Advanced Function Presentation Consortium
Data Stream and Object Architectures

Color Management Object Content Architecture Reference

AFPC-0006-01

Note:
Before using this information, read the information in "Notices" on page 135.

AFPC-0006-01
Second Edition (April 2012)

This edition applies to the Color Management Object Content Architecture (CMOCA). It is the first edition produced by the AFP Consortium and replaces and makes obsolete the previous edition (S550-0511-00) published by IBM. This edition remains current until a new edition is published. This publication also applies to any subsequent releases of Advanced Function Presentation (AFP) products that use the CMOCA architecture until otherwise indicated in a new edition.

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

Internet
Visit our home page: `http://www.afpcinc.org`

Preface

This book describes the architecture that has been approved for the Color Management Object Content Architecture™ (CMOCA™).

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

Who Should Read This Book

This book is for system programmers and other developers who need such information to develop or adapt a product or program to interoperate with other presentation products in an Advanced Function Presentation™ (AFP™) environment.

AFP Consortium

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. In February 2009, the consortium was incorporated under a new set of bylaws with tiered membership and shared governance resulting in the creation of a formal open standards body called the AFP Consortium™ (AFPC™). Ownership of and responsibility for the AFP architectures was transferred at that time to the AFP Consortium.

Publication History

The CMOCA Reference was first published by IBM in 2006 and has had several enhancements and updates since that time.

First Edition published by IBM Corporation
S550-0511-00 dated May 2006

How to Use This Book

This document is divided into six chapters and three appendixes:

- Chapter 1, "A Presentation Architecture Perspective," on page 1 introduces the AFP presentation architectures and describes the role of data streams and data objects.
- Chapter 2, "Introduction to CMOCA," on page 7 introduces the goals and purposes of the Color Management Object Content Architecture.
- Chapter 3, "Color Management Resource (CMR)," on page 9 discusses the format of the Color Management Resource (CMR) header and how CMRs are used to process data.
- Chapter 4, "CMR Types," on page 25 defines each type of CMR.
- Chapter 5, "CMR Data Architecture," on page 37 defines tag syntax and semantics.
- Chapter 6, "CMR Processing," on page 103 discusses how search paths are used to determine which CMRs to use, audit and instruction and link CMRs, generic vs. device-specific CMRs, and implications for drivers.
- Appendix A, "Tag Registry," on page 125 lists the tags that devices receiving CMRs must support.
- Appendix B, "Generic CMR Name Registry," on page 127 lists and explains all the registered generic CMR names.
- Appendix C, "Compliance with Color Management Object Content Architecture," on page 133 explains what is required to claim CMOCA support.

The "Glossary" on page 137 defines some of the terms used within this book.

Interpreting the Syntax

The basic data types used in the Color Management Object Content Architecture (CMOCA) are:

CODE Architected constant
BITS Bit string
UBIN Unsigned binary
BYTE 8 bits
ASCII ASCII-encoded characters
UTF16 UTF-16BE characters (2-bytes each)
UNDF Undefined data type

The following notation conventions apply to the CMR data structures.

- Each byte contains eight bits.

- Bytes of a CMR data structure are numbered from left to right beginning with byte 0. The left-most byte is the most significant; this is called big endian. For example, a two-byte field followed by a one-byte field would be numbered as follows:

  Bytes 0–1 Field 1
  Byte 2 Field 2
- Bit strings are numbered from left to right beginning with 0. For example, a one-byte bit string contains bit 0, bit 1, ..., bit 7.
- For numerical binary data, bit 0 is the most significant bit.
- Field values are expressed in hexadecimal or binary notation:

  X'7FFF' = +32,767
  B'0001' = 1
- Some bits or bytes are labeled *reserved*. The content of reserved fields is not checked by CMR receivers. However, CMR generators should set reserved fields to the specified value, if one is given, or to zero.
- Some fields or values are labeled *Retired item n*, where *n* is an identifying number. These fields or values are reserved for a particular purpose and must not be used for any other purpose.

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.

Table 1. AFP Consortium Architecture References

AFP Architecture Publication	Order Number
AFP Programming Guide and Line Data Reference	S544-3884
Bar Code Object Content Architecture Reference	AFPC-0005
Color Management Object Content Architecture Reference	AFPC-0006
Font Object Content Architecture Reference	S544-3285
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
Mixed Object Document Content Architecture Reference	AFPC-0004
Presentation Text Object Content Architecture Reference	SC31-6803

Table 2. Additional AFP Consortium Documentation

AFPC Publication	Order Number
AFP Color Management Architecture (ACMA)	AFPC-0015
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

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	SC09-2190
Font Summary for AFP Font Collection	S544-5633
Technical Reference for Code Pages	S544-3802

Table 4. UP^3I Architecture Documentation

UP^3I Publication	Order Number
Universal Printer Pre- and Post-Processing Interface (UP^3I) Specification	Available at the UP^3I website

Changes in This Edition

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

This edition provides enhanced detail to support the CMOCA products that were introduced in the years 2006 through 2011 and to support the work of the AFP Consortium. Specifically, the following new function and clarification has been provided:

- Expanded range of valid values for MediaFinish and MediaColor and provide AFPC suggested values
- ICC DeviceLink CMR
- ICC Profile Filename tag
- Indexed CMR refinements including new exception codes X'12' and X'13'
- Media finish values for coated, commodity, newsprint, and treated media
- Partial Support of TTC & HT CMRs
- Passthrough CMR
- Registered AFP-Consortium-provided standard audit CMRs
- Removed Color Conversion CMR subset X'0A' (Abstract Profile)
- Clarifications for:
  - Array Width and Array Height tag descriptions
  - Custom values for media-finish and media-color fields in the CMR header
  - Default audit CMRs color-space descriptions for grayscale and YCbCr/YCrCb
  - Named and highlight colors plus a recommendation for OCA black
  - Padding and use of device-specific values in the CMR header
  - Removed EC-xxxx05 (Invalid Count Value) exception conditions that were covered by EC-xxxx11 (Inconsistent Tag Contents)

Contents

Figures

Tables

Chapter 1. A Presentation Architecture Perspective

This chapter provides a brief overview of Presentation Architecture.

The Presentation Environment

Figure 1 shows today's presentation environment.

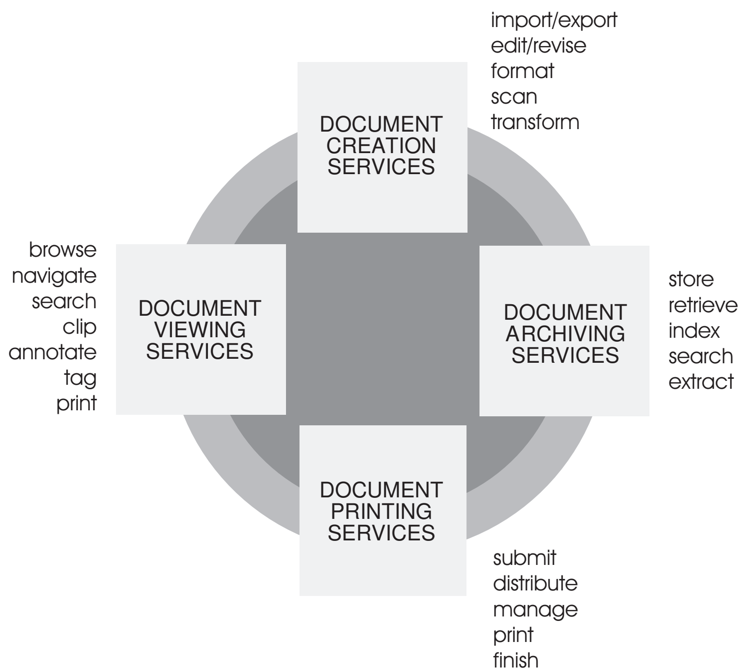


Figure 1. Presentation Environment. The environment is a coordinated set of services architected to meet the presentation needs of today's applications.

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

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

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

Architecture Components

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

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

Data Streams

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

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

The MO:DCA architecture defines the data stream used by applications to describe documents and object envelopes for interchange with other applications and application services. Documents defined in the MO:DCA format can be archived in a database, then later retrieved, viewed, annotated, and printed in local or distributed systems environments. Presentation fidelity is accommodated by including resource objects in the documents that reference them.

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

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

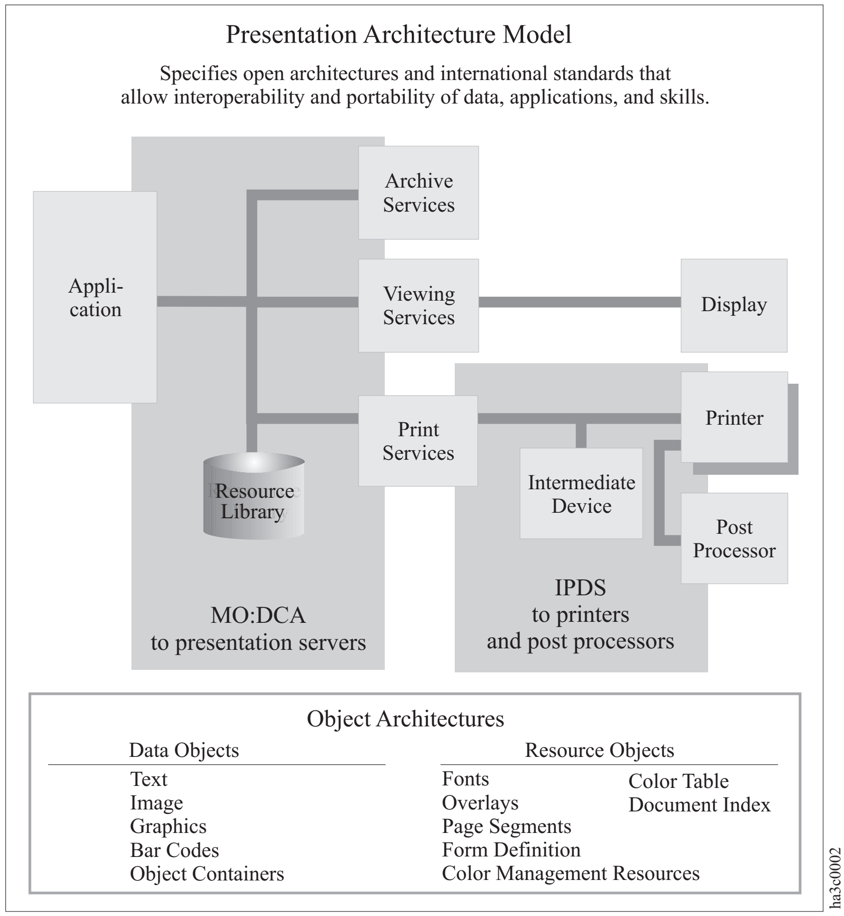


Figure 2. Presentation Model. This diagram shows the major components in a presentation system and their use of data stream and object architectures.

Objects

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

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

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

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

The AFP object content architectures are:

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

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

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

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

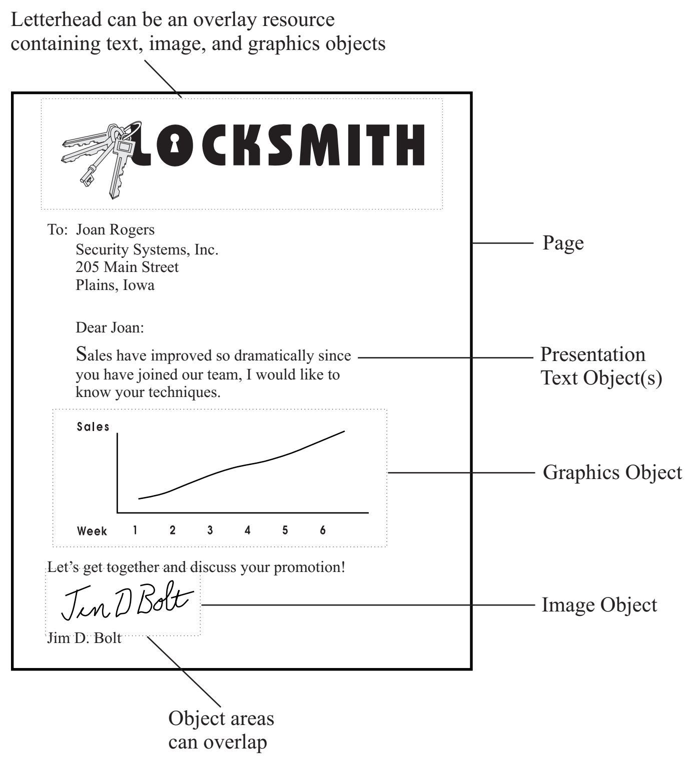


Figure 3. Presentation Page. This is an example of a mixed-object page that can be composed in a device-independent MO:DCA format and printed on an IPDS printer.

Chapter 2. Introduction to CMOCA

The Color Management Object Content Architecture (CMOCA) defines objects that provide color management in presentation environments. These objects are called Color Management Resources (CMRs). The CMOCA has the following objectives:

- Consistent output across different devices
- Accurate output, to the best of the device capability, from a wide variety of inputs provided that the input color information is properly described
- Consistent output across different data streams
- Flexible controls that enable customers to obtain output to their exact specifications

The architecture described in this document is defined in the terms of the AFP architectures to support color management in AFP environments, but care has been taken to make the mechanisms applicable to other presentation environments as well.

The device that presents the data could be a printer, a display or other system. This document frequently references printers but it should be understood that the architecture also applies to displays and other presentation devices.

A Color Management Resource (CMR) is an architected resource that is used to carry the color management information required to render a print file, document, page, or data object. Each CMR carries a single type of color management resource. There are five types of CMRs:

1. Halftone
2. Tone Transfer Curve
3. Color Conversion
4. Link Color Conversion
5. Indexed

Note: Not all CMR types are applicable for a particular kind of presentation device; for instance, halftones are not applicable for a display.

A CMR can reflect processing that has been done on an object, in which case it is referred to as an *audit* CMR, or it can specify processing that is to be done to an object, in which case it is referred to as an *instruction* CMR. Finally, there is a special case of an audit and instruction color conversion pair that has been combined to produce a link color conversion. This combined color conversion is called a *link* CMR.

In AFP environments, CMRs are processed as AFP resources by print servers so they can be downloaded once, captured, and used repeatedly without additional downloads. CMRs are also applicable to non-AFP environments such as PostScript, PDF, and PCL.

The primary **purpose** of the Color Management Object Content Architecture is to provide a standard definition for color management resources that are used for controlling presentation of color objects. “Color objects”, as used in the document, means full-color, grayscale and monochrome objects. This standardization provides conventions and directions for current and future products to present objects in a consistent way.

