately and out-of-gamut colors are mapped to the nearest value within the gamut. Colors are rendered with respect to the source white point and are adjusted for the media white point. Therefore colors printed on two different media with different white points do not match colorimetrically, but might match visually. This intent is typically used for vector graphics.
X'02'	Saturation

Gamut mapping is vendor-specific and colors are adjusted to emphasize saturation. This intent results in vivid colors and is typically used for business graphics.

X'03' ICC-absolute colorimetric

In-gamut colors are rendered accurately and out-of-gamut colors are mapped to the nearest value within the gamut. Colors are rendered only with respect to the source white point and are not adjusted for the media white point. Therefore colors printed on two different media with different white points should match colorimetrically, but might not match visually. This intent is typically used for logos.

X'FF' Rendering intent not specified

All others Reserved

XML Descriptor (XMD)

The XML Descriptor structured field contains information, such as data position, text orientation, font selection, field selection, and conditional processing identification, used to format XML data that consists of text delimited by start and end tags.

Note: The XMDs in a Data Map are numbered sequentially, starting with 1.

XMD (X'D3A68E') Syntax

Structured Field Introducer				
SF Length (2B)	ID = **X'D3A68E'**	Flags (1B)	Reserved X'0000'	Structured Field Data

Offset	Type	Name	Range	Meaning	M/O
0	CODE	ElmType	X'00'–X'03'	Element Type	M
1–3	BITS	XMDFlgs			M
Bit 0–1			B'00'	Reserved; should be zero	
Bit 2			B'0'–B'1'	Generate Inline Position	
Bit 3			B'0'–B'1'	Generate Baseline Position	
Bit 4			B'0'–B'1'	Generate Font Change	
Bit 5			B'0'–B'1'	Generate Suppression	
Bit 6			B'0'–B'1'	Field XMD	
Bit 7			B'0'–B'1'	Use Fixed Data	
Bit 8–9			B'00'	Reserved; should be zero	
Bit 10			B'0'–B'1'	Attribute XMD	
Bit 11			B'0'–B'1'	Conditional Processing XMD	
Bit 12			B'0'–B'1'	Relative Inline Position	
Bit 13			B'0'–B'1'	Relative Baseline Position	
Bit 14–15			B'00'	Reserved, should be zero	
Bit 16			B'0'–B'1'	New Page	
Bit 17			B'0'–B'1'	Print Page Number	
Bit 18			B'0'–B'1'	Reset Page Number	
Bit 19			B'0'–B'1'	Group Indicator	
Bit 20			B'0'–B'1'	Field Delimiter Size	
Bit 21			B'0'–B'1'	Use Start Tag	
Bit 22			B'0'	Reserved, should be zero	
Bit 23			B'0'–B'1'	Header/Trailer Continued	
4				Reserved; should be zero	M
5–6	UBIN	IPos	0 to page extent minus 1	Absolute inline position	M
	SBIN		X'8000'–X'7FFF'	Relative inline position	
7–8	UBIN	BPos	0 to page extent minus 1	Relative baseline position	M
	SBIN		X'8000'–X'7FFF'		
9–12	CODE	TxtOrent		Text (I,B) Orientation:	M
			X'0000 2D00'	0,90 degrees	
			X'2D00 5A00'	90,180 degrees	
			X'5A00 8700'	180,270 degrees	
			X'8700 0000'	270,360 degrees	

XML Descriptor (XMD)

Offset	Type	Name	Range	Meaning	M/O
13	CODE	FntLID	X'01'–X'FE'	Primary font local ID	M
			X'FF'	Presentation system default font	
14–15	UBIN	FLDxmd	X'0000'–X'FFFF'	Field XMD Pointer	M
16–17				Reserved; should be zero	M
18–25	CHAR	SupName		Suppression token name A value of X'FF...FF' (null value) is not valid.	M
26				Reserved; should be zero	M
27–30	UBIN	DataStrt	X'00000000'–X'00007FFF'	Data start position	M
31–32	UBIN	DataLgth	X'0000'–X'FFFE'	Data length	M
			X'FFFF'	Place remaining bytes	
33–34	UBIN	CONDxmd	X'0000'–X'FFFF'	Conditional Processing XMD Pointer	M
35	CODE	SubpgID	X'00'	Subpage ID (Always X'00' for XMDs)	M
36–37	CODE	CCPID	X'0000'–X'FFFF'	CCP Identifier	M
38–39	UBIN	Pgno	X'0001'–X'FFFF'	Starting page number	M
			X'0000'	Not specified	
40–41	UBIN	ESpac	0 to page extent minus 1	End Space	M
42	CODE	Align	X'00'–X'01'	Field Alignment	M
43–44	CODE	FldDelim	X'0000'–X'FFFF'	Field Delimiter	M
45–46	UBIN	Fldno	X'0001'–X'FFFF'	Field Number	M
			X'0000'	Not specified	
47–48	UBIN	AdBIncr	X'0000'–X'FFFF'	Additional baseline increment	M
49–61				Reserved; should be zero	M
62–*n*		Triplets		See XMD Semantics for triplet applicability.	O

XMD Semantics

The XMD uses many parameters that are defined for the LND or RCD. The definition of such parameters is deferred to the LND or RCD. When such definitions are applied to the XMD, the term “LND” or “RCD” should be read as “XMD” and the byte offsets of the parameters should be adjusted to the XMD.

There are four types of XMDs:

- Element XMDs, which define the processing of the data content of the XML element or define a default page header or trailer.
- Field XMDs, which define the processing for an input field or specify constant text or graphics.
- Attribute XMDs, which define the processing for attributes specified in an XML start tag. Attribute XMDs are a special type of Field XMD.
- Conditional Processing XMDs, which specify the Conditional Processing associated with an input element.

The Field XMDs, Attribute XMDs, and Conditional Processing XMDs associated with an Element XMD are chained to that XMD using XMD number pointers. An XMD is assumed to be an Element XMD if neither the Field XMD, Attribute XMD, nor the Conditional Processing XMD bits are on in the XMDFlgs byte.

ElmType Element Type

This parameter is ignored on XMDs other than Element XMDs.

Value **Description**

X'00' Body Element

This XMD does not have any special header or trailer properties associated with it and is used to format any input elements with a matching Qualified Tag.

X'01' Page Header Element

This XMD is used to automatically print a header (such as the current customer name) at the beginning of each new page. The baseline position of this XMD can be anywhere on a logical page and can be specified as relative. If an input element is received with a matching Qualified Tag, that input element is not presented on receipt but is saved as the active page header. If no Qualified Tag is specified for an XMD that has the Page Header element type, it is assumed to be a default Page Header XMD. Only one default Page Header XMD can be specified in a Data Map and no input element data is processed with a default XMD. See "Logical Page Eject Processing" on page 132 for page header and new page processing details (note that page eject processing for RCD and XMD is identical).

Note: Once a Page Header XMD is processed, the header element is saved for the duration of the document by the presentation services program. Whenever the same Data Map is re-invoked for that document, this saved header element is always presented with each page generated with the Data Map.

X'02' Page Trailer Element

This XMD is used to automatically print a trailer (for example, a footnote or page number) at the end of each page. The baseline position of this XMD can be anywhere on a logical page and can be specified as relative. If an input element is received with a matching Qualified Tag, that input element is not presented on receipt but is saved as the active page trailer. If no Qualified Tag is specified for an XMD that has the Page Trailer element type, it is assumed to be a default Page Trailer XMD. Only one default Page Trailer XMD can be specified in a Data Map and no input element data is processed with a default XMD. See "Logical Page Eject Processing" on page 132 for page trailer and new page processing details (note that page eject processing for RCD and XMD is identical).

Note: Once a Page Trailer XMD is processed, the trailer element is saved for the duration of the document by the presentation services program. Whenever the same Data Map is re-invoked for that document, this saved trailer element is always presented with each page generated with the Data Map.

X'03' Group Header Element

This XMD is used to automatically print a group header (for example, column headings for a group of elements) on a page. Note that the group header is not actually printed and causes no action until a Body Element XMD with Group Indicator (XMDFlgs bit 19) set to B'1' is processed for the page. The baseline position of this XMD can be specified as relative. If an input element is received with a matching Qualified Tag, that input element is saved as the active group header and then presented. If that input element or XMD causes a page eject, that input element is used as the active group header for the new page. See "Logical Page Eject Processing" on page 132 for active group header and new page processing details (note that page eject processing for RCD and XMD is identical).

Note: Once a Group Header XMD is processed and is still active when leaving the Data Map, the group header element is saved by the presentation services program. Whenever the same Data Map is re-invoked, this saved group header element is presented again if the first body element after re-invoking the Data Map selects a Body Element XMD that has the Group Indicator on.

Note: The formation of the Page Header, Group Header, or Page Trailer might require element content from more than one element. This is accomplished by the use of XMDFlgs bit 23 (Header/Trailer Continued). Refer to the description of the Header/Trailer Continued flag for more information about continued headers and trailers.

XMDFlgs Flag bits

Bits 0–5 For a definition of these flag bits see "Line Descriptor (LND)" on page 98. LND flag bits 0–1 are reserved in the XMD.

Bit 6 Field XMD

Value	Description
B'0'	If this bit and bits 10 and 11 are all B'0', this is an Element XMD.
B'1'	This is a Field XMD. If this bit and bit 11 is on, it is an error.

Bits 7–9 For a definition of these flag bits see "Line Descriptor (LND)" on page 98. LND flag bit 9 is reserved in the XMD.

Bit 10 Attribute XMD

Value	Description
B'0'	If this bit and bits 6 and 11 are all B'0', this is an Element XMD.
B'1'	This is an Attribute XMD. If this bit is on, bit 6 must also be on since an Attribute XMD is a special type of Field XMD. If this bit and bit 11 are on, it is an error.

When processing an attribute of the XML Start tag, the processor matches the attribute name to the name specified in the XML Name (X'8A') triplet that was specified for this XMD. Once a match is found, the attribute value is used as the data to be processed by this XMD.