Development of CMOCA has the following goals:

- To allow a means to represent color management information in any environment
- To use a format that is flexible enough to allow it to exist intact in interactive, presentation and interchange environments that are defined in the following data stream architectures:
  - Intelligent Printer Data Stream (IPDS) and
  - Mixed Object Document Content Architecture (MO:DCA)
- To describe the CMR in terms of architected tags
- To use industry-standard constructs when architecting the CMRs
- To allow the CMR to be fully described in device-independent and process-independent terms
- To use a naming convention for the CMRs that allow device-specific color resources to be substituted for generic resources
- To define CMRs so that multiple CMRs can be invoked at one time, and a hierarchy can be searched to determine the appropriate CMRs to use

In AFP environments, CMRs will be carried within an object container. CMRs can be associated with a document component at various levels:

1. Printfile
2. Document
3. A group of pages or sheets
4. Page/Overlay
5. Data Object - for example, IOCA, EPS, TIFF

Within the IPDS data stream, CMRs are activated and deactivated like all other IPDS resources but the CMR is not used until it is explicitly invoked (except for certain Link Color Conversions CMRs, that need not be invoked). Within the device, IPDS hierarchical rules are used to determine which CMRs are actually applied.

Chapter 3. Color Management Resource (CMR)

A CMR consists of a header followed by CMR data.

General Color Conversion Concepts

To start, for simplicity, assume that the source data is specified in a device-independent color space (that is, an ICC Profile Connection Space (PCS) such as CIEXYZ or CIELAB). The procedure for producing output uses the following sequence:

Figure 4. Procedure for Producing Halftone Output from PCS

When color data has been processed using the above sequence, the resulting object can be stored in a database. It might be useful to keep an audit trail of the operations that were performed to create the object. This audit information can be merely descriptive, or it can be used to undo some of the operations performed on the object and thus restore it to the original form when it was expressed in the PCS color space. In this case, the inverse of each function is applied in this sequence:

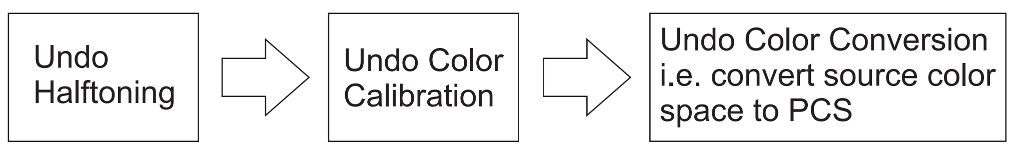


Figure 5. Procedure for Converting Input Data to Full Color in PCS

Here is a more mathematical way of expressing that sequence:

(halftone data)$^{-1}$ ==> (color calibration)$^{-1}$ ==> (color conversion)$^{-1}$

In actuality, halftoning is not typically undone. The halftoned version is typically not stored in the database. Further, undoing the halftoning is very complex. Note also that objects prepared for a display are not halftoned. Therefore, this document assumes that no attempt will be made to go from halftoned data back to the object represented in the PCS color space.

Now, let's assume that the source data is specified in a device-dependent color space and it is desired to render it on an output device. The source data must be converted to the color space of the device. This involves the following combined sequence:

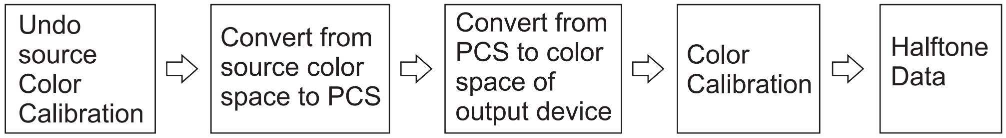


Figure 6. Procedure for Creating Halftone Data from Device-Specific Color Data

Information that relates to the creation of the source data is referred to as an audit CMR. It might describe how the source data was created, as in an audit halftone. Or it might describe how to undo an operation that was used to create the source data. For instance, an audit color conversion profile tells how to convert from the color space of the source data to a PCS. Audit tone transfer curves have been architected to describe an action that was done to create the source data. There is also an optional tag that describes how to *undo* the tone transfer curve.

The above discussion assumes that the color was specified using a color space. It is also possible to specify a color using an index and then define how to produce that color using a palette that tells which component inks/toners are used, in which concentration. An Indexed CMR is used to define how to produce the indexed colors.

Understanding the Use of CMRs

There are seven basic uses of the CMR that must be considered:

1. **Audit Halftone** can be attached to an image that has been halftoned and indicates which halftone was used.
2. **Audit Tone Transfer Curve** indicates a one-dimensional conversion that was applied to the input color before halftoning. The inverse curve needs to be applied to get back to the input color space.
3. **Audit Color Conversion** is used to convert from the input color space to a profile connection space.
4. **Instruction Color Conversion** is used to convert from the profile connection space to the color space of the output device.
5. **Instruction Tone Transfer Curve** is a one-dimensional conversion applied to the output color before halftoning (or a *set* of 1-D conversions, one for each color component).
6. **Instruction Halftone** is used to halftone the output colored data.
7. **Link Color Conversion** provides an efficient means for converting directly from the input color space to the output color space and can be substituted for the Audit Color Conversion/Instruction Color Conversion pair. A special type of Link Color Conversion, ICC DeviceLink, converts directly from input to output space without reference to an audit/instruction pair.

When a presentation device is processing data using the CMR system, these are the basic steps that are followed. Note that device-default CMRs are used if an applicable CMR is not explicitly invoked. If a CMR is *ignored*, the device must accept it but does not error check or process the contents.

1. The device ignores an **Audit Halftone**. It might be useful information to store with the image in a database, but it is not currently used in the presentation device since undoing halftoning is not a simple process.
2. The device applies an applicable **Audit Tone Transfer Curve**. If an inverse tone transfer curve is specified, it is used. Otherwise, the function specified by the

tone transfer curve must be inverted before it is applied. Note that the inverse tone transfer curve is not a well-defined function.

Note: It is strongly recommended that any audit Tone Transfer Curve that is used should be invoked at the object level. This helps avoid ambiguity that might occur since the selection of the TTC is based only on the number of components in the color space.

3. If an applicable **ICC DeviceLink** exists, it is used. This link color conversion converts the input color space to the output color space of the current device. If this CMR is selected for use, the next two steps (selection of Audit Color Conversion and Instruction Color Conversion CMRs) are skipped.
4. If an applicable **Audit Color Conversion** exists, it is used. This audit Color Conversion converts the input color space (RGB, CMYK, grayscale...) to a profile connection space.
5. An **Instruction Color Conversion** is used to convert from the profile connection space into the output color space of the current device.
6. If an **Instruction Tone Transfer Curve** exists, the color is modified using it.
7. The colored data is halftoned using an **Instruction Halftone**.

In reality, the processing of sequential steps can be combined to improve performance. A Link Color Conversion CMR can be used in place of a combined audit/instruction color conversion pair. Chapter 6 discusses how CMRs are used in more detail.

The Indexed CMR does not fit into the scheme discussed above. It is used to define how to produce indexed color.

CMR Syntax

This is the syntax of a CMR. Bytes 0–163 represent the CMR Header. Each field in the CMR header has a fixed length. The fields from byte 10–155 are character data encoded as UTF-16BE. Data in these fields are left aligned. If the number of the characters in these fields is smaller than the field length, the remaining bytes will be padded with @ (X'0040'). Any field from byte 44–139 is “unspecified” if it is filled with @ characters.

The **CMR name** is the concatenated fields in bytes 10–155, exactly in the order specified in the CMR header. The CMR name is 73 characters (146 bytes) long. It should be unique, since it is the name that will be used in the MO:DCA data stream to reference the resource. For example, if the CMR header fields are:

Alias=JohnMay4, CMRType=HT, CMRVersion=1.2, ManufacturerName=IBM, DeviceType=4100, DeviceModel=PD1, MediaBrightness=94, MediaColor=wht, MediaFinish=gl, MediaWeight=90, number of device levels=2, Halftone type=rnd, line screen frequency= 141, resolution= 600, and rotation=proc

then the CMR name is:

```
JohnMay4HT001.200IBM@@4100@@PD194@whtgl90@2@@@@rnd@@@141@600@proc@@@@@@@@
```

Table 5. CMR Object Syntax

Length in Bytes	Offset	Type	Name	Range	Meaning	M/O
4	0–3	4-byte UBIN	Length	X'000000A4' – X'FFFFFFFF'	CMR length, including length field	M
4	4–7	CODE	CMRSig	X'434D5239'	Signature of this CMR	M
2	8–9			X'0000'	Reserved: should be set to zero	M
CMR Name starts here. It is composed of bytes 10–155.						
16	10–25	UTF16	CMRAlias	No restriction	Human-readable alias	M
4	26–29	UTF16	CMRType	CC (X'0043 0043')	Color Conversion	M
				DL (X'0044 004C')	ICC DeviceLink Color Conversion	
				HT (X'0048 0054')	Halftone	
				IX (X'0049 0058')	Indexed	
				LK (X'004C 004B')	Link Color Conversion	
				TC (X'0054 0043')	Tone Transfer Curve	
14	30–43	UTF16	CMRVersion	ddd.ddd (where “d” is a decimal digit character)	CMRVersion number	M
				AFP.ddd (where “d” is a decimal digit character)	Special AFP version number	
				generic	“generic”	
				pasthru	“passthrough”	

Table 5. CMR Object Syntax (continued)

Length in Bytes	Offset	Type	Name	Range	Meaning	M/O
10	44–53	UTF16	Manufacturer Name	See description for name	Name of the manufacturer	M
				@@@@@		
12	54–65	UTF16	DeviceType	See description for type	Type of the device	M
				@@@@@@		
6	66–71	UTF16	DeviceModel	See description for model	Model of the device	M
				@@@		
6	72–77	UTF16	Media Brightness	0–100 for print media	For print media, it is the percentage of light reflected from the media.	M
				Zxy for screens	For screen, it's a CIE illuminant.	
				@@@		

Table 5. CMR Object Syntax (continued)

Length in Bytes	Offset	Type	Name	Range	Meaning	M/O
6	78–83	UTF16	MediaColor	No restriction on the range. A sample of CMOCA-recommended values is given below.	Color of the media:	M
				blu (X'0062 006C 0075')	blue	
				buf (X'0062 0075 0066')	buff	
				gdr (X'0067 0064 0072')	goldenrod	
				grn (X'0067 0072 006E')	green	
				gry (X'0067 0072 0079')	gray	
				ivy (X'0069 0076 0079')	ivory	
				noc (X'006E 006F 0063')	no-color	
				org (X'006F 0072 0067')	orange	
				pnk (X'0070 006E 006B')	pink	
				red (X'0072 0065 0064')	red	
				wht (X'0077 0068 0074')	white	
				ylw (X'0079 006C 0077')	yellow	
				A three-character value consisting of only upper-case characters in the range [A,Z]	custom	
				@@@ (X'0040 0040 0040')	not specified	

Table 5. CMR Object Syntax (continued)

Length in Bytes	Offset	Type	Name	Range	Meaning	M/O
4	84–87	UTF16	MediaFinish	No restriction on the range. A sample of CMOCA-recommended values is given below.	Surface characteristics of the media:	M
				cm (X'0063 006D')	commodity	
				ct (X'0063 0074')	coated	
				gl (X'0067 006C')	glossy	
				hg (X'0068 0067')	high-gloss	
				mt (X'006D 0074')	matte	
				no (X'006E 006F')	none	
				np (X'006E 0070')	newsprint	
				sg (X'0073 0067')	semi-gloss	
				st (X'0073 0074')	satin	
				tr (X'0074 0072')	treated	
				A two-character value consisting of only upper-case characters in the range [A,Z]	custom	
				@@ (X'0040 0040')	not specified	
6	88–93	UTF16	MediaWeight	1–999 @@@	The basic weight of the paper	M
10	94–103	UTF16	Prop1	See description No restriction	CMRType property-specific field 1	M
12	104–115	UTF16	Prop2	See description No restriction	CMRType property-specific field 2	M
8	116–123	UTF16	Prop3	See description No restriction	CMRType property-specific field 3	M
8	124–131	UTF16	Prop4	See description No restriction	CMRType property-specific field 4	M
8	132–139	UTF16	Prop5	See description No restriction	CMRType property-specific field 5	M
16	140–155	UTF16		@@@@@@@@	Reserved: should be set to @@@@@@@@	M
CMR Name ends here. It is composed of bytes 10–155.						
8	156–163			X'0000000000000000'	Reserved: should be set to zero	M
	164–end		CMRData	Any	Resource data	O

Notes:
1. M/O denotes a mandatory or optional field
2. UTF16 denotes UTF-16BE

CMR Header Semantics

Length
The length of the complete CMR, including the Length parameter. Length may be 164 (X'000000A4') bytes up to X'FFFFFFFF'.

CMRSig
The signature of the CMR that allows it to be easily recognized. It will be three ASCII characters "CMR" followed by X'39', that is, X'434D5239'.

CMRAlias
Eight-character user-defined string to enable an easy way of identifying the CMR.

CMRType
Five CMRTypes are defined in this Color Management Object Content Architecture. They are: Halftone, Tone Transfer Curve, Color Conversion, Link Color Conversion, and Indexed. The value of the CMRType must be specified in the header or an exception will be generated. Note that a Link Color Conversion CMR has two possible identifiers in this field: LK for LinkColorConversion LUT subset and DL for ICC DeviceLink subset.

CMRVersion
The CMRVersion number consists of a major version number (an integer of 1–3 digits) and a minor version number (an integer of 1–3 digits), separated by a decimal point (. =DECIMAL POINT=X'002E'). If the number of digits is smaller than 3, zeroes will be padded to the left side of the major number or to the right side of the minor number. For example, if the version number is 1.20 then the value of the CMRVersion is 001.200.

A value of "AFP.ddd" is used to specify a minor version number along with "AFP" to indicate that the CMR is a standard Color Conversion CMR that is supported by the AFP Consortium. The supported standard color spaces will be spaces like SWOP CMYK and sRGB.

A value of "generic" (X'0067 0065 006E 0065 0072 0069 0063') in this field identifies a generic CMR. Only Halftone and Tone Transfer Curve CMRs can be identified as generic. CMR data in generic CMRs is optional and is not used in AFP color management systems.

A value of "pasthru" (X'0070 0061 0073 0074 0068 0072 0075') identifies a color space that should not be color-converted. Only Color Conversion CMRs can be identified as passthrough. There is no data in a passthrough CMR.

The value of the CMRVersion must be specified in the header. The CMRVersion tracks changes besides the changes in the device-specific fields, media specific fields, and the CMRType property fields. It reflects changes of algorithm, toner, and so on.

Device-Specific Fields

For IPDS receivers, the ManufacturerName, DeviceType, and DeviceModel values must be provided in accordance with the IPDS description of the Product Identifier self-defining field of the XOH Obtain Printer Characteristics (OPC) reply. Refer to the *Intelligent Printer Data Stream Reference*. The field descriptions are as follows:

ManufacturerName
: Name of the manufacturer.

DeviceType
: Device type of the printer that corresponds to the device type imprinted on the serial number plate that is physically attached to the printer.

DeviceModel
: Model number of the printer that corresponds to the model number imprinted on the serial number plate that is physically attached to the printer.

For the non-IPDS devices, maximum five characters are allowed for the ManufacturerName. The stock symbol (maximum five characters), a unique name assigned by stock exchanges worldwide, is recommended to be used for the ManufacturerName. The DeviceType and DeviceModel have to be unique and meaningful for the devices. Alternatively, the ICC Manufacture ASCII Signature and Device ASCII Signature can be used for the ManufacturerName and the DeviceModel.

Implementation notes:

1. If the DeviceType is unspecified (@@@@@@), then it automatically matches the DeviceType of the target device. Similarly, if the DeviceModel is unspecified (@@@), then it automatically matches the DeviceModel of the target device. The DeviceType and DeviceModel are sometimes used by print servers to determine which CMRs to send to the presentation device. In particular, Link CMRs are targeted for a particular device based on the DeviceType and DeviceModel of the instruction Color Conversion CMR. Multiple Link CMRs can be associated with (or mapped to) an audit CC CMR in the CMR RAT. The Link CMRs that are sent down to the device are determined by finding matches with the DeviceType and DeviceModel of the target device. Furthermore, Generic Tone Transfer Curve and Halftone CMRs can have mapped device-specific CMRs in the CMR RAT; such mapped CMRs are sent to the device if the DeviceType and DeviceModel in the mapped CMR match the DeviceType and DeviceModel of the target device. In some situations, it is acceptable to let the CMR header values for DeviceType and DeviceModel be unspecified (@@@@@@ or @@@). For example, CMRs that will be used only as audit CMRs can have unspecified values for DeviceType and DeviceModel. If a Link CMR or a device-specific HT or TTC CMR is associated with another CMR in the CMR RAT and does not specify a DeviceType and/or DeviceModel, the unspecified parameter(s) match the DeviceType and/or DeviceModel of any target printer.
2. The device types and model numbers specified in the XOH-OPC reply and in the CMR header's DeviceType might not use the same format. For instance, for the InfoPrint 4100, the XOH-OPC reply for the device type would be "004100" encoded using EBCDIC. In the CMR header, the DeviceType is padded with "@" on the right. Therefore, depending on the input provided to the Installer, the CMR DeviceType field might be "004100" or "4100@@" encoded using UTF-16BE. Tools that compare the device type in the XOH-OPC reply and in the CMR header must be prepared to indicate a match taking into account the differences in padding practices.

Media-Specific Fields

Media-specific fields describe the media and consist of four attributes: media brightness, media color, media finish, and media weight. The values for the MediaColor and the MediaFinish are consistent with the values defined by the Internet Printing Protocol (IPP) of the Printer Working Group (PWG). If the target device is a display, only media brightness is specified.

To use an instruction CMR, its media type must match the media currently being used by the device. Similarly, in order to use an ICC DeviceLink CMR, its media attributes must match the device's media attributes. See "Matching Media Type of CMR with Media Type of Device" on page 117 for a discussion of this requirement.

MediaBrightness

For print media, it indicates the percentage of light reflected from the media. The brightness is measured with a brightmeter machine. The scale is based on the TAPPI GE scale in the US and the ISO scale in the rest of the world. The ISO scale is usually about two units higher than the GE value. For example, 100 ISO brightness is equivalent to 98 brightness on the GE scale. In order to ensure that the CMR's media type matches the media currently being used in the device, the scale that is used to specify each value must be the same.

For screens, the brightness is defined as the CIE standard illuminant as Zxy, where Z is a capitalized letter, and xy is a two-digit number. (ISO/CIE 10526:1999: CIE standard illuminants for colorimetry). For example, D50, D65, etc.

MediaColor

Indicates the color of the media being specified. CMOCA-recommended values exist to encourage interoperability; a CMOCA-recommended value should be used if appropriate for a CMR associated with a specific media. The value "noc" means transparency. Custom values can be defined by the administrator.

There is no restriction on what value can be entered for this field as it is not checked for validity.

MediaFinish

Indicates the surface characteristics of the media. CMOCA-recommended values exist to encourage interoperability; a CMOCA-recommended value should be used if appropriate for a CMR associated with a specific media. The value "no" means no coating. Custom values can be defined by the administrator.

There is no restriction on what value can be entered for this field as it is not checked for validity.

MediaWeight

Indicates the weight of the media rounded to the nearest whole number of grams per square meter.

CMRType Property-Specific Fields

CMRType=Halftone

Note: These fields are informational only. They are not checked for validity. Any value can be entered in the Prop fields since no error checking is done.

Prop1 Number of Device Levels

Defines the number of device levels. The number should be greater than 1. For example, if the number of device levels is equal to 3, then the device levels are 0, 1, and 2. If the number of device levels is different for different components, this property represents the maximum value.

Prop2 Halftone Type

Defines the halftone type. Halftone types are divided into four major categories: clustered-dot, stochastic, dispersed, and error diffusion. The dot shape is used to specify the type of the clustered-dot, and the error diffusion filter name is used to specify the type of error diffusion halftone.

Table 6. Halftone Types

Range	Meaning
rnd@@@ (X'0072 006E 0064 0040 0040 0040')	Round dot for the clustered-dot halftone
sqr@@@ (X'0073 0071 0072 0040 0040 0040')	Square dot for the clustered-dot halftone
dia@@@ (X'0064 0069 0061 0040 0040 0040')	Diamond dot for the clustered-dot halftone
rhm@@@ (X'0072 0068 006D 0040 0040 0040')	Rhombus dot for the clustered-dot halftone
elp@@@ (X'0065 006C 0070 0040 0040 0040')	Elliptical dot for the clustered-dot halftone
eud@@@ (X'0065 0075 0064 0040 0040 0040')	Euclidean dot for the clustered-dot halftone
lin@@@ (X'006C 0069 006E 0040 0040 0040')	Line shape dot for the clustered-dot halftone
sto@@@ (X'0073 0074 006F 0040 0040 0040')	Stochastic halftone
dsp@@@ (X'0064 0073 0070 0040 0040 0040')	Dispersed halftone
erd@@@ (X'0065 0072 0064 0040 0040 0040')	Unspecified error diffusion halftone
f-d@@@ (X'0066 002D 0064 0040 0040 0040')	Floyd-Steinberg error diffusion halftone
jjn@@@ (X'006A 006A 006E 0040 0040 0040')	Jarvis-Judice-Ninke error diffusion halftone
stu@@@ (X'0073 0074 0075 0040 0040 0040')	Stucki error diffusion halftone
brk@@@ (X'0062 0072 006B 0040 0040 0040')	Burkes error diffusion halftone
sra@@@ (X'0073 0072 0061 0040 0040 0040')	Sierra error diffusion halftone
s-a@@@ (X'0073 002D 0061 0040 0040 0040')	Stevenson Arce error diffusion halftone

Prop3 Line Screen Frequency

Defines the maximum line screen frequency of all the component screens. Line frequency is specified in terms of the printer's resolution. A line screen frequency of zero should be used for stochastic, dispersed and error diffusion type halftones.

Prop4 Resolution

Defines the printer resolution in dots per inch in the array width (screen width) direction. Halftone screens are normally designed for different printer resolutions. If the resolutions differ for different components, this property represents the maximum value.

Prop5 Rotation

Defines the orientation of the halftone. There are three possible values: orientation independent, along the scan direction, and along the process direction.

Table 7. Halftone Rotations

Range	Meaning
indp (X'0069 006E 0064 0070')	Orientation independent
scan (X'0073 0063 0061 006E')	Scan direction
proc (X'0070 0072 006F 0063')	Process direction

CMRtype=Tone Transfer Curve

Note: These fields are informational only. They are not checked for validity. Any value can be entered in the Prop fields since no error checking is done.

Prop1 Profile/Device Class Signature

The definition of the Device Class Signature is consistent with the definition in the ICC header. There are four basic profile/device classes: Input, Display, Output, and ColorSpace Conversion.

Table 8. ICC Profile/Device Classes for Tone Transfer Curve CMRs

Range	Meaning
scnr@ (X'0073 0063 006E 0072 0040')	Input Device
mntr@ (X'006D 006E 0074 0072 0040')	Display Device
prtr@ (X'0070 0072 0074 0072 0040')	Output Device
spac@ (X'0073 0070 0061 0063 0040')	ColorSpace Conversion

Prop2 Look-and-Feel

Look-and-Feel produced in the output when this Tone Transfer Curve is applied. See Appendix B, “Generic CMR Name Registry,” on page 127 for an explanation of what these values mean.

Table 9. Look-and-Feel Values

Range	Meaning
hilmid (X'0068 0069 006C 006D 0069 0064')	Highlight Midtone
standd (X'0073 0074 0061 006E 0064 0064')	Standard
dark@@ (X'0064 0061 0072 006B 0040 0040')	Dark
accutn (X'0061 0063 0063 0075 0074 006E')	Accutone

Prop3 Halftone Characterization

This Tone Transfer Curve was designed to work with a particular Halftone. This value is used to identify that Halftone. For clustered-dot halftones, it is the line screen frequency (Prop3 of Halftone). For other types of halftones, it is the halftone type (Halftone Prop2 but just the first four characters).

Prop4 Reserved for future use.

Prop5 Reserved for future use.

CMRType=Color Conversion

Note: These fields are informational only. They are not checked for validity. Any value can be entered in the Prop fields since no error checking is done.

Prop1 Profile/Device Class Signature

It is consistent with the definition of the Profile/Device Class Signature in the ICC header.

Table 10. ICC Profile/Device Classes for Color Conversion CMRs

Range	Meaning
scnr@ (X'0073 0063 006E 0072 0040')	Input Device profile
mntr@ (X'006D 006E 0074 0072 0040')	Display Device profile
prtr@ (X'0070 0072 0074 0072 0040')	Output Device profile
spac@ (X'0073 0070 0061 0063 0040')	ColorSpace Conversion profile

Prop2 Reserved for future use.

Prop3 Reserved for future use.

Prop4 Color Space of Data

It is consistent with the definition of the Color Space of Data in the ICC header. Table 11 shows the possible values.

Table 11. The ICC Color Space of Data

Range	Meaning
XYZ@ (X'0058 0059 005A 0040')	XYZData
Lab@ (X'004C 0061 0062 0040')	labData
Luv@ (X'004C 0075 0076 0040')	luvData
YCbr (X'0059 0043 0062 0072')	YCbCrData
Yxy@ (X'0059 0078 0079 0040')	YxyData
RGB@ (X'0052 0047 0042 0040')	rgbData
GRAY (X'0047 0052 0041 0059')	grayData
HSV@ (X'0048 0053 0056 0040')	hsvData
HLS@ (X'0048 004C 0053 0040')	hlsData
CMYK (X'0043 004D 0059 004B')	cmykData
CMY@ (X'0043 004D 0059 0040')	cmyData
2CLR (X'0032 0043 004C 0052')	2colorData
3CLR (X'0033 0043 004C 0052')	3colorData (if not listed above)
4CLR (X'0034 0043 004C 0052')	4colorData (if not listed above)
5CLR (X'0035 0043 004C 0052')	5colorData
6CLR (X'0036 0043 004C 0052')	6colorData
7CLR (X'0037 0043 004C 0052')	7colorData
8CLR (X'0038 0043 004C 0052')	8colorData
9CLR (X'0039 0043 004C 0052')	9colorData
ACLR (X'0041 0043 004C 0052')	10colorData
BCLR (X'0042 0043 004C 0052')	11colorData

Table 11. The ICC Color Space of Data (continued)

Range	Meaning
CCLR (X'0043 0043 004C 0052')	12colorData
DCLR (X'0044 0043 004C 0052')	13colorData
ECLR (X'0045 0043 004C 0052')	14colorData
FCLR (X'0046 0043 004C 0052')	15colorData

Prop5 PCS

The profile connection space specified as either CIEXYZ (XYZ) or CIELAB (Lab), encoded as for Prop4.

CMRType=Link Color Conversion

Note: These fields are informational only. They are not checked for validity. Any value can be entered in the Prop fields since no error checking is done.

Prop1 Input Device ManufacturerName

The ManufacturerName for the input device. Note that the ManufacturerName for the output device is defined in the device-specific fields of this CMR name.

Prop2 Input DeviceType

The DeviceType for the input device. Note that the DeviceType for the output device is defined in the device-specific fields of this CMR name.

Prop3 Input DeviceModel

The DeviceModel for the input device. Note that the DeviceModel for the output device is defined in the device-specific fields of this CMR name.

Prop4 Input Color Space

The same as Color Space of Data defined in the ICC header, encoded as in Table 11 on page 21.

Prop5 Output Color Space

Device-specific color space, a subset of the Color Space of Data defined in the ICC profile header. Possible values are shown in the following table.

Table 12. Output Color Spaces

Range	Meaning
RGB@ (X'0052 0047 0042 0040')	rgbData
GRAY (X'0047 0052 0041 0059')	grayData
CMYK (X'0043 004D 0059 004B')	cmykData
CMY@ (X'0043 004D 0059 0040')	cmyData
2CLR (X'0032 0043 004C 0052')	2colorData
3CLR (X'0033 0043 004C 0052')	3colorData (if not listed above)
4CLR (X'0034 0043 004C 0052')	4colorData (if not listed above)
5CLR (X'0035 0043 004C 0052')	5colorData
6CLR (X'0036 0043 004C 0052')	6colorData
7CLR (X'0037 0043 004C 0052')	7colorData
8CLR (X'0038 0043 004C 0052')	8colorData

Table 12. Output Color Spaces (continued)

Range	Meaning
9CLR (X'0039 0043 004C 0052')	9colorData
ACLR (X'0041 0043 004C 0052')	10colorData
BCLR (X'0042 0043 004C 0052')	11colorData
CCLR (X'0043 0043 004C 0052')	12colorData
DCLR (X'0044 0043 004C 0052')	13colorData
ECLR (X'0045 0043 004C 0052')	14colorData
FCLR (X'0046 0043 004C 0052')	15colorData

CMRType=Indexed

Note: These fields are informational only. They are not checked for validity. Any value can be entered in the Prop fields since no error checking is done.

Prop1 Reserved for future use.

Prop2 Reserved for future use.

Prop3 Reserved for future use.

Prop4 Reserved for future use.

Prop5 Reserved for future use.

CMR Data

Data CMR data.

The content of the data is defined by the CMR type. The CMR data carries the color resource data. The resource data is carried in a tagged format. The tags are loosely based on the TIFF tag syntax, but with significant changes and additions. The tag syntax is defined in Chapter 5, “CMR Data Architecture,” on page 37.

| CMR data is optional for generic and passthrough CMRs. If CMR data is specified for a generic or passthrough CMR, it is ignored.

CMR Header Exception Conditions

On encountering an error, an exception is raised. Exception conditions have a format of EC-xxxxyy. xxxx represents the tag value. For the purposes of error reporting, the fields in the CMR header are treated as “implied tags”. The architecture defines the tags that describe data fields to have tag IDs of X'0000'–X'FFFF'. However, IDs in the range X'EF00'–X'EFFF' have been reserved for error handling in the CMR Header. Currently, IDs in the range X'EFF0'–X'EFF7' are used for CMR header error codes.

The error codes (yy in EC-xxxxyy) are used as follows:

- X'03' - invalid length
- X'10' - invalid or unsupported field value

The exception conditions are as follows:

EC-EFF003 Invalid Length Value

The specified Length is invalid.