Notes:

1. The attribute value is the quoted string not containing the quotation mark used as a delimiter for that string.
2. The color for the data presented by this XMD is always the color specified by the Color Specification (X'4E') triplet, if specified. If this triplet is not specified, the data is presented in the presentation process default color.

Bit 11 For a definition of this flag bit see "Line Descriptor (LND)" on page 98.

Bit 12 Relative Inline Position

This bit indicates whether the inline position specified in bytes 5–6 of this XMD is an absolute position or a relative position. If an absolute inline position is specified, it is measured as a positive offset in the inline direction from the current text (I,B) coordinate system origin. If a relative position is specified, it is measured as a positive or negative offset from the current inline position using the current text (I,B) coordinate system, which is defined by the text orientation specified in bytes 9–12.

Value	Description
B'0'	The inline position specified in bytes 5–6 is an absolute position.
B'1'	The inline position specified in bytes 5–6 is a relative position.

The following restriction applies to relative inline positioning:

- The text orientation of an XMD that specifies relative inline positioning must be the same as the text orientation of the XMD that defines the inline position from which the relative offset is measured.

Bits 13–15
For a definition of these flag bits see “Line Descriptor (LND)” on page 98.

Bits 16–20
For a definition of these flag bits see “Record Descriptor (RCD)” on page 124.

Bit 21
Use Start Tag

This bit selects the Start tag (including the angle brackets '<' and '>') as the data field to be used for presentation.

Value	Description
B'0'	Do not select the Start Tag
B'1'	Select the Start Tag

This function is restricted to Field XMDs; it is ignored on all other XMDs.

Bit 22
Reserved; should be zero

Bit 23
Header/Trailer Continued

This bit indicates that this XMD is a continuation of a Page Header, Group Header, or Page Trailer definition. The formation of the Page Header, Group Header, or Page Trailer might require the element content from more than one element. This is accomplished by specifying this continuation indicator along with specifying the appropriate ElmType parameter to identify which type of header or trailer that is to be continued. If the header or trailer has not started, this header or trailer XMD starts the header or trailer regardless of the setting of this flag.

For Page Headers and Page Trailers, the elements that are used to build the continued header or trailer do not need to be contiguous in the XML data, but do need to be on the same page. If a page break occurs before a continued header or trailer is reached, the continued header or trailer acts as a new header or trailer. If the continued header or trailer is on the same page, the XMD structured fields used to print the various pieces of the continued header or trailer are processed as though the elements were specified contiguously. This means any relative positioning specified is relative to data placed using a previous header or trailer XMD, if any.

For Group Headers, the elements that are used to build the continued header do not need to be contiguous, but cannot have body elements placed between the pieces of the Group Header. If body elements are placed between the pieces of a continued Group Header, the Group Header is not continued but acts as a new Group Header.

Value	Description
B'0'	Not a continuation XMD
B'1'	Continued Header or Trailer definition If the header or trailer has started, data continues to be collected to form the header or trailer.

This function is ignored on Body Element XMDs.

IPos
Inline Position

See “Line Descriptor (LND)” on page 98.

Relative Inline Position for Element, Field, and Attribute XMDs: If the inline position is relative, the offset is relative to the current inline position. If there is no prior XMD, the relative inline position is relative to the left margin.

Note: Data must not exceed the boundaries of the page, which are defined in the Page Descriptor (PGD) structured field. If the new print position is outside these boundaries, printing of the page stops.

Note that the actual location of the “left margin” on a page is affected by the text orientation; see “Margin Definition (X'7F') Triplet” on page 73.

BPos
Baseline Position

See “Record Descriptor (RCD)” on page 124.

TxtOrent
Text (I,B) Orientation

See “Line Descriptor (LND)” on page 98.

FntLID
Primary Font Local Identifier

See “Line Descriptor (LND)” on page 98.

FLDxmd
XMD number of a Field XMD

See “Record Descriptor (RCD)” on page 124.

SupName
Suppression token name

See “Line Descriptor (LND)” on page 98.

DataStrt
Data Start Position

See “Line Descriptor (LND)” on page 98.

DataLgth
Data Length

See “Line Descriptor (LND)” on page 98.

CONDxmd
XMD number of a Conditional Processing XMD

See “Record Descriptor (RCD)” on page 124.

SubpgID
Subpage Identifier

See “Line Descriptor (LND)” on page 98.

CCPID
CCP Identifier

See “Line Descriptor (LND)” on page 98.

Pgno
Page Number

See “Record Descriptor (RCD)” on page 124.

ESpac
End Space

See “Record Descriptor (RCD)” on page 124.

Align
Field Alignment

See “Record Descriptor (RCD)” on page 124.

FldDelim
Field Delimiter

See “Record Descriptor (RCD)” on page 124.

Fldno
Field Number

See “Record Descriptor (RCD)” on page 124.

AdBlncr Additional baseline increment

See “Record Descriptor (RCD)” on page 124.

Triplets See the following:
“Fully Qualified Name (X'02') Triplet” on page 158
“Extended Resource Local Identifier (X'22') Triplet” on page 159
“Color Specification (X'4E') Triplet” on page 160
“Bar Code Symbol Descriptor (X'69') Triplet” on page 161
“Resource Object Include (X'6C') Triplet” on page 162
“Additional Bar Code Parameters (X'7B') Triplet” on page 163
“Graphics Descriptor (X'7E') Triplet” on page 164
“XML Name (X'8A') Triplet” on page 165
“Color Management Resource Descriptor (X'91') Triplet” on page 166
“**Concatenate Bar Code Data (X'93') Triplet**” on page 167
“Rendering Intent (X'95') Triplet” on page 168

Logical Page Eject Processing

See “Logical Page Eject Processing” on page 132, which describes page eject processing with a Record Descriptor (RCD); note that page eject processing for RCD and XMD is identical.

XMD Triplets

Fully Qualified Name (X'02') Triplet

This triplet is optional and may occur one or more times when a Bar Code Symbol Descriptor (X'69') triplet or a Graphics Descriptor (X'7E') triplet is specified on the XMD. The Fully Qualified Name type that may appear is X'DE'- *Data Object External Resource Reference*. The FQN triplet specifies the external identifier of a Color Management Resource (CMR) object that is used for the Bar Code object or Graphics object being generated. The identifier is used by the presentation system to locate the resource object in the resource hierarchy. The identifier is a character-encoded name that must be specified using FQNFmt = X'00'. The encoding for the external identifier of the CMR must be UTF-16BE.

See “Fully Qualified Name (X'02') Triplet” on page 107.

Extended Resource Local Identifier (X'22') Triplet

This triplet is optional and may occur one or more times to reference an IOB structured field in the PageDef. This triplet is ignored if the Graphics Descriptor (X'7E') triplet is specified on the XMD.

See “Extended Resource Local Identifier (X'22') Triplet” on page 108.

Color Specification (X'4E') Triplet

This is an optional triplet that specifies the color for text processed by this XMD, bar code generated by this XMD, and for graphics generated by this XMD.

See “Color Specification (X'4E') Triplet” on page 136.

Bar Code Symbol Descriptor (X'69') Triplet

This is an optional triplet and may occur once. If this triplet is specified more than once, only the first is used. This triplet specifies that the data selected by the descriptor is to be presented as a bar code symbol. This triplet is ignored if the Graphics Descriptor (X'7E') triplet is specified on the XMD.

See “Bar Code Symbol Descriptor (X'69') Triplet” on page 110. Note that the LND/RCD/XMD parameters used by this triplet may be at different offsets in the LND, RCD, and XMD.

Resource Object Include (X'6C') Triplet

This is an optional triplet that identifies an overlay or page segment object to be presented on the page at a specified position. Multiple Resource Object Include triplets may be specified on the same XMD. This triplet is ignored if the Graphics Descriptor (X'7E') triplet is specified on the XMD.

See “Resource Object Include (X'6C') Triplet” on page 116.

Additional Bar Code Parameters (X'7B') Triplet

This is an optional triplet that specifies additional parameters for non-linear bar code symbologies (for example, 2D bar codes). This triplet may occur one or more times when a Bar Code Symbol Descriptor (X'69') triplet is specified. This triplet is ignored if the Graphics Descriptor (X'7E') triplet is specified on the XMD.

See “Additional Bar Code Parameters (X'7B') Triplet” on page 118.

Graphics Descriptor (X'7E') Triplet

This is an optional Field XMD triplet and may occur once. If this triplet is specified more than once, only the first is used. Text input and fixed data text are ignored on a Field XMD that specifies a Graphics Descriptor triplet. When present, the Graphics Descriptor triplet specifies the generation of a graphics primitive. The triplet may specify the complete primitive, or the start of the primitive, or the end of the primitive. This triplet is ignored on XMDs other than Field XMDs.

See “Graphics Descriptor (X'7E') Triplet” on page 140.

XML Name (X'8A') Triplet

Architecture Note: The XML Name triplet is registered in MO:DCA as a private-use triplet since it is used only in the PageDef object, which is not a MO:DCA object.

This triplet is used to build a Qualified Tag. A Qualified Tag is built by concatenating the names specified on each XML Name triplet in the order the triplets are specified. Each XML Name used in the concatenation is separated by a single space character.

Note: Multiple XML Name triplets do not have to be contiguous.

This triplet is mandatory on Body and Group Header Element XMDs and must occur at least once to build a Qualified Tag.

This triplet is optional for Page Header and Page Trailer Element XMDs. If omitted, this Page Header or Page Trailer XMD is the default Page Header or Page Trailer XMD.

This triplet is mandatory on Attribute XMDs and must occur once to identify the name of an attribute specified on an XML start tag. If this triplet occurs more than once on an Attribute XMD, only the first occurrence is used.

This triplet is ignored on XMDs other than Element XMDs and Attribute XMDs.

The name specified in this triplet must be encoded using the encoding specified on the Encoding Scheme ID (X'50') triplet of the BDM structured field.