EC-EFF110 Invalid Field Value

The specified value for CMRSig is not X'434D5239'.

EC-EFF210 Invalid Field Value

The specified CMRType is invalid.

EC-EFF310 Invalid Field Value

The specified CMRVersion is invalid.

EC-EFF410 Invalid Field Value

The specified MediaBrightness is invalid.

EC-EFF510 Retired item 1.

EC-EFF610 Retired item 2.

EC-EFF710 Invalid Field Value

The specified MediaWeight is invalid.

CMRData Exceptions in the data are defined by the actual tags.

Prop 1–5 are informational. The values are not checked.

Chapter 4. CMR Types

The following CMRTypes are defined:
- Halftone
- Tone Transfer Curve
- Color Conversion
- Link Color Conversion
- Indexed

Each of the CMRTypes is described in more detail below.

In the following descriptions, the Optional Tags and Mandatory Tags are listed to show which tags are meaningful for each type and subset. Any other tags present are ignored by the receiver. The Comment Tag is optional. The End Data Tag is required in all CMR objects.

Halftone CMR

CMR Type HT (X'0048 0054')

Description

Halftone can be classified into two types: point-operation halftone and neighborhood-operation halftone. The output of the point-operation halftone depends only on the value of the current pixel. It can be used for numerous common halftone types including clustered dot, dispersed and stochastic. Threshold arrays are commonly used for the bilevel point-operation halftones, and lookup tables are commonly used for the multilevel point-operation halftones. The error diffusion halftone is commonly used for the neighborhood-operation halftone. The most common error diffusion filters include Floyd-Steinberg, Jarvis-Judice-Ninke, Stucki, etc. The Halftone Subset tag indicates the halftone type, and thus determines required and optional tags for this Halftone CMR.

Subset X'01': Bilevel Point-Operation Halftone

Mandatory Tags
Halftone Subset, Array Width, Array Height, Bilevel Point-Operation Screen Data, End Data

Optional Tags
Comment, Date and Time Stamp, Number of Components, Offset Tiling

Subset X'02': Multilevel Point-Operation Halftone

Mandatory Tags
Halftone Subset, Array Width, Array Height, Max Image Value, Number of Device Levels, Multilevel Point-Operation Screen Data, End Data

Optional Tags
Comment, Date and Time Stamp, Number of Components, Offset Tiling

Subset X'03': Bilevel Error Diffusion Halftone

Mandatory Tags
: Halftone Subset, Array Width, Array Height, Error Diffusion Filter, Location of Current Pixel, Raster Direction, Boundary Condition, Threshold Value, End Data

Optional Tags
: Comment, Date and Time Stamp, Number of Components

Subset X'04': Multilevel Error Diffusion Halftone

Mandatory Tags
: Halftone Subset, Array Width, Array Height, Number of Device Levels, Error Diffusion Filter, Location of Current Pixel, Raster Direction, Boundary Condition, Quantization Boundary Table, End Data

Optional Tags
: Comment, Date and Time Stamp, Number of Components

Tone Transfer Curve CMR

CMR Type TC (X'0054 0043')

Description

Tone transfer curves are applied to data prior to halftoning or output. The inverse tone transfer curves are applied to restore data to the original state. The printer tone transfer curve produces a desired appearance by compensating for dot gain. The tone transfer curve for a display or other input device is used to correct non-linearity (gamma) of the device. Currently, there are two tone transfer curve subsets: ToneTransferCurve Array, and ToneTransferCurve Identity. The Subset tag indicates the tone transfer curve type, and thus determines required and optional tags for this Tone Transfer Curve CMR.

Subset X'01': ToneTransferCurve Array

Mandatory Tags
Tone Transfer Curve Subset, Tone Transfer Curve Data, End Data

Optional Tags
Comment, Date and Time Stamp, Number of Components, Tone Transfer Curve Length, Inverse Tone Transfer Curve Data

Subset X'02': ToneTransferCurve Identity

The tone transfer curve for each component is the identity. No data is sent with this subset, that is, no tone transfer curve is to be applied. This subset is implemented for performance reasons.

Mandatory Tags
Tone Transfer Curve Subset, End Data

Optional Tags
Comment, Date and Time Stamp

Color Conversion CMR

CMR Type CC (X'0043 0043')

Description

Each instance of this CMR type is a subset of the standard ICC profile. This allows the CMR to be used in any color management system.

The ICC Profile Data starts with a 128-byte header followed by the ICCTags. The ICCHeaderFields are contained in pre-defined byte positions as defined in Table 36 on page 70.

Each subset of the ICC profile type, selected by the ICC Profile Subset tag, defines a subset of the ICC specification. For each subset, the Color Management Object Content Architecture defines the mandatory and optional ICCHeaderFields and ICCTags. Optional ICCTags and ICCHeaderFields will be processed as applicable if encountered. Any other ICCTags will be ignored.

Note: The chromaticAdaptationTag is shown as optional for each subset. However, it is mandatory if the value in the mediaWhitePointTag is not D50.

Two ICCHeaderFields are mandatory for the Color Conversion CMR: Color Space of Data and Profile Connection Space. The descriptions are as follows:

Color Space of Data
The ICC supported color spaces and their signatures are listed in Table 11 on page 21.

Profile Connection Space
The ICC profile connection space is either CIELAB D50 or CIEXYZ D50 for all ICC profiles except the ICC DeviceLink profile. The CIELAB signature is “Lab” and the CIEXYZ signature is “XYZ”.

The currently allowed ICC profile subsets for Color Conversion CMR include all the ICC profile types except for the DeviceLink and the Named Colour profiles. (For complete information, please refer section 8.6 and 8.9 in ICC 1:2004-10 Version 4.2.0.0.)

The ICC profile subsets for the CMR Color Conversion are listed in Table 13.

Table 13. ICC Profile Subsets for the CMR Color Conversion

Type	Usage	Basic Intents
Monochrome input profile	Scanner, digital camera	Device --> PCS
Monochrome display profile	Display	Device <----> PCS
Monochrome Output profile	Printer	Device --> PCS
Three-component matrix-based input profile	Scanner, digital camera	Device --> PCS
Three-component matrix-based display profile	Display	Device <----> PCS
N-component LUT-based input profile	Scanner, digital camera	Device -->PCS
N-component LUT-based display profile	Display	Device <----> PCS

Table 13. ICC Profile Subsets for the CMR Color Conversion (continued)

Type	Usage	Basic Intents
N-Component LUT-based output profiles	Printer, film recorder	PCS --> Device
ColorSpace conversion profile	Non-device color space, for example, sRGB, D65	Non-device <----> PCS

The Basic Intents Column describes the most commonly used color conversion direction (s) for each ICC profile subset.

Mandatory Tags
ICC Profile Subset, ICC Profile Data, End Data

Optional Tags
Comment, Date and Time Stamp, ICC Profile Filename

Implementation note: It is very important to include the ICC Profile Filename tag for debugging purposes regardless of the fact that it is optional.

Subset X'01': Monochrome Input Profile

Mandatory ICCHeaderFields

Table 14. Mandatory ICCHeaderFields for Subset X'01': Monochrome Input Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, grayTRCTag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB0Tag, AToB1Tag, AToB2Tag, BToA0Tag, BToA1Tag, BToA2Tag

Subset X'02': Monochrome Display Profile

Mandatory ICCHeaderFields

Table 15. Mandatory ICCHeaderFields for Subset X'02': Monochrome Display Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, grayTRCTag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB0Tag, AToB1Tag, AToB2Tag, BToA0Tag, BToA1Tag, BToA2Tag

Subset X'03': Monochrome Output Profile

Mandatory ICCHeaderFields

Table 16. Mandatory ICCHeaderFields for Subset X'03': Monochrome Output Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, grayTRCTag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB0Tag, AToB1Tag, AToB2Tag, BToA0Tag, BToA1Tag, BToA2Tag

Subset X'04': Three-Component Matrix-Based Input Profile

Mandatory ICCHeaderFields

Table 17. Mandatory ICCHeaderFields for Subset X'04': Three-Component Matrix-Based Input Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
| profileDescriptionTag, redMatrixColumnTag, greenMatrixColumnTag,
| blueMatrixColumnTag, redTRCTag, greenTRCTag, blueTRCTag,
mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB0Tag, AToB1Tag, AToB2Tag, BToA0Tag, BToA1Tag, BToA2Tag, gamutTag

Subset X'05': Three-Component Matrix-Based Display Profile

Mandatory ICCHeaderFields

Table 18. Mandatory ICCHeaderFields for Subset X'05': Three-Component Matrix-Based Display Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, redMatrixColumnTag, greenMatrixColumnTag, blueMatrixColumnTag, redTRCTag, greenTRCTag, blueTRCTag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB0Tag, AToB1Tag, AToB2Tag, BToA0Tag, BToA1Tag, BToA2Tag, gamutTag

Subset X'06': N-Component LUT-Based Input Profile

Mandatory ICCHeaderFields

Table 19. Mandatory ICCHeaderFields for Subset X'06': N-Component LUT-Based Input Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, AToB0Tag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB1Tag, AToB2Tag, BToA0Tag, BToA1Tag, BToA2Tag, gamutTag

Subset X'07': N-Component LUT-Based Display Profile

Mandatory ICCHeaderFields

Table 20. Mandatory ICCHeaderFields for Subset X'07': N-Component LUT-Based Display Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, AToB0Tag, BToA0Tag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB1Tag, AToB2Tag, BToA1Tag, BToA2Tag, gamutTag

Subset X'08': N-Component LUT-Based Output Profiles

Mandatory ICCHeaderFields

Table 21. Mandatory ICCHeaderFields for Subset X'08': N-Component LUT-Based Output Profiles

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, AToB0Tag, BToA0Tag, AToB1Tag, BToA1Tag, AToB2Tag, BToA2Tag, gamutTag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, colorantTableTag

Subset X'09': ColorSpace Conversion Profile

This subset should be used as an audit color conversion.

Mandatory ICCHeaderFields

Table 22. Mandatory ICCHeaderFields for Subset X'09': ColorSpace Conversion Profile

Byte Offset	Content
16–19	Color Space of Data
20–23	Profile Connection Space

Optional ICCHeaderFields
All other header fields

Mandatory ICCTags
profileDescriptionTag, AToB0Tag, BToA0Tag, mediaWhitePointTag, copyrightTag

Optional ICCTags
chromaticAdaptationTag, AToB1Tag, AToB2Tag, BToA1Tag, BToA2Tag, gamutTag

Subset X'0A': Retired item 3 (Abstract profile)

Link Color Conversion CMR

CMR Type LK (X'004C 004B') or DL (X'0044 004C')

Description

The purpose of the Link Color Conversion CMR is to convert directly from the input to the output color space. There are two major types of Link Color Conversion:

- Link Color Conversion with up to four Lookup Tables (LUTs) representing different rendering intents. It is selected for use based on the audit and instruction OIDs that are contained in its tags.
- ICC DeviceLink contains an ICC profile of type DeviceLink. It contains a complex color conversion description (with up to five processing elements) for exactly one rendering intent. It is selected for use when it is found during a search of the hierarchy of invoked Link CMRs (only ICC DeviceLink CMRs are invoked, other Link subsets are not invoked). It is not based on an audit and instruction Color Conversion pair.

LinkColorConversion LUT subset: Its purpose is to combine an audit Color Conversion CMR with an instruction Color Conversion CMR to improve performance. It allows up to four LUTs, each representing one of the four ICC rendering intents. It is permissible to reference the same LUT tag data for multiple rendering intents. The LUT is constructed by combining the processing required for the audit and the instruction color conversions. If one of the AToB/BToA tags in an audit or instruction CMR is missing when constructing the link LUT, the tag data for the rendering intent specified in that CMR's ICC profile header is used in place of the missing tag data. The default rendering intent profile header is used in place of the missing tag data. The default rendering intent for the Link Color Conversion CMR is the rendering intent specified in the ICCHeader Field of the instruction Color Conversion CMR. The processing detail is described in "Creating Link Color Conversion CMRs – LinkColorConversion LUT subset" on page 109.

ICC DeviceLink subset: Its purpose is to provide a customized path to convert directly from input to output space with no dependency on an audit and instruction CMR. It allows only one single AToB0Tag representing one rendering intent. The rendering intent is indicated in the header of the ICC profile. The AToB0Tag contains up to five processing elements, possibly making the conversion more complex than if a single LUT were used. Whereas Link CMRs in general are not invoked, CMRs with this subset ID must be invoked in order to be used. A hierarchical search is performed to determine if there is an applicable ICC DeviceLink that can be used. The device should search for an ICC DeviceLink before searching for an Audit/Instruction Color Conversion pair. The currently active Rendering Intent is ignored when an ICC DeviceLink is selected for use.

Currently, three Link Color Conversion subsets are defined. Table 23 on page 34 lists the Link Color Conversion CMR subsets and the descriptions:

Table 23. Link Color Conversion Subsets

Subset ID	Type	Usage
X'01'	LinkColorConversion LUT	Connection between two device color spaces, or the connection between a non-device color space and a device color space, for example, Scanner-->printer sRGB-->printer This CMR is selected for use based on the OIDs of the selected audit and instruction CC CMRs.
X'02'	LinkColorConversion Identity	No conversion needed. This CMR is selected for use based on the OIDs of the selected audit and instruction CC CMRs.
X'03'	ICC DeviceLink	Direct conversion between an input color space and an output color space without reference to an audit and instruction CC pair and without use of a PCS.

Subset X'01': LinkColorConversion LUT

Mandatory Tags

Link Color Conversion Subset, Link Audit CMR OID, Link Instruction CMR OID, Link Audit CMR Name, Link Instruction CMR Name, Default Rendering Intent, Link LUT Perceptual, End Data

Note: LUTs for all four rendering intents must be provided, but duplicate LUTs can be identified by setting the appropriate additional use flag in a specified Link LUT tag. For example, when all four LUTs are identical, only the Link LUT Perceptual tag is specified. When an additional use flag for a specific rendering intent is set to B'1', the Link-LUT tag for that rendering intent is omitted.

Optional Tags

Comment, Date and Time Stamp, Link LUT Media-Relative Colorimetric, Link LUT Saturation, Link LUT ICC-Absolute Colorimetric, Link CMRE Identifier

Subset X'02': LinkColorConversion Identity

This subset is used when the input space is the same as the device's output space and no color conversion is to be done. There is no data with this subset. The OIDs for the audit and instruction Color Conversion CMR must be the same.

Mandatory Tags

Link Color Conversion Subset, Link Audit CMR OID, Link Instruction CMR OID, Link Audit CMR Name, Link Instruction CMR Name, End Data

Optional Tags

Comment, Date and Time Stamp, Link CMRE Identifier

Subset X'03': ICC DeviceLink

Mandatory Tags
Link Color Conversion Subset, ICC Profile Data, End Data

Optional Tags
Comment, Date and Time Stamp, ICC Profile Filename

Mandatory ICCTags
Profile Description Tag
Copyright Tag
ProfileSequenceDescTag
AtoB0Tag
colorantTableTag (required only if the Data Colour Space Field is xCLR)
colorantTableOutTag (required only if the Profile Connection Space Field is xCLR)

Optional ICCTags
None

Indexed CMR

CMR Type IX (X'0049 0058')

Description

An Indexed CMR contains one or more Color Palette tags that translate 2-byte indexed color values to the target color space. Five Color Palette tags are defined for the color spaces of gray, RGB, CMYK, CIELAB, and named colorants. The named colorants are defined through a set of colorant names that are specified in the Colorant Identification List tag. Currently, only one Indexed CMR subset is defined for the multi-output color spaces. It allows the mixture of different output color spaces in an Indexed CMR. When multiple Color Palette tags are present in a CMR, and the same indexed color value is specified in different Color Palette tags, the indexed color value in the Color Palette tag with the lower tag ID number is used. If the color space of that Color Palette tag is not applicable for the output device, the CIELAB value specified for this indexed color value in the Color Palette tag is used for the substitution.

Table 24. List of Color Palette Tags

Name	Meaning
Color Palette Gray Color	Color Palette tag for monochrome output devices
Color Palette CMYK	Color Palette tag for CMYK output devices
Color Palette RGB	Color Palette tag for RGB output devices
Color Palette LAB	Color Palette tag for D50 CIELAB color space
Color Palette Named Colorants	Color Palette tag for named colorants color space

Subset X'01': Multi-Output Color Spaces

Mandatory Tags
Indexed Subset, one of Color Palette tags, End Data

Note: Number of Named Colorants tag and Colorant Identification List Tag are mandatory if Color Palette Named Colorants tag is specified.

Optional Tags
Comment, Date and Time Stamp, Color Palette tags

Exception Condition:
If no Color Palette tag is specified, exception condition EC-50400E exists. It is shown in "Color Palette Named Colorants" on page 99

EC-50400E Missing Required Tag

At least one Color Palette Tag is required but none were specified.

Chapter 5. CMR Data Architecture

The CMR Data field carries all the actual color resource data. The resource data is carried in a tagged format. CMR is big endian. The tags are loosely based on the TIFF tag syntax, but with significant changes and additions. The tags are carried first, optionally followed by the tag data. The last tag is always the End Data tag.

Tag Syntax

Each tag consists of 12 bytes in the following format:

Table 25. CMR Data Tag Syntax

Bytes	Type	Name	Range	Meaning
0–1	CODE	TagID	X'0000'–X'FFFF'	Unique identifier for the tag
2		Reserved	X'00'	Should be set to zero
3	CODE	Field Type	X'01'	1-byte UBIN
			X'02'	2-byte UBIN
			X'04'	4-byte UBIN
			X'05'	BYTE (8 bits)
			X'06'	ASCII
			X'07'	UTF16 (UTF-16BE)
			X'08'	CODE (8 bit architected constant)
			X'09'	BITS
4–7	UBIN	Count	X'00000000' – X'FFFFFFFF'	Number of values of the indicated Field Type (may be zero)
8–11		ValueOffset	Any	Data, left-aligned, if it fits into 4 bytes. Otherwise, offset to data is an offset from byte 164 of the CMR (i.e. from the start of CMRData).

Field type X'05' (BYTE) is used for the tags whose data has a defined structure, such as OID, Date and Time Stamp, ICC Profile Data, and Link LUT tags. Field type X'06' (ASCII) is defined in the MO:DCA architecture with encoding scheme ID X'2100' – PC-Data, single byte. UBIN is defined as unsigned binary.

Tags X'F000'–X'FFFE' are private tags. Organizations can use a private tag in this range for their exclusive use without disclosing the tag contents. The architecture requires that such tag be non-essential, in the sense that any receivers not supporting the tag will not fail on parsing or using the resource.

X'EF00'–X'EFFF' are reserved for error handling for the CMR header.

The tags in a CMR must be specified in increasing order by their TagIDs. If they are out of order, the Exception EC-xxxx0F exists. Unless otherwise specified within the individual tag description, each TagID can appear at most once and Exception EC-xxxx0F exists if it is specified more than once. Multiple tags with the same ID may be accepted if the particular tag description explicitly states that it may repeat. The description in the tag must explain how the multiple tags are used and which one wins in cases of conflict. Tag values in the CMR tags are listed in the tag registry, that can be found in Appendix A. Private tags are ignored. Any TagID not supported by the device causes the Exception EC-xxxx04. The last tag must be the End Data tag (TagID of X'FFFF'), or exception EC-FFFF0E exists.

There is no restriction on where the actual data fields are located, as long as they are within the CMRData field scope. Note that all the offsets are from the beginning of the CMRData field, so that the location of the CMR header can be changed without any need to update the ValueOffset values. The offsets do not have to be increasing as the tag IDs increase, nor do they have to follow any other rule. There is no requirement that all the data in the scope be used, that is, it is permissible to have data not referenced by any tag.

The number of bytes of data for a tag is the value of Count multiplied by the size of each data item as indicated by Field Type. For example, a Count of 1 indicates two bytes if Field Type is X'02' (2-byte UBIN) or X'07' (UTF16).

Each type of CMR has a list of Mandatory Tags and a list of optional tags. The receivers should ignore any unknown tags. If an optional tag is not present, the default value (if one exists) should be used.

Exception Syntax

On encountering an error, an exception is raised. Each exception is defined by a three byte value. The first two bytes are the relevant Tag ID value (X'0000'–X'FFFF'), while the third byte is the exception code. The exception codes are defined as follows:

- X'04' – Unsupported TagID Value in a CMR tag
- X'05' – Invalid Count Value
- X'06' – Invalid Field Type
- X'0E' – Missing Required Tag
- X'0F' – Invalid Sequence
- X'10' – Invalid or unsupported field value or an offset that causes the tag data to start or end after the end of the CMR (as defined by the CMR length)
- X'11' – Inconsistent Tag Contents
- X'12' – Incorrect order of repeating groups
- X'13' – Duplicate value

Coordinate System

The CMR coordinate system is a two dimensional Cartesian coordinate system. The horizontal axis is labeled x, and the vertical axis is labeled y. Positive x is to the right of the origin, and positive y is above the origin. The measurement unit is pixel.

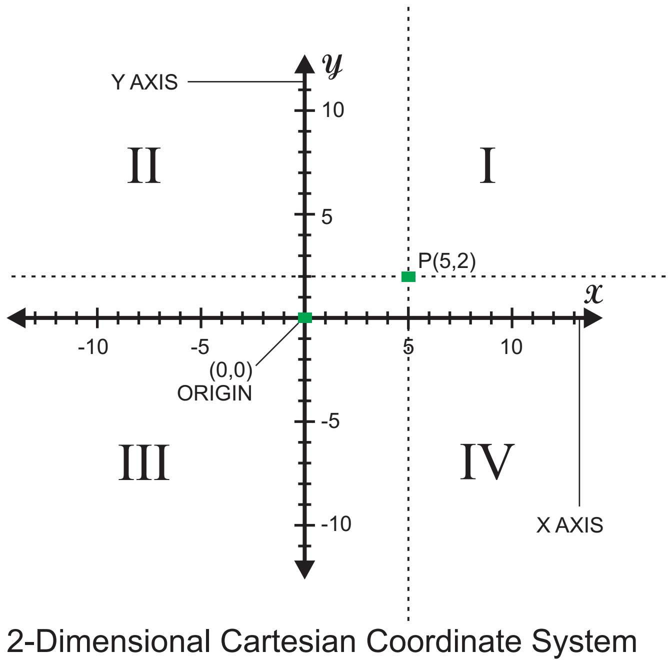


Figure 7. Cartesian Coordinate System

Tag Semantics

This section defines the CMR tags.

Comment

Tag ID X'0004'

Field Type X'06' (ASCII), X'07' (UTF16)

Count Number of Characters

This tag defines arbitrary comment text, ignored by receivers.

There is no default.

Exception Conditions:

EC-000406 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-00040F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-000410 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

Date and Time Stamp

Tag ID X'0008'

Field Type X'05' (BYTE)

Count 10

This tag contains the date and time of the creation of the CMR. It is defined consistently with the MO:DCA definition of the Universal Date and Time Stamp Triplet X'72' that is specified in accordance with the format defined in ISO 8601: 1988 (E). The tag is informational. The date and time values are not checked for validity.

Table 26. Date and Time Stamp Tag Syntax

Offset	Type	Name	Range	Meaning
0–1	2-byte UBIN	YearAD	0–65,535	Year AD using Gregorian calendar
2	1-byte UBIN	Month	1–12	Month of the year
3	1-byte UBIN	Day	1–31	Day of the month
4	1-byte UBIN	Hour	0–23	Hour of the day in 24-hour format
5	1-byte UBIN	Minute	0–59	Minute of the hour
6	1-byte UBIN	Second	0–59	Second of the minute
7	CODE	TimeZone	 X'00' X'01' X'02'	Relationship of time to UTC: Coordinated Universal Time Ahead of UTC Behind UTC
8	1-byte UBIN	UTCDiffH	0–23	Hours ahead of or behind UTC
9	1-byte UBIN	UTCDiffM	0–59	Minutes ahead of or behind UTC

YearAD Specifies the year AD using the Gregorian calendar. For example, the year 1999 is specified as X'07CF', the year 2000 as X'07D0', and the year 2001 as X'07D1'. Represents the YYYY component of a date in the format YYYYMMDD.

Month Specifies the month of the year. January is specified as X'01', and subsequent months are numbered in ascending order. Represents the MM component of a date in the format YYYYMMDD.

Day Specifies the day of the month. The first day of any month is specified as X'01', and subsequent days are numbered in ascending order. Represents the DD component of a date in the format YYYYMMDD. For example, the date December 31, 1999 is specified as X'07CF0C1F', and January 1, 2000 is specified as X'07D00101'.

Hour Specifies the hour of the day in 24 hour format. Represents the hh component of a time in the format hhmmss.

Minute Specifies the minute of the hour. Represents the mm component of a time in the format hhmmss.

Second Specifies the second of the minute. Represents the ss component of a time in the format hhmmss. For example, the time 4:35:21 PM is specified as X'102315'.

TimeZone Defines the relation of the specified time with respect to Coordinated Universal Time (UTC). This parameter, along with the UTCDiffH and UTCDiffM parameters, is used to accommodate differences between a specified local time and UTC because of time zones and daylight savings programs. For example, Mountain Time in the US is seven hours behind UTC when daylight savings is inactive, and six hours behind UTC when daylight savings is active.

Value	**Description**
X'00'	Time is specified in Coordinated Universal Time (UTC). With this value, the UTCDiffH and UTCDiffM parameters should be set to X'00'. When this time is displayed or printed, the equivalence with UTC time is normally indicated with a Z suffix, that is, hhmmssZ.
X'01'	Specified time is ahead of UTC. The number of hours ahead of UTC is specified by the UTCDiffH parameter; and the number of minutes ahead of UTC is specified by the UTCDiffM parameter. When this time is displayed or printed, the relationship with UTC time is normally indicated with a + character, followed by the actual time difference in hours and minutes, that is hhmmss+hhmm.
X'02'	Specified time is behind UTC. The number of hours behind UTC is specified by the UTCDiffH parameter; and the number of minutes behind UTC is specified by the UTCDiffM parameter. When this time is displayed or printed, the relationship with UTC time is normally indicated with a − character, followed by the actual time difference in hours and minutes, that is hhmmss−hhmm.
All others	Reserved

There is no default.

Exception Conditions:

EC-000805 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-000806 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-00080F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-000810 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

Number of Components

Tag ID X'0011'

Field Type X'01' (1-byte UBIN)

Count 1

This tag defines the number of color components referenced by this resource. To comply with ICC, the number of components must be in the range of 1–15. The ordering sequences of different color spaces are listed in Table 27.

Table 27. Ordering Sequence of Color Spaces

Color Space	Component 1	Component 2	Component 3	Component 4
XYZ	X	Y	Z	
Lab	L	a	b	
Luv	L	U	v	
Yxy	Y	X	y	
YCbr	Y	Cb	Cr	
RGB	R	G	B	
GRAY	K			
HSV	H	S	V	
HLS	H	L	S	
CMYK	C	M	Y	K
CMY	C	M	Y	
2CLR	Component 1	Component 2		
3CLR	Component 1	Component 2	Component 3	
4CLR	Component 1	Component 2	Component 3	Component 4

The components are numbered according to the order in the ICC data tag. Additional color spaces can be added simply by defining the signature component assignments.

The default is 1.

Exception Conditions:

EC-001105 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-001106 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-00110F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-001110 Invalid Value

The specified number of components is zero or greater than 15.

Halftone Subset

Tag ID X'1011'

Field Type X'08' (CODE)

Count 1

This tag denotes a subset of the Halftone CMR type. Currently, point-operation halftones and error diffusion halftones are defined in the Halftone Subset. The point-operation halftone operates only on the input pixel and not its neighbors. It includes clustered-dot, dispersed-dot, and stochastic halftones. The error diffusion halftone requires neighborhood operations and thresholding. Each subset has a list of required and optional CMR tags. The subsets are defined in Table 28.

Table 28. Halftone Subsets

Subset ID	Name
X'01'	Bilevel Point-Operation Halftone
X'02'	Multilevel Point-Operation Halftone
X'03'	Bilevel Error Diffusion Halftone
X'04'	Multilevel Error Diffusion Halftone

There is no default.

Exception Conditions:

EC-101105 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-101106 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10110E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10110F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-101110 Invalid Value

The specified subset value is none of X'01', X'02', X'03', or X'04'.

Array Width

Tag ID X'1021'

Field Type X'01' (1-byte UBIN), X'02' (2-byte UBIN)

Count Number Of Color Components

This tag defines the width of the array along the x-direction in pixels for each color component. It represents the screen width for the point-operation halftones, and the error diffusion filter width for the error diffusion halftones. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag. Each specified width must be greater than zero.

Note: For processing efficiency, the values of Array Width for Error Diffusion Filter should be small, preferably less than 255.

There is no default.

Exception Conditions:

EC-102106 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10210E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10210F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-102110 Invalid Value

One or more width values are zero or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-102111 Inconsistent Tag Contents

The count is inconsistent with the number of color components

Array Height

Tag ID X'1025'

Field Type X'01' (1-byte UBIN), X'02' (2-byte UBIN)

Count Number Of Color Components

This tag defines the height of the array along the y-direction in pixels for each color component. It represents the screen height for the point operation halftones, and the error diffusion filter height for the error diffusion halftones. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag. Each specified height must be greater than zero.

| **Note:** For processing efficiency, the values of Array Height for Error Diffusion Filter should be small, preferably less than 255.

There is no default.

Exception Conditions:

EC-102506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10250E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10250F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-102510 Invalid Value

One or more height values are zero or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-102511 Inconsistent Tag Contents

The count is inconsistent with the number of color components.