Triplet X'8A' Syntax

Offset	Type	Name	Range	Meaning	M/O
0	UBIN	Tlength	5–254	Length of the triplet, including Tlength	M
1	CODE	Tid	X'8A'	Identifies the XML Name triplet	M
2–3				Reserved; should be zero	M
4–*n*	CHAR	XMLName		Name of Start Tag or Attribute in XML data	M

Triplet X'8A' Semantics

Tlength Contains the length of the triplet

Tid Identifies the XML Name triplet

XMLName Specifies the name of the Start tag or the name of an attribute of a Start tag contained in the XML data

This XMLName is used to build Qualified Tags when used on Element XMDs.

Color Management Resource Descriptor (X'91') Triplet

The Color Management Resource Descriptor triplet specifies the processing mode and scope for a Color Management Resource (CMR). This triplet is mandatory when the XMD references a Color Management Resource (CMR) with the FQN type X'DE' triplet, in which case this triplet specifies the processing mode for the CMR and must occur once for each FQN type X'DE' specified. It is ignored in all other cases. This triplet must immediately follow the FQN type X'DE' triplet that specifies the CMR name.

See “Color Management Resource Descriptor (X'91') Triplet” on page 121.

Concatenate Bar Code Data (X'93') Triplet

This is an optional triplet and may occur once. If this triplet is specified more than once, only the first will be used.

In the XMD, completion of a given concatenated bar code symbol is defined as the end of the page or an XMD with the Start New Symbol flag (bit 0 of CBCFlgs) set in the X'93' triplet being reused.

See "Concatenate Bar Code Data (X'93') Triplet" on page 122.

Rendering Intent (X'95') Triplet

The Rendering Intent triplet specifies the rendering intent parameter, which is used to modify the final appearance of color data. This parameter is based on the rendering intents defined by the International Color Consortium (ICC). This triplet is optional and may occur once when a Graphics Descriptor (X'7E') triplet is specified on the XMD. If this triplet is specified more than once, only the first is used. This triplet specifies the rendering intent that is to be used when presenting the Graphics object that is generated with this structured field. Only the rendering intent that applies to the object type of the referenced object is used; the other rendering intents are ignored.

See “Rendering Intent (X'95') Triplet” on page 149.

Appendix A. Document and Resource Object Diagrams

This appendix contains diagrams of the elements that make up a data stream accepted by presentation services program servers in the AFP environment. Some presentation services programs might accept additional elements and objects. For the formal definition of all valid AFP (MO:DCA) object structures, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Unless otherwise noted, the objects and structured fields are presented from left to right in the order in which they must appear.

The following symbols apply to the syntax structures in this appendix:

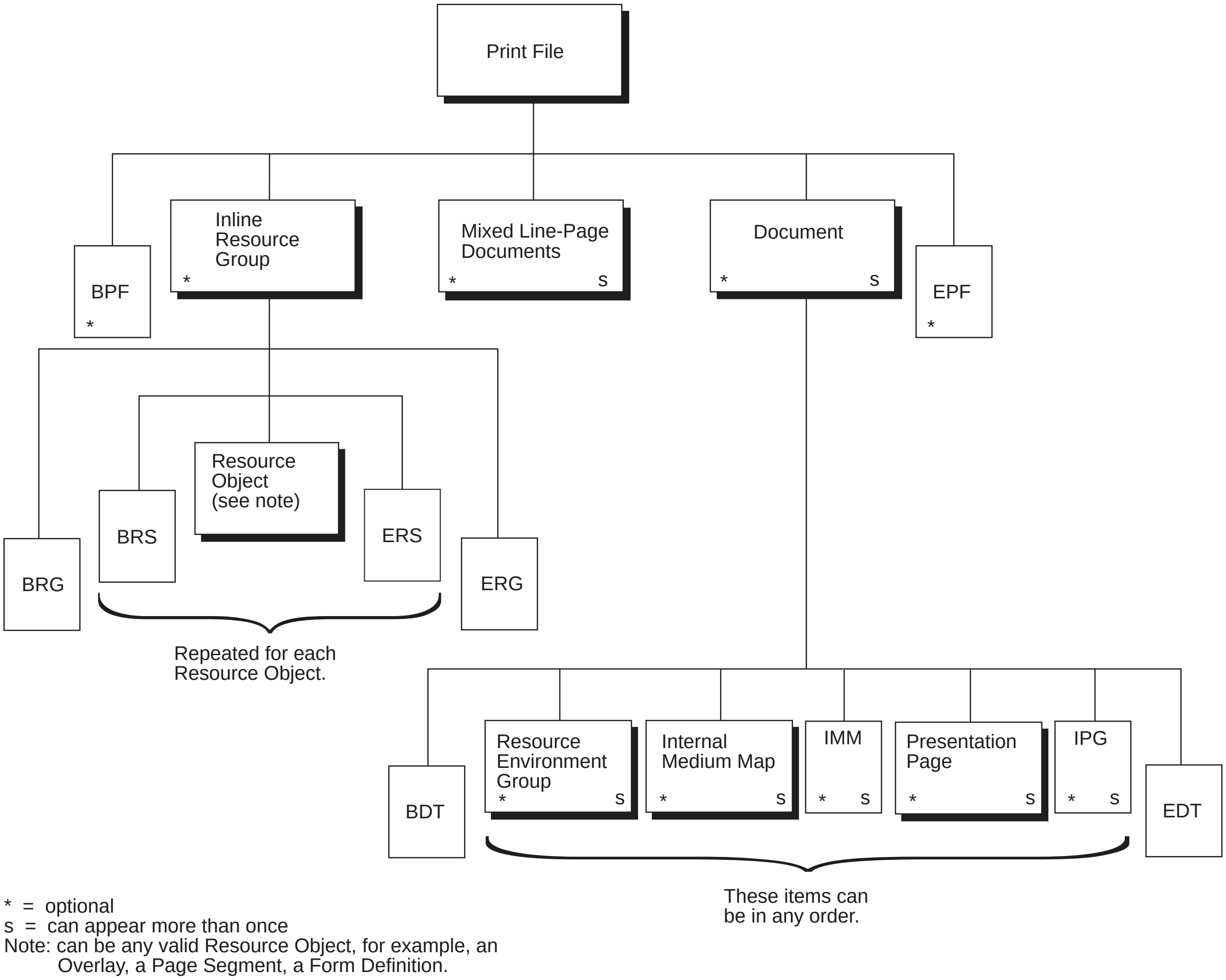
***** An asterisk indicates optional structured fields or objects. Those not marked are required.

S The letter “S” indicates structured fields or objects that can appear more than once in the document. Those not marked can appear only once in the document.

xx object A shaded box indicates objects that contain structured fields or other objects.

Other All other symbols are explained in the figures.

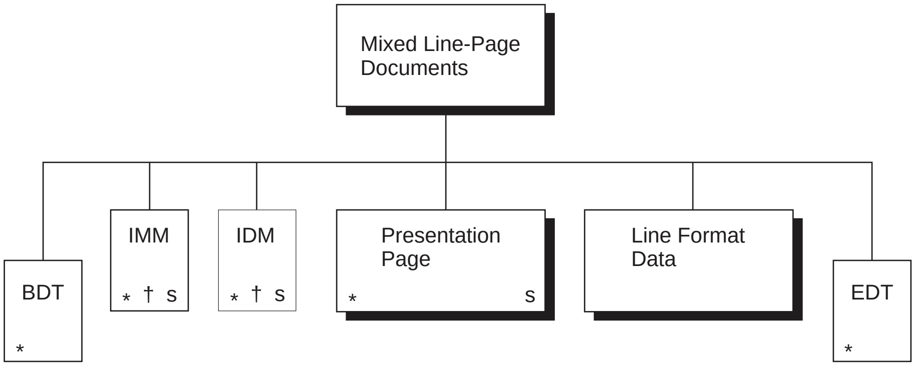
Figure 30. Structure of a Print File



Notes:

1. The BPF/EPF structured fields are optional as a pair; if one is specified, the other must be specified as well.
2. The mixed line-page documents and composed documents can occur in any order following the inline resource group.
3. Each AFP (MO:DCA) document may optionally be preceded by a single document index that is implicitly tied to the document and that indexes the document. For the formal definition of the MO:DCA document index see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.
4. An AFP (MO:DCA) document may contain Link Logical Element (LLE) structured fields following the BDT and may also group presentation pages into named page groups. MO:DCA page groups may in turn contain Tag Logical Element (TLE) structured fields following BNG. These structures do not affect the presentation of the document. For the formal definition of these structures, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.
5. If a Medium Map is included internal (inline) to the document, it is activated by immediately following it with an IMM that explicitly invokes it, otherwise the internal Medium Map is ignored. An IMM that does not follow an internal Medium Map may not invoke an internal Medium Map elsewhere in the document and is assumed to reference a Medium Map in the current Form Definition.