Max Image Value

Tag ID X'1030'

Field Type X'01' (1-byte UBIN), X'02' (2-byte UBIN), X'04' (4-byte UBIN)

Count Number Of Color Components

This tag defines the maximum input image value per component. For instance, the maximum Max Image Value of an 8-bit contone image is 255, but the Max Image Value could be 255, 252, 200, etc. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag. Each specified Max Image Value must be greater than 0.

There is no default.

Exception Conditions:

EC-103006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10300E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10300F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-103010 Invalid Value

One or more Max Image Values are zero or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-103011 Inconsistent Tag Contents

The count is inconsistent with the number of color components.

Number of Device Levels

Tag ID X'1035'

Field Type X'01' (1-byte UBIN)

Count Number Of Color Components

This tag defines the number of device levels per component for multilevel devices. The device level starts with 0 and ends with the number of device levels minus 1. For example, if the number of device levels is equal to 3, then the device levels are 0, 1, and 2. Each specified number of device levels must be greater than 2. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag.

There is no default.

Exception Conditions:

EC-103506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10350E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10350F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-103510 Invalid Value

One or more Number of Device Levels are smaller than 3 or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-103511 Inconsistent Tag Contents

The count is inconsistent with the number of color components.

Offset Tiling

Tag ID X'1040'

Field Type X'01' (1-byte UBIN), X'02' (2-byte UBIN)

Count Number Of Color Components

This tag defines the amount of shift in pixels between the halftone tiles in two adjacent rows for each component. The first row of tiles is arranged across device space pixels with the upper left pixel of the threshold array coincident with the upper left pixel of the image. The offset specifies the shift to the right of each subsequent row of tiles below the top row of tiles. The offset is specified with respect to the left side of a complete tile to the left side of the first tile in the row of tiles underneath. Each row of tiles is right shifted the same amount relative to the row above it. An example of Offset Tiling for an offset of two is illustrated in Figure 8. Partial tiles tile across the remaining device pixels. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag.

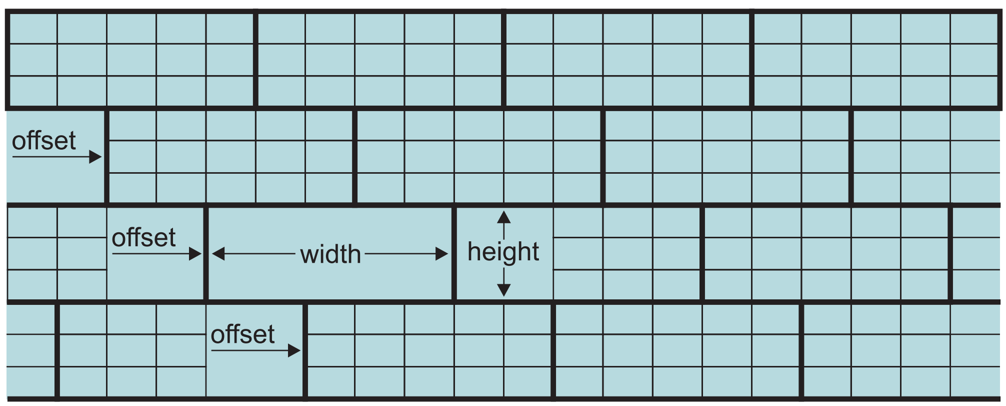


Figure 8. Illustration of Offset Tiling with Offset Tiling=2

The default is zero.

Exception Conditions:

EC-104006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10400F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-104010 Invalid Value

One or more Offset Tiling values are greater than the array width or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-104011 Inconsistent Tag Contents

The count is inconsistent with the number of color components.

Bilevel Point-Operation Screen Data

Tag ID X'1045'

Field Type X'01' (1-byte UBIN), X'02' (2-byte UBIN), X'04' (4-byte UBIN)

Count the sum of (*array width * array height*) over all color components

This tag specifies the threshold array values for each screen. Each screen has array width times array height entries of the specified size, arranged in row-major format. The data is component-wise structured: that is, it starts with the first component's threshold array followed by the second, and so on. The order of the components is specified by the Number of Components tag. If the input pixel level is less than the pixel value in the threshold array, the output pixel value should be B'0'; otherwise it should be B'1'.

Each halftone screen is developed for a particular output device. The inputs to the threshold array will be values expressed in the color space of the device (CMYK or RGB) and the meaning of the values in the threshold array depend on the device's color space. For CMYK devices, 255 or (2^16-1) or (2^32 -1) represents black. For RGB devices, those values represent white. Similarly, the meaning of the output value depends on the device's color space. For CMYK devices, B'0' represents white. For RGB devices, B'0' represents black.

There is no default.

Exception Conditions:

EC-104506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10450E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10450F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-104510 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

EC-104511 Inconsistent Tag Contents

The count is inconsistent with *array width * array height*, or the number of color components.

Multilevel Point-Operation Screen Data

Tag ID X'1050'

Field Type X'01' (1-byte UBIN)

Count the sum of (*array width* * *array height* * (*Max Image Value* + *1*)) over all color components

This tag gives the device gray level for each pixel. Each screen is a 3-d table lookup, defined as:

- o(x, y) = f (x', y', g)
- x' = x mod m
- y' = y mod n

Where:

- x, y = position of the pixel
- g = the input gray (or color) of the pixel at position (x, y), the maximum g is Max Image Value.
- o = the output multilevel at position (x, y)
- x' and y' are the reduced coordinate
- m and n are the array width and array height respectively

The data is structured component-wise, that is, it starts with the first component's screen followed by the second, and so on. The output data must be no greater than the number of device levels-1. The dimensions of the 3-d array are m x n x (Max Image Value +1). Offset Tiling applies if the tables are tiled with a shift.

There is no default.

Exception Conditions:

EC-105006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10500E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10500F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-105010 Invalid Value

One or more data values are greater than (Number of Device Levels -1) or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-105011 Inconsistent Tag Contents

The count is inconsistent with array width * array height * (Max Image Value + 1), or the number of color components.

Error Diffusion Filter

Tag ID X'1055'

Field Type X'01' (1-byte UBIN)

Count the sum of (*array width* * *array height*) over all color components

This tag specifies a set of values in the error diffusion filter. The values are arranged in a 2-dimensional array for each color plane. Each filter has array width times array height entries of the specified size, arranged in row-major format. The data is component-wise structured: that is, it starts with the first component's error diffusion filter array values followed by the second, and so on. The order of the components is specified by the Number of Components tag.

The value in the error diffusion filter is called weight representing the proportion of the error distributed to the pixel in that position. B'0's are assigned to the pixel that is currently processed, the pixels that are processed before the current pixel, or the pixels that no error is distributed to. The error distributed to a pixel is the weight of that pixel divided by the total weight, that is the sum of all values in the filter.

Error diffusion propagates the error between the initial value and the corrected value of each pixel to the neighboring pixels that are still to be processed. The error is defined as:

error = initial value - corrected value

Where the initial value is the sum of the original image value plus the errors if errors have been propagated to this pixel from the previously processed neighboring pixels. The corrected value is this value after threshold. For example, for a binary device, the threshold value is 128, and the current pixel value is 140. Then:

- initial value = 140
- corrected value = 255 (since 140>128)
- error = 140 - 255 = -115

This error is then propagated to its surrounding future pixels. Below is the example of Floyd-Steinberg filter:

The filter is defined as:

```
0 0 7
3 5 1
```

The errors are distributed as illustrated in Figure 9 on page 54.

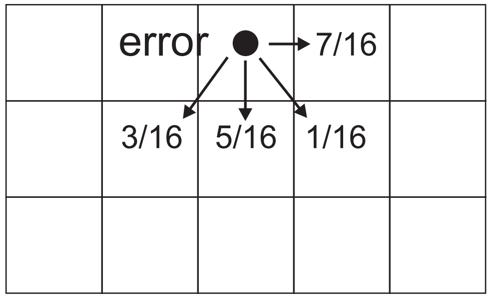


Figure 9. Illustration of Error Distribution with Floyd-Steinberg Filter

There is no default.

Exception Conditions:

EC-105506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10550E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10550F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-105510 Invalid Value

All data values are equal to 0 or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-105511 Inconsistent Tag Contents

The count is inconsistent with *array width * array height*, or the number of color components.

Location of Current Pixel

TagID X'1060'

Field Type X'01' (1-byte UBIN)

Count 2*Number of Color Components

This tag specifies a pair of values describing the location of the pixel that is currently being processed in an error diffusion filter. The first value indicates the number of the row, and the second value indicates the number of the column, where the current processed pixel is located. The rows and columns are numbered starting with 1. The data is component-wise structured: that is, it starts with the first component's location indices followed by the second, and so on. The count must be equal to two times the number of color components referenced by this resource. When multiple components are specified, the order of the components is specified by the Number of Components tag.

There is no default.

Exception Conditions:

EC-106005 Invalid Count Value

The specified Count field value is invalid for the tag. The Count is not a multiple of two.

EC-106006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10600E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10600F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-106010 Invalid Value

The first index value is either zero or greater than the array height, the second index value is either zero or greater than the array width, or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-106011 Inconsistent Tag Contents

The count is inconsistent with the number of color components

Raster Direction

Tag ID X'1065'

Field Type X'08' (CODE)

Count Number of Color Components

This tag denotes the raster direction for the error diffusion filter processing. Currently, two directions are defined: the normal raster direction and the serpentine raster direction. The normal raster direction alternates left-to-right and top-to-bottom. The serpentine raster direction alternates left-to-right, then right-to-left. The raster direction is always left to right when the first row of the input image is being processed. The error diffusion filter needs to be flipped 180 degree when it processes right-to-left direction. For example, the error diffusion filter of Floyd-Steinberg is:

```
0 x 7
3 5 1
```

Where x is the current pixel. The filter becomes

```
7 x 0
1 5 3
```

when the raster direction is right-to-left.

Table 29. Raster Direction Subsets

Subset ID	Name
X'01'	Normal raster
X'02'	Serpentine raster

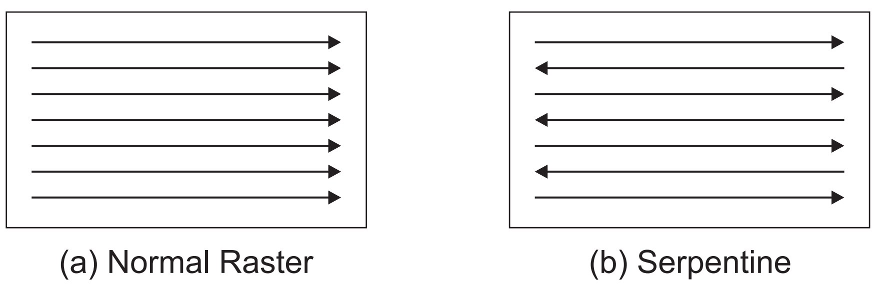


Figure 10. Illustration of the Normal Raster and Serpentine Raster

There is no default.

Exception Conditions:

EC-106506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10650E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10650F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-106510 Invalid Value

The specified subset value is neither X'01' nor X'02' or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-106511 Inconsistent Tag Contents

The count is inconsistent with the number of color components

Boundary Condition

Tag ID X'1070'

Field Type X'08' (CODE)

Count Number of Color Components

This tag denotes the boundary conditions for the error diffusion halftones. It defines the assumed values for the image when the error diffusion filter crosses the boundary of the image. The boundary conditions apply to the top, the left, and the right boundary of the image. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag.

Four boundary conditions are defined: none, zero boundary, reflect, and periodic.

- None: no boundary condition is applied.
- Zero boundary: zeros are assigned for the image values outside the boundary.
- Reflect: image values that reflect at the boundary are assigned for the image values outside the boundary. For example, if the image values in one scan line are 1 2 3...6 7 8, then the image values outside the right boundary are 8 7 6 ...
- Periodic: image values that wrap around the axis in the same scan line are assigned for the image values outside the boundary. For example, if the image values in one scan line are 1 2 3...5 6 7 8, then the image values outside the right boundary are 1 2 3 ...

The boundary conditions are defined in Table 30.

Table 30. Boundary Conditions

Subset ID	Name
X'01'	None
X'02'	Zero boundary
X'03'	Reflect
X'04'	Periodic

There is no default.

Exception Conditions:

EC-107006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10700E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10700F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-107010 Invalid Value

The specified subset value is none of the subsets X'01', X'02', X'03', and X'04', or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-107011 Inconsistent Tag Contents

The count is inconsistent with the number of color components

Threshold Value

TagID X'1075'

Field Type X'01' (1-byte UBIN), X'02' (2-byte UBIN), X'04' (4-byte UBIN)

Count Number Of Color Components

This tag specifies a single threshold value for the bilevel error diffusion halftones. If the initial pixel value is less than the threshold value, the output pixel value should be B'0'; otherwise it should be B'1'. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag.

Each halftone screen is developed for a particular output device. The values being compared to the threshold will be values expressed in the color space of the device (CMYK or RGB) and the meaning of the values in the threshold depends on the device's color space. For CMYK devices, 255 or (2^16-1) or (2^32 -1) represents black. For RGB devices, those values represent white. Similarly, the meaning of the output value depends on the device's color space. For CMYK devices, B'0' represents white. For RGB devices, B'0' represents black.

There is no default.

Exception Conditions:

EC-107506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10750E Missing Required Tag

The tag is required for the resource, but is missing.

EC-10750F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-107510 Invalid Value

One or more values are equal to zero or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-107511 Inconsistent Tag Contents

The count is inconsistent with the number of color components

Quantization Boundary Table

TagID X'1080'

Field Type X'01' (1-byte UBIN), X'02' (2-byte UBIN), X'04' (4-byte UBIN)

Count the sum of (the number of device levels -1) over all color components

The tag specifies *n* one-dimensional arrays for the multilevel error diffusion halftone, where *n* is the number of color components. The array entries are threshold values for input pixels. The length of the array for each component is equal to the number of device levels minus 1 for that component. The indices (*i*) of the array are in the range [1, number of device levels-1]. For a threshold value T_i in the array, where 1≤i≤number of device levels -1, T_i is the threshold value defined between device level i-1 and i. That is, the first value in the array is the threshold value defined between device level 0 and 1, the second value in the array is the threshold value defined between device level 1 and 2, and so on. The values in the entries thus are monotonically increasing. The digital value corresponds to a device level *i* is defined as $(2^n-1)*i/$(number of device levels-1) rounded to the nearest integer, where n is the number of bits in Field Type.

For a pixel value I:

- If $I<T_1$, the corrected value after threshold is equal to the digital value corresponding to the device level zero.
- If $T_i \leq I < T_{i+1}$, the corrected value after threshold is equal to the digital value corresponding to the device level i.
- If $I \geq T_{j}$, where j is equal to the number of device levels minus 1, the corrected value after threshold is equal to the digital value corresponding to the maximum device level.

When multiple components are specified, the order of the components is specified by the Number of Components tag. The data is structured component-wise, that is, it starts with the first component array followed by the second, and so on.

Table 31 is an example of Quantization Boundary Table. The number of device levels = 4, and the field type is X'01' (1-byte UBIN). The array entries are: 60, 120, and 200, where 60 is the threshold value defined between the device level 0 and 1, 120 is the threshold value defined between the device level 1 and 2, and 200 is the threshold value defined between device level 2 and 3.