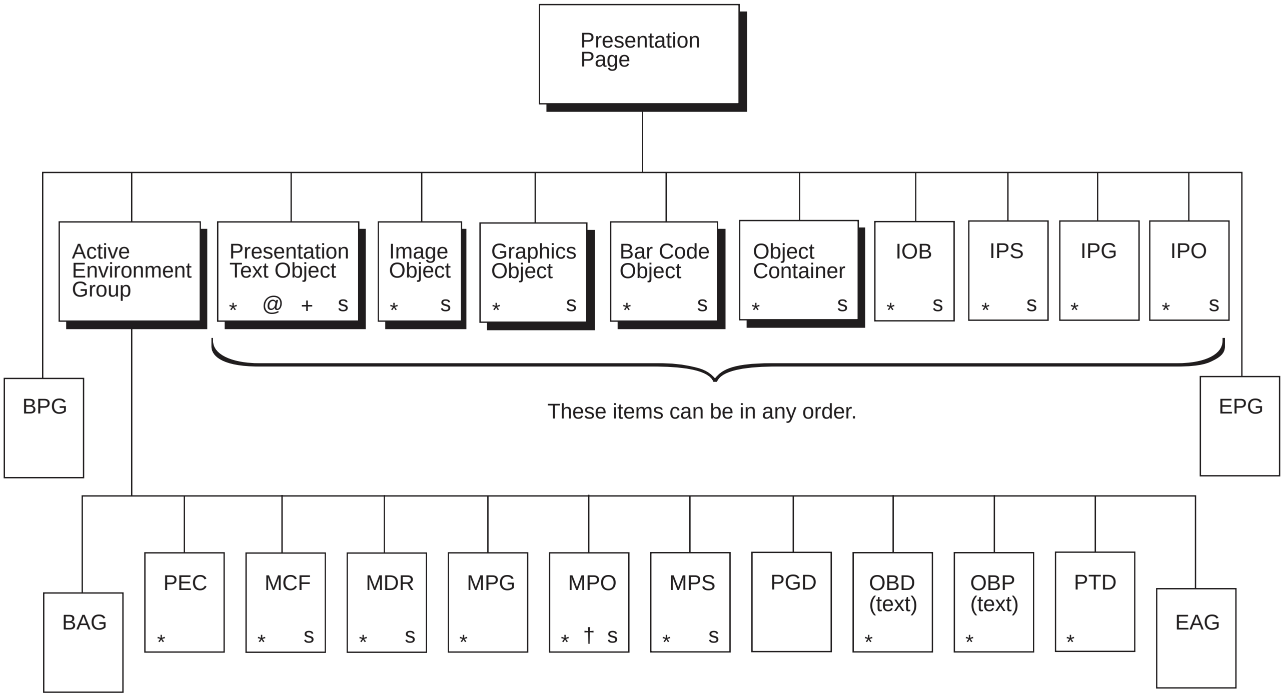
Figure 31. Structure of a Mixed Line-Page Document



* = optional
s = can appear more than once
† = an IMM or IDM may be specified before
any page; multiple IMMs and IDMs are allowed

Note: The No Operation (NOP) structured field may appear anywhere in a mixed document and thus is not listed in the structured field groupings.

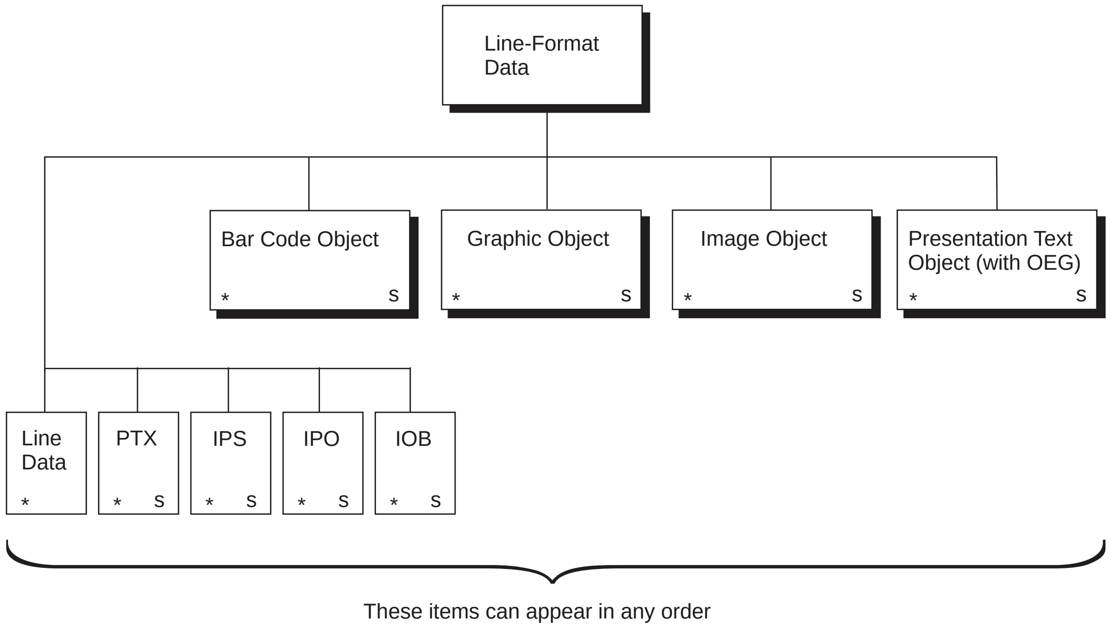
Figure 32. Structure of a Presentation Page Object



* = optional
s = can appear more than once
† = required for every IPO specified in a page
@ = without OEG
+ = with OEG

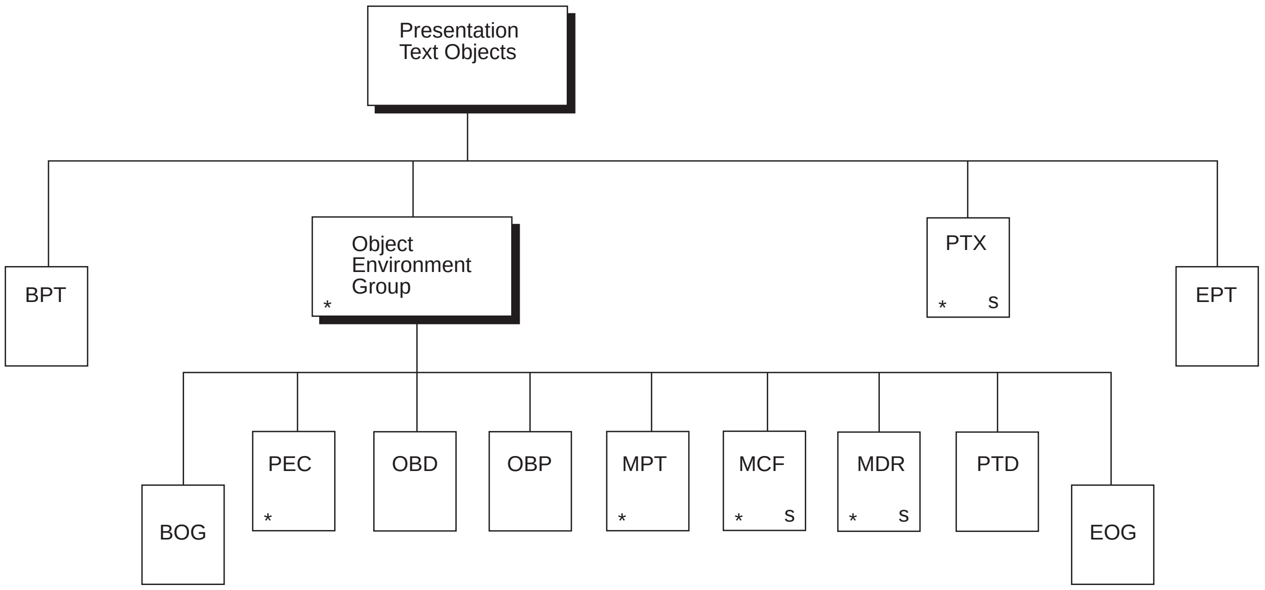
Note: An AFP (MO:DCA) presentation page can contain one or more Tag Logical Element (TLE) or Link Logical Element (LLE) structured fields following the AEG. These structures do not affect the presentation of the page. For the formal definition of these structures, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Figure 33. Structure of Line Format Data



* = optional
s = can appear more than once

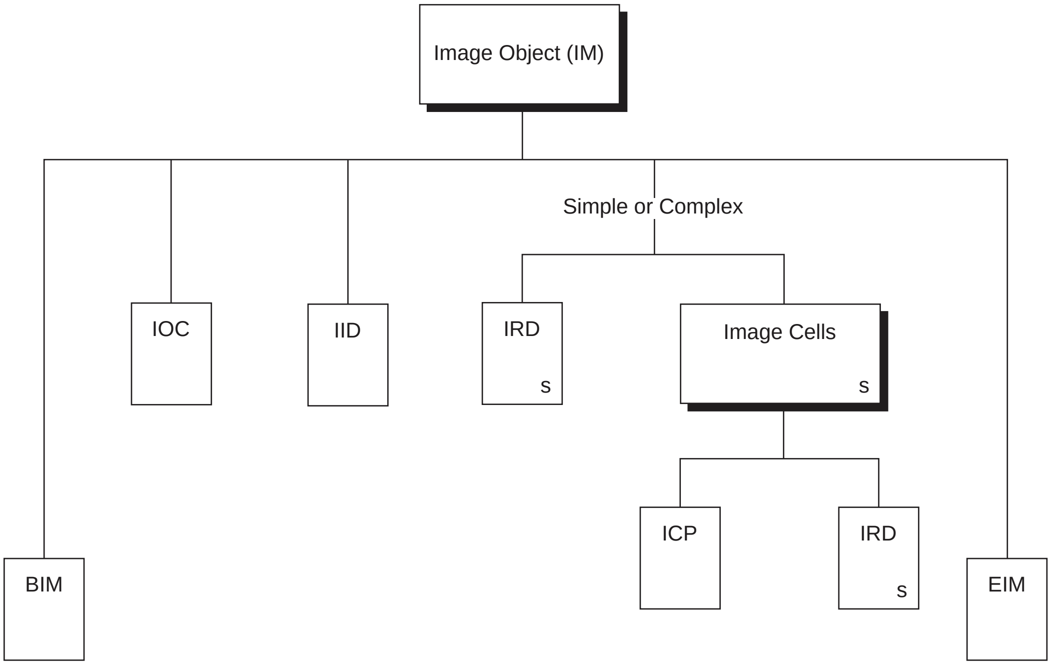
Figure 34. Structure of a Presentation Text Data Object



* = optional
s = can appear more than once

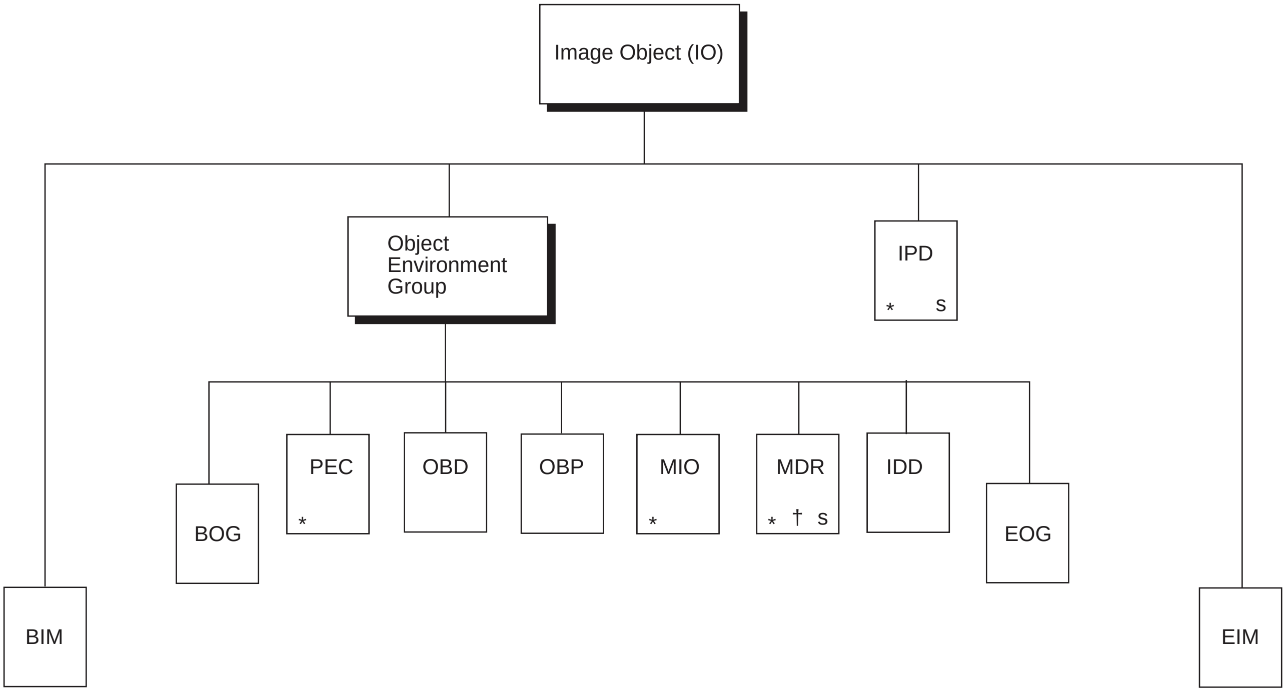
Note: A Presentation Text Descriptor is required in an Active Environment Group when a text object is used in a page.

Figure 35. Structure of an IM Image Data Object



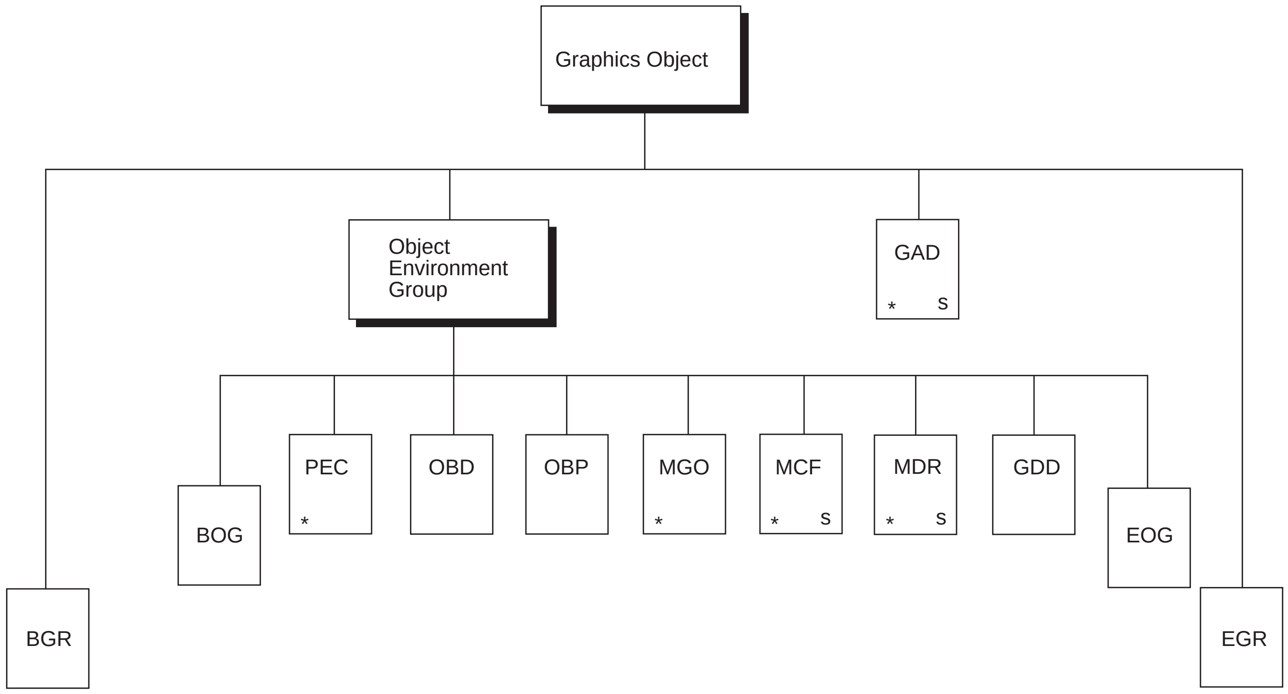
s = can appear more than once

Figure 36. Structure of an IO Image Data Object



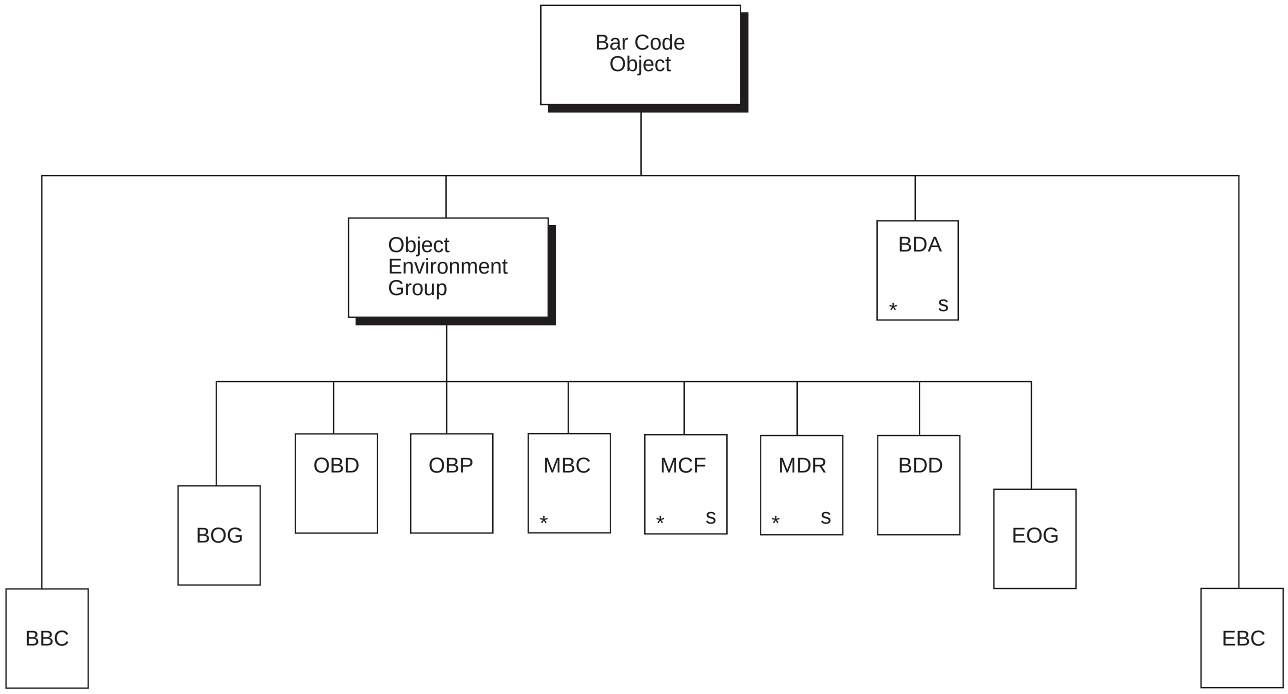
* = optional
s = can appear more than once
† = allowed in FS45 only

Figure 37. Structure of a Graphics Data Object



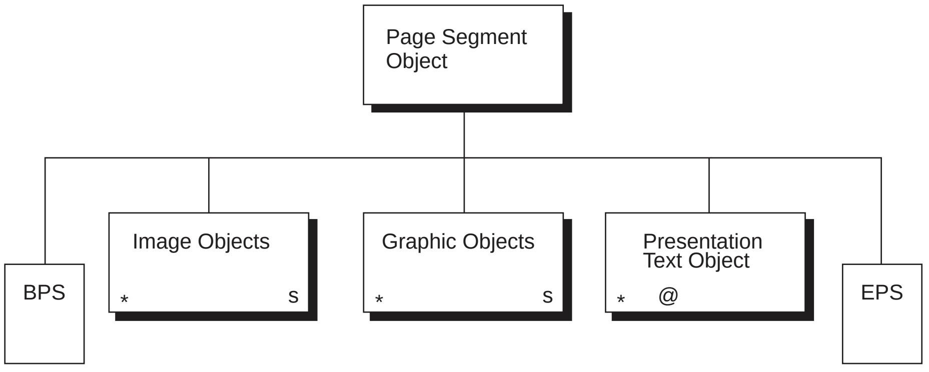
* = optional
s = can appear more than once