Table 31. Illustration of Quantization Boundary Table

Index (i)	Array Entry (threshold value)
1	60
2	120
3	200

Table 32. Implementation of Quantization Boundary Table

Initial Value	Output Device Level	Corrected value
I∈ [0, 60)	0	0
I∈ [60, 120)	1	85
I∈ [120, 200)	2	170
I∈ [200, 255]	3	255

There is no default.

Exception Conditions:

EC-108006
Invalid Field Type

The specified Field Type is invalid for the tag.

EC-10800E
Missing Required Tag

The tag is required for the resource, but is missing.

EC-10800F
Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-108010
Invalid Value

One or more values are equal to zero, the values in the table are not monotonically increasing, or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-108011
Inconsistent Tag Contents

The count is inconsistent with *the number of device levels -1* or the number of color components.

Tone Transfer Curve Subset

Tag ID X'2004'

Field Type X'08' (CODE)

Count 1

This tag denotes a subset of the Tone Transfer Curve CMR type. There are two tone transfer curve subsets defined: ToneTransferCurve Array and ToneTransferCurve Identity. The ToneTransferCurve Array defines transfer curves that map from the input space to a modified space. If the Tone Transfer Curve Subset is the identity, then no tone transfer curve is to be applied, that is, no data is sent with the ToneTransferCurve Identity subset.

Table 33. Tone Transfer Curve Subsets

Value	Name
X'01'	ToneTransferCurve Array
X'02'	ToneTransferCurve Identity

There is no default.

Exception Conditions:

EC-200405 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-200406 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-20040E Missing Required Tag

The tag is required for the resource, but is missing.

EC-20040F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-200410 Invalid Value

The specified subset value is neither X'01' nor X'02'.

Tone Transfer Curve Length

Tag ID X'2011'

Field Type X'08' (CODE)

Count Number Of Color Components

This tag gives the number of entries in the tone transfer curve for each component. The count must be equal to the number of color components referenced by this resource, and must match the value specified by the Number of Components tag or its default. When multiple components are specified, the order of the components is specified by the Number of Components tag. The only values allowed for length are encoded as shown in Table 34.

Table 34. Tone Transfer Curve Length Values

Value	Meaning
X'01'	256 1-byte entries in tone transfer curve
X'02'	65,536 2-byte entries in tone transfer curve

The default is X'01' (256 entries) for each component.

Architecture note: It is possible that the desired tone transfer curve has a number of entries that are less than 256 or 65,536, say 250. If this were allowed and input image data having a value of 253 were received, special handling would be required. To avoid this, the options for length are limited to 256 and 65,536. The application or color management software can adjust the desired tone transfer curve to meet this requirement. For instance, entries can be added at the end. These entries would all have the same value as the last value in the original tone transfer curve. Another approach would be to scale the original tone transfer curve so that it has 256 entries.

Exception Conditions:

EC-201106 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-20110F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-201110 Invalid Value

The specified value is not X'01' or X'02', or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-201111 Inconsistent Tag Contents

The count is inconsistent with the number of color components.

Tone Transfer Curve Data

Tag ID X'2015'

Field Type X'05' (byte)

Count the total length of the data (see below)

This tag gives the data for all tone transfer curves. The Tone Transfer Curve Data has *n* one-dimensional LUTs, where *n* is the number of color components. When multiple components are specified, the order of the components is specified by the Number of Components tag. The data is structured component-wise, that is, it starts with the first component tone transfer curve data followed by the second, and so on. The length of each curve is given by the corresponding Tone Transfer Curve Length tag (256 or 65,536). The index for the tone transfer curve starts with 0 and ends with the value specified by the Tone Transfer Curve Length tag minus 1.

The **Count** is defined to be the total length of the data. This total length is the sum of the lengths of each tone transfer curve over the number of color components. For each color component:

- If the Tone Transfer Curve Length tag specifies X'01' (256 1-byte entries) for that component, then the length of the tone transfer curve is 256 bytes.
- If the Tone Transfer Curve Length tag specifies X'02' (65,536 2-byte entries) for that component, then the length of the tone transfer curve is 131,072 bytes.

This tag always contains an offset to the data.

There is no default.

Tone Transfer Curve usage

Tone transfer curves are used as lookup tables to correct or calibrate the pixel values prior to output or halftoning. For each pixel component value, the device will choose the corresponding curve and use the value as an index. The value found at that index will be used instead of the original value.

The input and output of the tone transfer curve are interpreted to be in the color space of the device. For CMYK devices, 255 or 65,535 represent black. For RGB devices, they represent white.

Architecture note: This is different from PostScript, where the transfer function is always interpreted as if the component were additive and where subtractive input and output must be converted.

Mismatches between the data type of the Tone Transfer Curve and the type of the input image data will be handled as follows. For each color component:

- Tone Transfer Curve Length is X'01' (data size is one byte) and image data is 2-byte values: Each image value should be truncated to be the upper 8 bits of the value, that is, the image value is divided by 256.
- Tone Transfer Curve Length is X'02' (data size is two bytes) and image data type is 1-byte: Each image value should be converted to a 2-byte value by adding 8 bits of zero on the right side (by multiplying by 256). Alternatively and equivalently, the tone transfer curve can be collapsed to include only entries from the original array whose indices were multiples of 256.
- A similar process would be followed for image data having 4-byte values.

Tone Transfer Curve Data tag

Exception Conditions:

EC-201506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-20150E Missing Required Tag

The tag is required for the resource, but is missing.

EC-20150F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-201510 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

EC-201511 Inconsistent Tag Contents

The count is inconsistent with the tone transfer curve length, or the number of color components.

Inverse Tone Transfer Curve Data

Tag ID X'2020'

Field Type X'05' (Byte)

Count the total length of the data (see below)

This tag represents the inverse of the Tone Transfer Curve Data, that is another part of the same Tone Transfer Curve CMR. It is an optional tag. If it is used, its data must be created by inverting the tone transfer curve lookup table. Because the inverse is not a well-defined function, this tag allows the application or color management system to clearly define the inverse. The Inverse Tone Transfer Curve Data has n one-dimensional LUTs, where n is the number of color components. When multiple components are specified, the order of the components is specified by the Number of Components tag. The data is structured component-wise, that is, it starts with the first component tone transfer curve data followed by the second, and so on. The length of each curve is given by the corresponding Tone Transfer Curve Length tag (256 or 65,536).

The **Count** is defined to be the total length of the data. This total length is the sum of the lengths of each tone transfer curve over the number of color components. For each color component:

- If the Tone Transfer Curve Length tag specifies X’01’ (256 1-byte entries) for that component, then the length of the tone transfer curve is 256 bytes.
- If the Tone Transfer Curve Length tag specifies X’02’ (65,536 2-byte entries) for that component, then the length of the tone transfer curve is 131,072 bytes.

This tag always contains an offset to the data.

There is no default.

Inverse Tone Transfer Curve usage

The inverse color calibration data is used when the Tone Transfer Curve CMR is being used as an audit CMR and it is desired to undo the tone transfer curve that was applied to the image data. The inverse tone transfer curves are used as lookup tables to convert or calibrate the input pixel values back to the original data. For each pixel component value, the device chooses the corresponding curve and uses the value as an index. The value found at that index is used instead of the original value. Exact details for processing are the same as those for the Tone Transfer Curve Data.

The input and output of the inverse tone transfer curve are interpreted to be in the color space of the input. For CMYK devices, 255 or 65,535 represent black. For RGB devices, they represent white.

Architecture note: This is different from Post Script, where the transfer function is always interpreted as if the component were additive and where subtractive input and output must be converted.

Exception Conditions:

EC-202006 Invalid Field Type

The specified Field Type is invalid for the tag.

Inverse Tone Transfer Curve Data tag

EC-20200F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-202010 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

EC-202011 Inconsistent Tag Contents

The count is inconsistent with the tone transfer curve length, or the number of color components.

ICC Profile Subset

Tag ID X'3011'

Field Type X'08' (CODE)

Count 1

This tag denotes a subset of the ICC profile. Each subset has a list of mandatory and optional ICCHeaderFields, and ICCtags. The receiver will ignore any other tags. The subsets are defined in Table 35.

Table 35. ICC Profile Subsets

Subset ID	Name
X'01'	Monochrome input profile
X'02'	Monochrome display profile
X'03'	Monochrome output profile
X'04'	Three-component matrix-based input profile
X'05'	Three-component matrix-based display input profile
X'06'	N-component LUT-based input profile
X'07'	N-component LUT-based display profile
X'08'	N-Component LUT-based output profiles
X'09'	ColorSpace conversion profile
X'0A'	Retired item 3 (Abstract profile)

There is no default.

Exception Conditions:

EC-301105 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-301106 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-30110E Missing Required Tag

The tag is required for the resource, but is missing.

EC-30110F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-301110 Invalid Value

The specified subset value is invalid.

ICC Profile Data

Tag ID X'3015'

Field Type X'05' (BYTE)

Count N, where N is the number of bytes in the profile

This tag contains a complete encapsulated ICC profile. The ICC profile is interpreted based on the ICC Profile Subset tag. Each subset denotes a subset of the ICC specification, listing required and optional tags and ICCHeaderFields. Any other ICC profile tags are ignored.

Table 36. ICC Header Fields

Byte Offset	Content
0–3	Profile size
4–7	CMM Type signature
8–11	Profile version number
12–15	Profile/Device Class signature
16–19	Color Space of Data (possibly a derived space) [“the canonical input space”]
20–23	Profile Connection Space (PCS) [“the canonical output space”]
24–35	Date and time this profile was first created
36–39	acsp (61637370h) profile file signature
40–43	Primary Platform signature
44–47	Flags to indicate various options for the CMM such as distributed processing and caching options
48–51	Device manufacturer of the device for which this profile is created
52–55	Device model of the device for which this profile is created
56–63	Device attributes unique to the particular device setup such as media type
64–67	Rendering Intent
68–79	The XYZ values of the illuminant of the Profile Connection Space. These values must correspond to D50.
80–83	Profile Creator signature
84–99	Profile ID
100–127	28 bytes reserved for future expansion. These bytes must be set to zeros There is no default.

Exception Conditions:

EC-301505 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-301506 Invalid Field Type

The specified Field Type is invalid for the tag or the header content.

EC-30150E Missing Required Tag

The tag or the header content is required for the resource, but is missing.

EC-30150F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-301510 Invalid Value

An ICC tag or an ICCHeaderField content required for this subset of the ICC profile is missing from the encapsulated ICC profile or the offset caused some portion of the tag data to be outside of the CMRdata.

ICC Profile Filename

Tag ID X'3025'

Field Type X'06' (ASCII), X'07' (UTF16)

Count Number of Characters

This tag holds the filename of the ICC Profile that was used to create this CMR. The ICC Profile that was in that file is copied into the ICC Profile Data Tag.

Exception Conditions:

EC-302506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-30250F Invalid sequence

The tag has been encountered out of sequence or more than once.

EC-302510 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

Link Color Conversion Subset

Tag ID X'4011'

Field Type X'08' (CODE)

Count 1

This tag defines a subset of the Link Color Conversion CMR type. The LinkColorConversion LUT subset combines an audit Color Conversion CMR with an instruction Color Conversion CMR. The LinkColorConversion Identity subset is used when the input space of the data is the same as the devices output space and no color conversion is to be done on the data. The ICC DeviceLink subset provides a direct conversion from input to output space with no involvement of audit and instruction Color Conversion CMRs.

Table 37. Link Color Conversion Subsets

Subset ID	Name
X'01'	LinkColorConversion LUT
X'02'	LinkColorConversion Identity
X'03'	ICC DeviceLink

There is no default.

Exception Conditions:

EC-401105 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-401106 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40110E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40110F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-401110 Invalid or Unsupported Value

The specified value is neither X'01', X'02', nor X'03'. This error is also reported if the CMRType in the CMR Header does not match the subset value. It is an error if:

- CMRType = LK and Subset ID is not X'01' or X'02'
- CMRType = DL and Subset ID is not X'03'

Note: It is possible that some implementations were complete before subset X'03' was added to the architecture and they may NACK if it is specified. Server software should verify that the STM Device-Control command-set vector shows support for the ICC DeviceLink subset (X'E006') before sending down a CMR with the CMRType field of the CMR header showing “DL” (ICC DeviceLink).

Link Audit CMR OID

Tag ID X'4015'

Field Type X'05' (BYTE)

Count Number of bytes in the OID

This tag defines the OID (Object Identifier) of the audit Color Conversion CMR used in the Link Color Conversion CMR. The OID is used to provide a universally unique identifier for the audit CMR.

There is no default.

Exception Conditions:

EC-401506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40150E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40150F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-401510 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

Link Instruction CMR OID

Tag ID X'4020'

Field Type X'05' (BYTE)

Count Number of bytes in the OID

This tag defines the OID (Object Identifier) of the instruction Color Conversion CMR used in the link color conversion CMR. The OID is used to provide a universally unique identifier for the instruction CMR.

There is no default.

Exception Conditions:

EC-402006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40200E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40200F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-402010 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

EC-402011 Inconsistent Tag Contents

OID tags are different in the LinkColorConversion Identity.

Link Audit CMR Name

Tag ID X'4025'

Field Type X'07' (UTF16)

Count Number of Characters

This tag specifies the name of the audit Color Conversion CMR used in the Link Color Conversion CMR. The tag is informational and is not checked for validity.

There is no default.

Exception Conditions:

EC-402506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40250E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40250F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-402510 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

Link Instruction CMR Name

Tag ID X'4030'

Field Type X'07' (UTF16)

Count Number of Characters

This tag specifies the name of the instruction Color Conversion CMR used in the Link Color Conversion CMR. The name is informational and is not checked for validity.

There is no default.

Exception Conditions:

EC-403006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40300E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40300F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-403010 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

Default Rendering Intent

Tag ID X'4035'

Field Type X'08' (CODE)

Count 1

This tag defines the rendering intent that was found in the ICCHeaderFields in the instruction Color Conversion CMR. The defined values are consistent with the ICC rendering intents.

Table 38. ICC Rendering Intents

Rendering Intent	Value
Perceptual	X'00'
Media-Relative Colorimetric	X'01'
Saturation	X'02'
ICC-Absolute Colorimetric	X'03'

There is no default.

Exception Conditions:

EC-403505 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-403506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40350E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40350F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-403510 Invalid Value

The specified value is invalid.

Link LUT Perceptual

Tag ID X'4040'

Field Type X'05' (BYTE)

Count N, where N is the number of bytes in the LUT + 20 bytes of the header

This tag defines a lookup table that converts the input color space of an audit Color Conversion CMR into the output color space of an instruction Color Conversion CMR. The rendering intent of the LUT is perceptual. However, flags can be set to indicate that the LUT is also used for other rendering intents (media-relative colorimetric, saturation, or ICC-absolute colorimetric). The lookup table (LUT) has an input table and an output table. The dimension of the input table is n (rows) by m (columns), where n is the number of data points, that is, the product of the numbers of the grid points over the number of components of the input color space, and m is the number of components of the input color space in the audit Color Conversion CMR. The dimension corresponding to the first input component varies least rapidly and the dimension corresponding to the last input component varies most rapidly. The grid point values in each color component of input color space are obtained by using the following equation:

E_i = M*i/(q-1),

where i is the i^{th} grid point of that color component (i starts from 0 and ends with q-1) and q is the number of the grid points for that color component. M is 255 for 1-byte UBIN and 65,535 for 2-byte UBIN.