Figure 38. Structure of a Bar Code Data Object



* = optional
s = can appear more than once

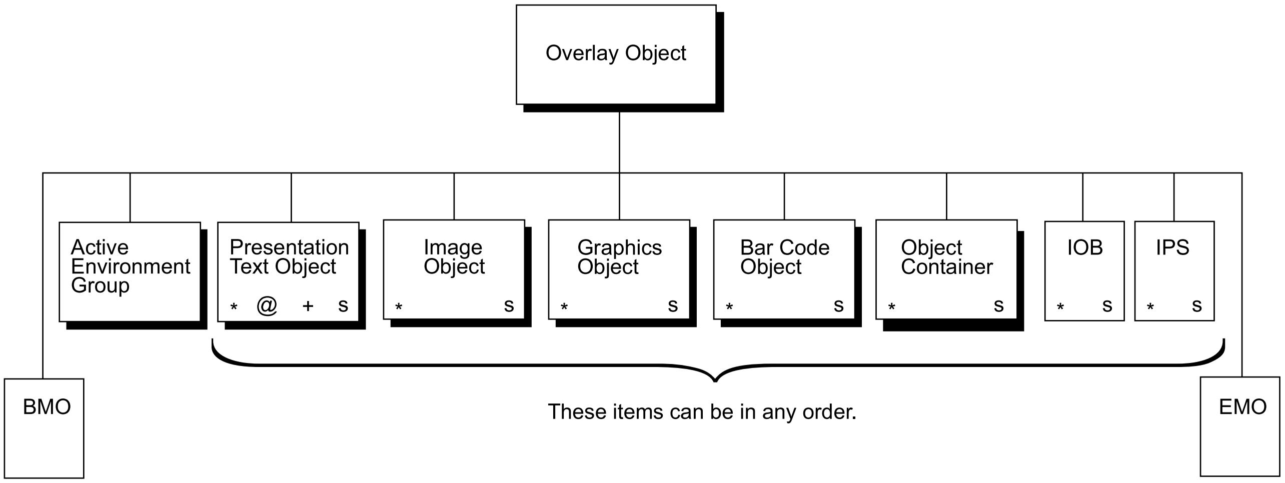
Figure 39. Structure of a Page Segment Resource Object



* = optional
s = can appear more than once
@ = without OEG

Note: This is the structure of an AFP page segment. This structure is supported but is replaced strategically with the MO:DCA page segment. For more information, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Figure 40. Structure of an Overlay Resource Object

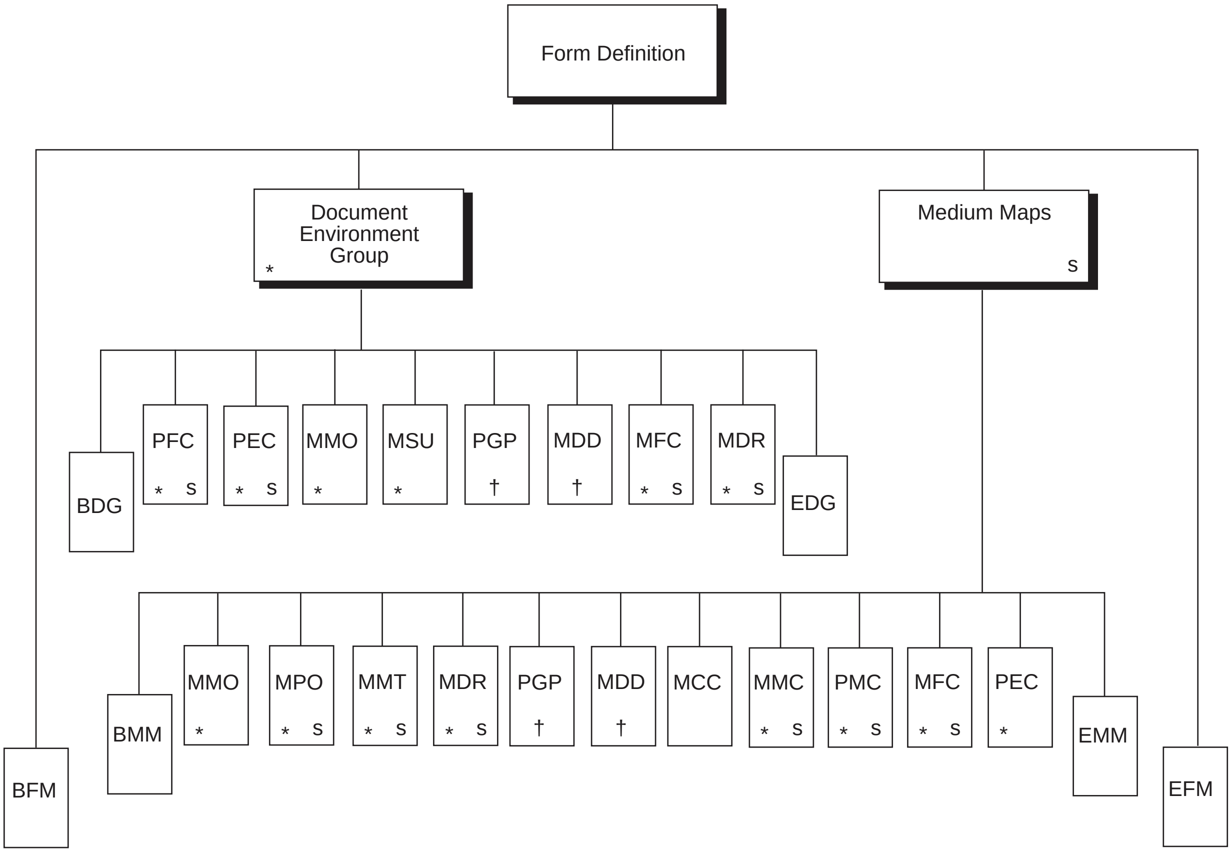


* = optional
s = can appear more than once
@ = without OEG
+ = with OEG

Notes:

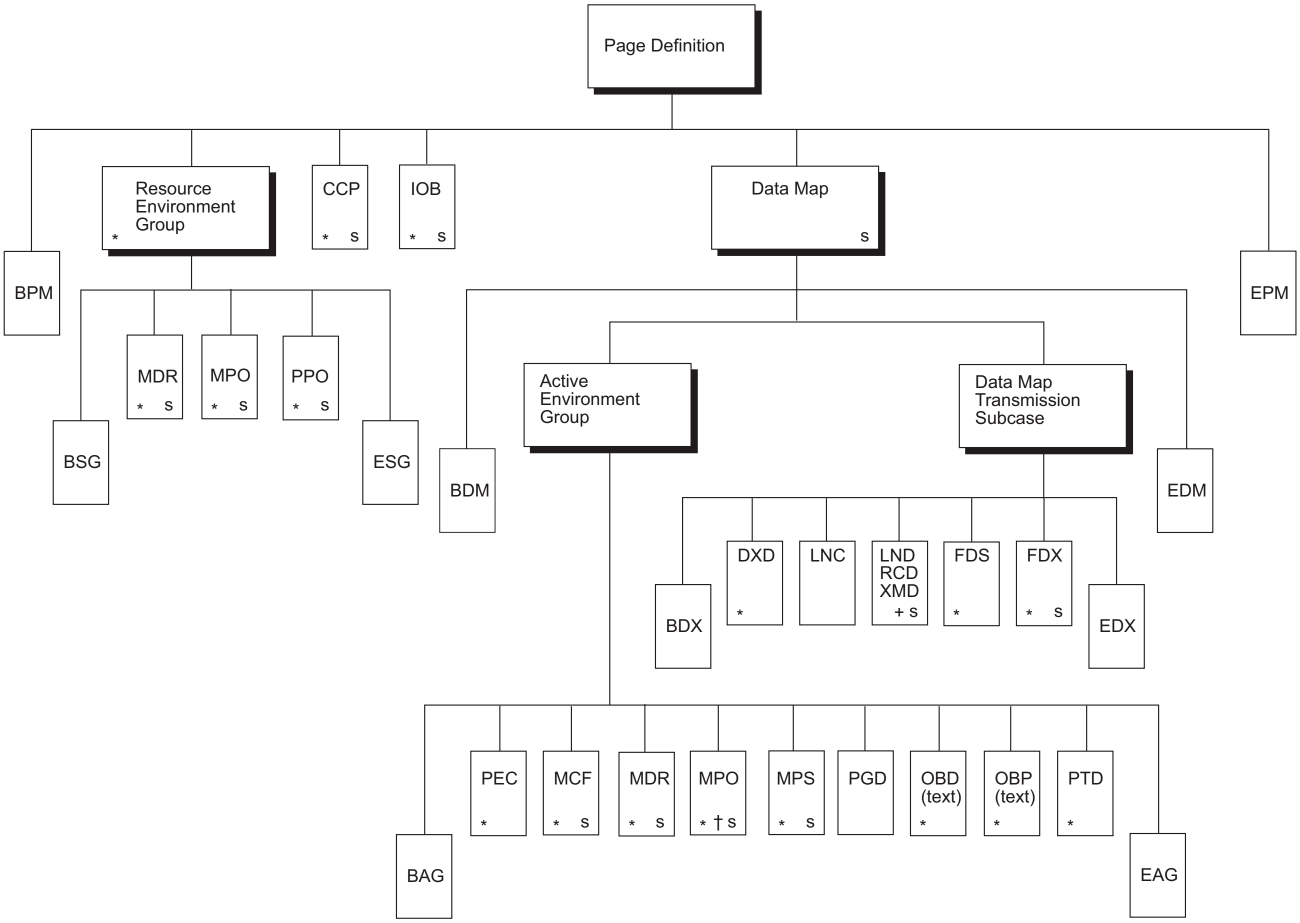
1. An AFP (MO:DCA) overlay object may contain one or more Tag Logical Element (TLE) or Link Logical Element (LLE) structured fields following the AEG. These structures do not affect the presentation of the overlay. For the formal definition of these structures, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*.
2. The MPG and MPO structured fields are not supported in the AEG for an overlay.

Figure 41. Structure of a Form Definition Resource Object



* = optional
S = can appear more than once
† = the structured field is required in either the Document Environment Group or the Medium Map Group

Figure 42. Structure of a Page Definition Resource Object



* = optional
s = can appear more than once
+ = the Data Map Transmission Subcase can contain LNDs, RCDs, or XMDs but not a mixture
† = required for every IPO specified in a page

Notes:

1. The Data Map Transmission Subcase may contain RCDs or XMDs instead of LNDs.
2. The Data Maps in a Page Definition must all contain LNDs, RCDs, or XMDs. A mixture is not allowed.
3. A Presentation Text Descriptor (PTD) is required in the AEG when a presentation text object is used on a page.

Appendix B. Cross-References

This appendix lists structured fields and PTOCA control sequences in alphabetical order by abbreviation name and in numerical order by hexadecimal code.

Structured Fields Arranged Alphabetically

This list includes not only the structured fields used in line-data and mixed data applications, but also the structured fields used in MO:DCA objects that are supported in AFP environments. It is possible that additional structured fields have been added to the MO:DCA architecture since this book was published; for a complete list of MO:DCA structured fields, refer to the *Mixed Object Document Content Architecture (MO:DCA) Reference*.

Table 15. Structured Fields Arranged Alphabetically

BAG	D3A8C9	Begin Active Environment Group
BBC	D3A8EB	Begin Bar Code Object
BCT	D3A89B	Begin Composed Text Object (renamed BPT)
BDA	D3EEEB	Bar Code Data
BDD	D3A6EB	Bar Code Data Descriptor
BDG	D3A8C4	Begin Document Environment Group
BDI	D3A8A7	Begin Document Index
BDM	D3A8CA	Begin Data Map
BDT	D3A8A8	Begin Document
BDX	D3A8E3	Begin Data Map Transmission Subcase
BFG	D3A8C5	Begin Form Environment Group (obsolete)
BFM	D3A8CD	Begin Form Map
BGR	D3A8BB	Begin Graphics Object
BII	D3A87B	Begin Image Object IM
BIM	D3A8FB	Begin Image Object IO
BMM	D3A8CC	Begin Medium Map
BMO	D3A8DF	Begin Overlay
BNG	D3A8AD	Begin Named Page Group
BOC	D3A892	Begin Object Container
BOG	D3A8C7	Begin Object Environment Group
BPF	D3A8A5	Begin Print File
BPG	D3A8AF	Begin Page
BPM	D3A8CB	Begin Page Map
BPS	D3A85F	Begin Page Segment
BPT	D3A89B	Begin Presentation Text Object

Table 15 Structured Fields Arranged Alphabetically (cont'd.)

BRG	D3A8C6	Begin Resource Group
BRS	D3A8CE	Begin Resource
BSG	D3A8D9	Begin Resource Environment Group
CCP	D3A7CA	Conditional Processing Control
CDD	D3A692	Container Data Descriptor
CTC	D3A79B	Composed Text Control (obsolete)
CTD	D3A69B	Composed Text Descriptor (renamed PTD Format 1)
CTX	D3EE9B	Composed Text Data (renamed PTX)
DXD	D3A6E3	Data Map Transmission Subcase Descriptor
EAG	D3A9C9	End Active Environment Group
EBC	D3A9EB	End Bar Code Object
ECT	D3A99B	End Composed Text Object (renamed EPT)
EDG	D3A9C4	End Document Environment Group
EDI	D3A9A7	End Document Index
EDM	D3A9CA	End Data Map
EDT	D3A9A8	End Document
EDX	D3A9E3	End Data Map Transmission Subcase
EFG	D3A9C5	End Form Environment Group (obsolete)
EFM	D3A9CD	End Form Map
EGR	D3A9BB	End Graphics Object
EII	D3A97B	End Image Object IM
EIM	D3A9FB	End Image Object IO
EMM	D3A9CC	End Medium Map
EMO	D3A9DF	End Overlay
ENG	D3A9AD	End Named Page Group
EOC	D3A992	End Object Container
EOG	D3A9C7	End Object Environment Group
EPF	D3A9A5	End Print File
EPG	D3A9AF	End Page
EPM	D3A9CB	End Page Map
EPS	D3A95F	End Page Segment
EPT	D3A99B	End Presentation Text
ERG	D3A9C6	End Resource Group
ERS	D3A9CE	End Resource
ESG	D3A9D9	End Resource Environment Group
FDS	D3AAEC	Fixed Data Size

Table 15 Structured Fields Arranged Alphabetically (cont'd.)

FDX	D3EEEC	Fixed Data Text
FGD	D3A6C5	Form Environment Group Descriptor (obsolete)
GAD	D3EEBB	Graphics Data
GDD	D3A6BB	Graphics Data Descriptor
ICP	D3AC7B	Image Cell Position
IDD	D3A6FB	Image Data Descriptor IO
IDM	D3ABCA	Invoke Data Map
IEL	D3B2A7	Index Element
IID	D3A67B	Image Input Descriptor IM
IMM	D3ABCC	Invoke Medium Map
IOB	D3AFC3	Include Object
IOC	D3A77B	Image Output Control IM
IPD	D3EEFB	Image Picture Data IO
IPG	D3AFAF	Include Page
IPO	D3AFD8	Include Page Overlay
IPS	D3AF5F	Include Page Segment
IRD	D3EE7B	Image Raster Data IM
LIN	————	Line Record
Note: Line Record is not a structured field. However, the abbreviation “LIN” can appear as a variable insert in a number of presentation services program messages that can also contain structured field abbreviations.		
LLE	D3B490	Link Logical Element
LNC	D3AAE7	Line Descriptor Count
LND	D3A6E7	Line Descriptor
MBC	D3ABEB	Map Bar Code
MCC	D3A288	Medium Copy Count
MCD	D3AB92	Map Container Data
MCF-1	D3B18A	Map Coded Font (Format 1)
MCF-2	D3AB8A	Map Coded Font (Format 2)
MDD	D3A688	Medium Descriptor
MDR	D3ABC3	Map Data Resource
MFC	D3A088	Medium Finishing Control
MGO	D3ABBB	Map Graphic Object
MIO	D3ABFB	Map IO Image Object
MMC	D3A788	Medium Modification Control
MMD	D3ABCD	Map Media Destination
MMO	D3B1DF	Map Medium Overlay

Table 15 Structured Fields Arranged Alphabetically (cont'd.)

MMT	D3AB88	Map Media Type
MPG	D3ABAF	Map Page
MPO	D3ABD8	Map Page Overlay
MPS	D3B15F	Map Page Segment
MPT	D3AB9B	Map Presentation Text
MSU	D3ABEA	Map Suppression
NOP	D3EEEE	No Operation
OBD	D3A66B	Object Area Descriptor
OBP	D3AC6B	Object Area Position
OCD	D3EE92	Object Container Data
PEC	D3A7A8	Presentation Environment Control
PFC	D3B288	Presentation Fidelity Control
PGD	D3A6AF	Page Descriptor
PGP-1	D3ACAF	Page Position (Format 1)
PGP-2	D3B1AF	Page Position (Format 2)
PMC	D3A7AF	Page Modification Control
PPO	D3ADC3	Preprocess Presentation Object
PTD-1	D3A69B	Presentation Text Descriptor (Format 1)
PTD-2	D3B19B	Presentation Text Descriptor (Format 2)
PTX	D3EE9B	Presentation Text Data
RCD	D3A68D	Record Descriptor
TLE	D3A090	Tag Logical Element
XMD	D3A68E	XML Descriptor

Structured Fields Arranged Numerically by Hexadecimal Code

Table 16. Structured Fields Arranged Numerically by Hexadecimal Code

D3A088	MFC	Medium Finishing Control
D3A090	TLE	Tag Logical Element
D3A288	MCC	Medium Copy Count
D3A66B	OBD	Object Area Descriptor
D3A67B	IID	Image Input Descriptor
D3A688	MDD	Medium Descriptor
D3A68D	RCD	Record Descriptor
D3A68E	XMD	XML Descriptor
D3A692	CDD	Container Data Descriptor
D3A69B	CTD	Composed Text Descriptor (renamed PTD Format 1)
D3A69B	PTD-1	Presentation Text Descriptor (Format 1)
D3A6AF	PGD	Page Descriptor
D3A6BB	GDD	Graphics Data Descriptor
D3A6C5	FGD	Form Environment Group Descriptor (obsolete)
D3A6E3	DXD	Data Map Transmission Subcase Descriptor
D3A6E7	LND	Line Descriptor
D3A6EB	BDD	Bar Code Data Descriptor
D3A6FB	IDD	Image Data Descriptor IO
D3A77B	IOC	Image Output Control
D3A788	MMC	Medium Modification Control
D3A79B	CTC	Composed Text Control (obsolete)
D3A7A8	PEC	Presentation Environment Control
D3A7AF	PMC	Page Modification Control
D3A7CA	CCP	Conditional Processing Control
D3A85F	BPS	Begin Page Segment
D3A87B	BII	Begin Image Block
D3A892	BOC	Begin Object Container
D3A89B	BCT	Begin Composed Text Object (renamed BPT)
D3A89B	BPT	Begin Presentation Text Object
D3A8A5	BPF	Begin Print File
D3A8A7	BDI	Begin Document Index
D3A8A8	BDT	Begin Document
D3A8AD	BNG	Begin Named Page Group

Table 16 Structured Fields Arranged Numerically by Hexadecimal Code (cont'd.)

D3A8AF	BPG	Begin Page
D3A8BB	BGR	Begin Graphics Object
D3A8C4	BDG	Begin Document Environment Group
D3A8C5	BFG	Begin Form Environment Group (obsolete)
D3A8C6	BRG	Begin Resource Group
D3A8C7	BOG	Begin Object Environment Group
D3A8C9	BAG	Begin Active Environment Group
D3A8CA	BDM	Begin Data Map
D3A8CB	BPM	Begin Page Map
D3A8CC	BMM	Begin Medium Map
D3A8CD	BFM	Begin Form Map
D3A8CE	BRS	Begin Resource
D3A8D9	BSG	Begin Resource Environment Group
D3A8DF	BMO	Begin Overlay
D3A8E3	BDX	Begin Data Map Transmission Subcase
D3A8EB	BBC	Begin Bar Code Object
D3A8FB	BIM	Begin Image Object IO
D3A95F	EPS	End Page Segment
D3A97B	EII	End Image Block IM
D3A992	EOC	End Object Container
D3A99B	ECT	End Composed Text Object (renamed EPT)
D3A99B	EPT	End Presentation Text Object
D3A9A5	EPF	End Print File
D3A9A7	EDI	End Document Index
D3A9A8	EDT	End Document
D3A9AD	ENG	End Named Page Group
D3A9AF	EPG	End Page
D3A9BB	EGR	End Graphics Object
D3A9C4	EDG	End Document Environment Group
D3A9C5	EFG	End Form Environment Group (obsolete)
D3A9C6	ERG	End Resource Group
D3A9C7	EOG	End Object Environment Group
D3A9C9	EAG	End Active Environment Group
D3A9CA	EDM	End Data Map
D3A9CB	EPM	End Page Map
D3A9CC	EMM	End Medium Map

Table 16 Structured Fields Arranged Numerically by Hexadecimal Code (cont'd.)

D3A9CD	EFM	End Form Map
D3A9CE	ERS	End Resource
D3A9D9	ESG	End Resource Environment Group
D3A9DF	EMO	End Overlay
D3A9E3	EDX	End Data Map Transmission Subcase
D3A9EB	EBC	End Bar Code Object
D3A9FB	EIM	End Image Object IO
D3AAE7	LNC	Line Descriptor Count
D3AAEC	FDS	Fixed Data Size
D3AB88	MMT	Map Media Type
D3AB8A	MCF-2	Map Coded Font (Format 2)
D3AB92	MCD	Map Container Data
D3AB9B	MPT	Map Presentation Text
D3ABAF	MPG	Map Page
D3ABBB	MGO	Map Graphic Object
D3ABC3	MDR	Map Data Resource
D3ABCA	IDM	Invoke Data Map
D3ABCC	IMM	Invoke Medium Map
D3ABCD	MMD	Map Media Destination
D3ABD8	MPO	Map Page Overlay
D3ABEA	MSU	Map Medium Suppression
D3ABEB	MBC	Map Bar Code
D3ABFB	MIO	Map IO Image Object
D3AC6B	OBP	Object Area Position
D3AC7B	ICP	Image Cell Position
D3ACAF	PGP-1	Page Position (Format 1)
D3ADC3	PPO	Preprocess Presentation Object
D3AF5F	IPS	Include Page Segment
D3AFAF	IPG	Include Page
D3AFC3	IOB	Include Object
D3AFD8	IPO	Include Page Overlay
D3B15F	MPS	Map Page Segment
D3B18A	MCF-1	Map Coded Font (Format 1)
D3B19B	PTD-2	Presentation Text Descriptor (Format 2)
D3B1AF	PGP-2	Page Position (Format 2)

Table 16 Structured Fields Arranged Numerically by Hexadecimal Code (cont'd.)

D3B1DF	MMO	Map Medium Overlay
D3B288	PFC	Presentation Fidelity Control
D3B2A7	IEL	Index Element
D3B490	LLE	Link Logical Element
D3EE7B	IRD	Image Raster Data
D3EE92	OCD	Object Container Data
D3EE9B	CTX	Composed Text Data (renamed PTX)
D3EE9B	PTX	Presentation Text Data
D3EEBB	GAD	Graphics Data
D3EEEB	BDA	Bar Code Data
D3EEEC	FDX	Fixed Data Text
D3EEEE	NOP	No Operation
D3EEFB	IPD	Image Picture Data IO

PTOCA Control Sequences Arranged Alphabetically

This is a list in alphabetical order by abbreviation name of text control sequences that can appear in the presentation text (PTX) structured field. An even function-type code indicates that the control sequence is followed by code points or by an unchained control sequence. An odd function-type code indicates that the control sequence is followed by a chained control sequence. Chaining control sequences reduces the number of bytes in the PTX structured field by eliminating the two-byte prefix and class code from each chained control sequence. It is possible that additional control sequences have been added to the PTOCA architecture since this book was published; for a complete list and for details of these control sequences, refer to the *Presentation Text Object Content Architecture Reference*.

Table 17. PTOCA Control Sequences Arranged Alphabetically

AMB	D2(D3)	Absolute Move Baseline
AMI	C6(C7)	Absolute Move Inline
BLN	D8(D9)	Begin Line
BSU	F2(F3)	Begin Suppression
DBR	E6(E7)	Draw Baseline Rule
DIR	E4(E5)	Draw Inline Rule
ESU	F4(F5)	End Suppression
GAR	8C(8D)	Glyph Advance Run
GIR	–(8B)	Glyph ID Run
GLC	–(6D)	Glyph Layout Control
GOR	8E(8F)	Glyph Offset Run
NOP	F8(F9)	No Operation
OVS	72(73)	Overstrike
RMB	D4(D5)	Relative Move Baseline
RMI	C8(C9)	Relative Move Inline
RPS	EE(EF)	Repeat String
SBI	D0(D1)	Set Baseline Increment
SCFL	F0(F1)	Set Coded Font Local
SEC	80(81)	Set Extended Text Color
SIA	C2(C3)	Set Intercharacter Adjustment
SIM	C0(C1)	Set Inline Margin
STC	74(75)	Set Text Color
STO	F6(F7)	Set Text Orientation
SVI	C4(C5)	Set Variable Space Character Increment
TBM	78(79)	Temporary Baseline Move
TRN	DA(DB)	Transparent Data

Table 17 PTOCA Control Sequences Arranged Alphabetically (cont'd.)

UCT	6A(–)	Unicode Complex Text
USC	76(77)	Underscore

PTOCA Control Sequences Arranged Numerically

These are the PTOCA control sequences listed in numerical order by hexadecimal code.

6A(–)	UCT	Unicode Complex Text
–(6D)	GLC	Glyph Layout Control
72(73)	OVS	Overstrike
74(75)	STC	Set Text Color
76(77)	USC	Underscore
78(79)	TBM	Temporary Baseline Move
80(81)	SEC	Set Extended Text Color
–(8B)	GIR	Glyph ID Run
8C(8D)	GAR	Glyph Advance Run
8E(8F)	GOR	Glyph Offset Run
C0(C1)	SIM	Set Inline Margin
C2(C3)	SIA	Set Intercharacter Adjustment
C4(C5)	SVI	Set Variable Space Character Increment
C6(C7)	AMI	Absolute Move Inline
C8(C9)	RMI	Relative Move Inline
D0(D1)	SBI	Set Baseline Increment
D2(D3)	AMB	Absolute Move Baseline
D4(D5)	RMB	Relative Move Baseline
D8(D9)	BLN	Begin Line
DA(DB)	TRN	Transparent Data
E4(E5)	DIR	Draw Inline Rule
E6(E7)	DBR	Draw Baseline Rule
EE(EF)	RPS	Repeat String
F0(F1)	SCFL	Set Coded Font Local
F2(F3)	BSI	Begin Suppression
F4(F5)	ESU	End Suppression
F6(F7)	STO	Set Text Orientation
F8(F9)	NOP	No Operation

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 changes in the architecture described in this publication at any time without notice.

Any references in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this architecture and use of those Web sites is at your own risk.

The AFP Consortium may use or distribute any information you supply in any way it believes appropriate without incurring any obligation to you.

This information contains examples of data and reports used in daily business operations. To illustrate them in a complete manner, some examples include the names of individuals, companies, brands, or products. These names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

These terms are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries:

Acrobat
Adobe
Photoshop
PostScript

These terms are trademarks of the AFP Consortium:

AFPC
AFP Consortium

These terms are registered trademarks of Hewlett-Packard Company in the United States of America and other countries/regions:

Hewlett-Packard
PCL

These terms are trademarks or registered trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX
GDDM
IBM
MVS
MVS/ESA
Operating System/2
OS/390
OS/400
Print Services Facility
System i5
z/OS
z/VM
z/VSE

Linux is a trademark of Linus Torvalds.

These terms are registered trademarks of Microsoft Corporation in the United States, other countries, or both:

Microsoft
Windows

PANTONE is a registered trademark of Pantone LLC.

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, other countries, or both:

ACMA
Advanced Function Presentation
AFP
AFPCC
AFP Color Consortium
AFP Color Management Architecture
Bar Code Object Content Architecture
BCOCA
CMOCA

Color Management Object Content Architecture
InfoPrint
Intelligent Printer Data Stream
IPDS
Mixed Object Document Content Architecture
MO:DCA
Ricoh

Intelligent Mail is a registered trademark of the United States Postal Service.

UP^{3}I is a trademark of UP^{3}I Limited.

Other company, product, or service names might be the 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 line-data terms or 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 Consortium (AFPC). A formal open standards body that develops and maintains AFP architecture. Information about the consortium can be found at `www.afpcinc.org`.

AFP data stream. A presentation data stream that is processed in AFP environments. The MO:DCA architecture defines the strategic AFP interchange data stream. The IPDS architecture defines the strategic AFP printer data stream.

AFPDS. A term formerly used to identify the composed-page MO:DCA-based data stream interchanged in AFP environments. See also MO:DCA and AFP data stream.

AFP GOCA. 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).

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.

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 which 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.

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 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-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 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.

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 which 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. A collection of architecture constructs and associated values. Function sets can be defined across or within subsets.

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 which 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 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, which 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.

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.

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 device-dependent manner. Synonymous with medium presentation space. See also logical page, physical medium, and presentation space.

Medium Map. A print control object in a Form Map that defines resource mappings and controls modifications to a form, page placement on a form, and form copy generation. See also Form Map.

medium 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 device-dependent manner. Synonymous with medium. See also logical page, physical medium, and presentation space.

metadata. Descriptive information that is associated with and augments other data.

metadata object. 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, device-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 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.afpcinc.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.

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 device-dependent default color, typically white on a screen and black on a printer. Note that neutral white and color of medium are two different colors.

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 device-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 which permits linking of multiple font files and formats together. The code points conform to a particular character coding system which 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. 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.

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 device-independent, self-defining representation of a two-dimensional presentation space, called the text object space, which 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.

UP^{3}I. Universal Printer Pre- and Post-Processing Interface; an industry standard interface designed for use in complex printing systems. A specification for this interface can be obtained at `www.afpcinc.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

Advanced Function Presentation Consortium

AFP Programming Guide and Line Data Reference

AFPC-0010-05

AFPConsortium
Advanced Function Presentation