The output table is in form of an n (rows) by p (columns) array, where p is the number of components of the output color space in the instruction Color Conversion CMR. Each entry in the output table is a function value of the corresponding entry in the input table. Only the output table is shown in the LUT. The table values are arrays of 8 bit or 16 bit values.

The size of the LUT = n*p*(precision of data elements in bytes).

Table 39. The Link LUT Perceptual Encoding

Bytes	Length	Type	Range	Meaning
0	1	1-byte UBIN	1–15	Number of components of input color space
1	1	1-byte UBIN	1–15	Number of components of output color space
2–16	15	For each entry: 1-byte UBIN	For each entry: 0–255	Number of grid points in each component of input color space. There are 15 entries, each encoded as one byte. Only the first m entries are used, where m is the number of components of the input color space. Unused entries should be set to zeros.
17	1	1-byte UBIN	 1 2	Precision of data elements: 1-byte UBIN 2-byte UBIN
18	1	BITS	Additional use flags: set to B'0' if false, set to B'1' if true	
bit 0			B'0', B'1'	Media-relative colorimetric

Table 39. The Link LUT Perceptual Encoding (continued)

Bytes	Length	Type	Range	Meaning
bit 1			B'0', B'1'	Saturation
bit 2			B'0', B'1'	ICC-absolute colorimetric
bit 3–7			B'00000'	Reserved
19	1		0	Reserved
20 to end		For each data point: 1-byte UBIN 2-byte UBIN	For each data point: 0–255 0–65,535	LUT data

Number of components of input color space
The number of components of the input color space in an audit Color Conversion CMR.

Number of components of output color space
The number of components of the output color space in an instruction Color Conversion CMR.

Number of grid points
Number of grid points in each component of the input color space. The number of grid points in each dimension is not necessarily the same. The decision on these numbers is implementation dependent. It could be different from the number of grid points in each dimension of the input color space in the audit color conversion. ICC allows a maximum of 15 color components in a color space. 15 bytes are allocated for this header field.

Precision of data elements
The entry values can be either 1 byte or 2 bytes. The decision is implementation dependent.

Additional use flags
Each flag indicates that the LUT is also used for another rendering intent (media-relative colorimetric, saturation, or ICC-absolute colorimetric).

Table data
The data in the LUT.

There is no default.

Exception Conditions:

EC-404005 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-404006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40400E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40400F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-404010 Invalid Value

The specified value is invalid or the offset caused some portion of the tag data to be outside of the CMRdata.

Link LUT Media-Relative Colorimetric

Tag ID X'4045'

Field Type X'05' (BYTE)

Count N, where N is the number of bytes in the LUT + 20 bytes of the header

This tag defines a lookup table that converts the input color space of an audit Color Conversion CMR into the output color space of an instruction Color Conversion CMR. The rendering intent of the LUT is media-relative colorimetric. However, flags can be set to indicate that the LUT is also used for other rendering intents (saturation or ICC-absolute colorimetric). The lookup table (LUT) has an input table and an output table. The dimension of the input table is n (rows) by m (columns), where n is the number of data points, that is, the product of the numbers of the grid points over the number of components of the input color space , and m is the number of components of the input color space in the audit Color Conversion CMR. The dimension corresponding to the first input component varies least rapidly and the dimension corresponding to the last input component varies most rapidly. The grid point values in each color component of input color space are obtained by using the following equation:

E_i = M*i/(q-1),

where i is the i^{th} grid point of that color component (i starts from 0 and ends with q-1) and q is the number of the grid points for that color component. M is 255 for 1-byte UBIN and 65,535 for 2-byte UBIN.

The output table is in form of an n (rows) by p (columns) array, where p is the number of components of the output color space in the instruction Color Conversion CMR. Each entry in the output table is a function value of the corresponding entry in the input table. Only the output table is shown in the LUT. The table values are arrays of 8 bit or 16 bit values.

The size of the LUT = n*p*(precision of data elements in bytes).

Table 40. The Link LUT Media-Relative Colorimetric Encoding

Bytes	Length	Type	Range	Meaning
0	1	1-byte UBIN	1–15	Number of components of input color space
1	1	1-byte UBIN	1–15	Number of components of output color space
2–16	15	For each entry: 1-byte UBIN	For each entry: 0–255	Number of grid points in each component of input color space. There are 15 entries, each encoded as one byte. Only the first m entries are used, where m is the number of components of the input color space. Unused entries should be set to zeros.
17	1	1-byte UBIN	 1 2	Precision of data elements: 1-byte UBIN 2-byte UBIN
18	1	BITS	Additional use flags: set to 0 if false, set to 1 if true	
bit 0			B'0'	Reserved

Table 40. The Link LUT Media-Relative Colorimetric Encoding (continued)

Bytes	Length	Type	Range	Meaning
bit 1			B'0', B'1'	Saturation
bit 2			B'0', B'1'	ICC-absolute colorimetric
bit 3–7			B'00000'	Reserved
19	1		0	Reserved
20 to end		For each data point: 1-byte UBIN 2-byte UBIN	For each data point: 0–255 0–65,535	LUT data

Number of components of input color space
The number of components of the input color space in an audit Color Conversion CMR.

Number of components of output color space
The number of components of the output color space in an instruction Color Conversion CMR.

Number of grid points
Number of grid points in each component of the input color space. The number of grid points in each dimension is not necessarily the same. The decision on these numbers is implementation dependent. It could be different from the number of grid points in each dimension of the input color space in the audit color conversion. ICC allows a maximum of 15 color components in a color space. 15 bytes are allocated for this header field.

Precision of data elements
The entry values can be either 1 byte or 2 bytes. The decision is implementation dependent.

Additional use flags
Each flag indicates that the LUT is also used for another rendering intent (saturation or ICC-absolute colorimetric).

Table data
The data in the LUT.

There is no default.

Exception Conditions:

EC-404505 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-404506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40450E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40450F Invalid Sequence

The tag has been encountered out of sequence or more than once.

Link LUT Media-Relative Colorimetric tag

EC-404510
Invalid Value

The specified value is invalid or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-404511
Inconsistent Tag Contents

The LUT was provided in a previous tag.

- A Link LUT tag is provided and the flag for the rendering intent of this LUT was set in a previous Link LUT tag
- A flag for a rendering intent is set in multiple Link LUT tags

Link LUT Saturation

Tag ID X'4050'

Field Type X'05' (BYTE)

Count N, where N is the number of bytes in the LUT + 20 bytes of the header

This tag defines a lookup table that converts the input color space of an audit Color Conversion CMR into the output color space of an instruction Color Conversion CMR. The rendering intent of the LUT is saturation. However, a flag can be set to indicate that the LUT is also used for ICC-absolute colorimetric. The lookup table (LUT) has an input table and an output table. The dimension of the input table is n (rows) by m (columns), where n is the number of data points, that is, the product of the numbers of the grid points over the number of components of the input color space , and m is the number of components of the input color space in the audit Color Conversion CMR. The dimension corresponding to the first input component varies least rapidly and the dimension corresponding to the last input component varies most rapidly. The grid point values in each color component of input color space are obtained by using the following equation:

$E_i = M*i/(q-1)$,

Where i is the i^{th} grid point of that color component (i starts from 0 and ends with q-1) and q is the number of the grid points for that color component. M is 255 for 1-byte UBIN and 65,535 for 2-byte UBIN.

The output table is in form of an n (rows) by p (columns) array, where p is the number of components of the output color space in the instruction Color Conversion CMR. Each entry in the output table is a function value of the corresponding entry in the input table. Only the output table is shown in the LUT. The table values are arrays of 8 bit or 16 bit values.

The size of the LUT = n*p*(precision of data elements in bytes).

Table 41. The Link LUT Saturation Encoding

Bytes	Length	Type	Range	Meaning
0	1	1-byte UBIN	1–15	Number of components of input color space
1	1	1-byte UBIN	1–15	Number of components of output color space
2–16	15	For each entry: 1-byte UBIN	For each entry: 0–255	Number of grid points in each component of input color space. There are 15 entries, each encoded as one byte. Only the first m entries are used, where m is the number of components of the input color space. Unused entries should be set to zeros.
17	1	1-byte UBIN	 1 2	Precision of data elements: 1-byte UBIN 2-byte UBIN
18	1	BITS	Additional use flags: set to 0 if false, set to 1 if true	
bits 0–1			B'00'	Reserved

Table 41. The Link LUT Saturation Encoding (continued)

Bytes	Length	Type	Range	Meaning
bit 2			B'0', B'1'	ICC-absolute colorimetric
bits 3–7			B'00000'	Reserved
19	1		0	Reserved
20 to end		For each data point: 1-byte UBIN 2-byte UBIN	For each data point: 0–255 0–65,535	LUT data

Number of components of input color space
The number of components of the input color space in an audit Color Conversion CMR.

Number of components of output color space
The number of components of the output color space in an instruction Color Conversion CMR.

Number of grid points
Number of grid points in each component of the input color space. The number of grid points in each dimension is not necessarily the same. The decision on these numbers is implementation dependent. It could be different from the number of grid points in each dimension of the input color space in the audit color conversion. ICC allows a maximum of 15 color components in a color space. 15 bytes are allocated for this header field.

Precision of data elements
The entry values can be either 1 byte or 2 bytes. The decision is implementation dependent.

Additional use flags
The flag indicates that the LUT is also used for ICC-absolute colorimetric.

Table data
The data in the LUT.

There is no default.

Exception Conditions:

EC-405005 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-405006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40500E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40500F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-405010 Invalid Value

The specified value is invalid or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-405011 Inconsistent Tag Contents

The LUT was provided in a previous tag.

- A Link LUT tag is provided and the flag for the rendering intent of this LUT was set in a previous Link LUT tag
- A flag for a rendering intent is set in multiple Link LUT tags

Link LUT ICC-Absolute Colorimetric

Tag ID X'4055'

Field Type X'05' (BYTE)

Count N, where N is the number of bytes in the LUT + 20 bytes of the header

This tag defines a lookup table that converts the input color space of an audit Color Conversion CMR into the output color space of an instruction Color Conversion CMR. The rendering intent of the LUT is ICC-Absolute colorimetric. The lookup table (LUT) has an input table and an output table. The dimension of the input table is n (rows) by m (columns), where n is the number of data points, that is, the product of the numbers of the grid points over the number of components of the input color space, and m is the number of components of the input color space in the audit Color Conversion CMR. The dimension corresponding to the first input component varies least rapidly and the dimension corresponding to the last input component varies most rapidly. The grid point values in each color component of input color space are obtained by using the following equation:

$E_i = M*i/(q-1)$,

wWhere i is the i^{th} grid point of that color component (i starts from 0 and ends with q-1) and q is the number of the grid points for that color component. M is 255 for 1-byte UBIN and 65,535 for 2-byte UBIN.

The output table is in form of an n (rows) by p (columns) array, where p is the number of components of the output color space in the instruction Color Conversion CMR. Each entry in the output table is a function value of the corresponding entry in the input table. Only the output table is shown in the LUT. The table values are arrays of 8 bit or 16 bit values.

The size of the LUT = n*p*(precision of data elements in bytes).

Table 42. The Link LUT ICC-Absolute Colorimetric Encoding

Bytes	Length	Type	Range	Meaning
0	1	1-byte UBIN	1–15	Number of components of input color space
1	1	1-byte UBIN	1–15	Number of components of output color space
2–16	15	For each entry: 1-byte UBIN	For each entry: 0–255	Number of grid points in each component of input color space. There are 15 entries, each encoded as one byte. Only the first m entries are used, where m is the number of components of the input color space. Unused entries should be set to zeros.
17	1	1-byte UBIN	 1 2	Precision of data elements: 1-byte UBIN 2-byte UBIN
18–19	2		0	Reserved

Table 42. The Link LUT ICC-Absolute Colorimetric Encoding (continued)

Bytes	Length	Type	Range	Meaning
20 to end		For each data point: 1-byte UBIN 2-byte UBIN	For each data point: 0–255 0–65,535	LUT data

Number of components of input color space
The number of components of the input color space in an audit Color Conversion CMR.

Number of components of output color space
The number of components of the output color space in an instruction Color Conversion CMR.

Number of grid points
Number of grid points in each component of the input color space. The number of grid points in each dimension is not necessarily the same. The decision on these numbers is implementation dependent. It could be different from the number of grid points in each dimension of the input color space in the audit color conversion. ICC allows a maximum of 15 color components in a color space. 15 bytes are allocated for this header field.

Precision of data elements
The entry values can be either 1 byte or 2 bytes. The decision is implementation dependent.

Table data
The data in the LUT.

There is no default.

Exception Conditions:

EC-405505 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-405506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40550E Missing Required Tag

The tag is required for the resource, but is missing.

EC-40550F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-405510 Invalid Value

The specified value is invalid or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-405511 Inconsistent Tag Contents

The LUT was provided in a previous tag for the cases:

- A Link LUT tag is provided and the flag for the rendering intent of this LUT was set in a previous Link LUT tag
- A flag for a rendering intent is set in multiple Link LUT tags

Link CMRE Identifier

Tag ID X'4090'

Field Type X'07' (UTF16)

Count Number of Characters

This tag specifies the name and version of the CMR Engine used to generate the Link Color Conversion CMR.

There is no default.

Exception Conditions:

EC-409006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-40900F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-409010 Invalid Value

The offset caused some portion of the tag data to be outside of the CMRdata.

Indexed Subset

Tag ID X’5011’

Field Type X’08’ (CODE)

Count 1

This tag denotes a subset of the Indexed CMR type. Currently, only one Indexed CMR subset is defined for the multi-output color spaces. It allows a mixture of different output color spaces for an Indexed CMR. The color spaces include gray, CMYK, RGB, CIELAB D50, and named colorants.

Table 43. Indexed CMR Subset

Subset ID	Name
X’01’	Multi-output color spaces

There is no default.

Exception Conditions:

EC-501105 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-501106 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-50110E Missing Required Tag

The tag is required for the resource, but is missing.

EC-50110F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-501110 Invalid Value

The specified subset value is not X'01'.

Number of Named Colorants

Tag ID X'5015'

Field Type X'01' (1-byte UBIN)

Count 1

The tag defines the number of named colorants referenced by this resource. This tag determines the number of repeating groups in the Colorant Identification List tag and the length of each repeating group in the Color Palette Named Colorants tag. The number of named colorants must be in the range of 1 to 15.

There is no default.

Exception Conditions:

EC-501505 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-501506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-50150E Missing Required Tag

The tag is required when the Color Palette Named Colorants tag and/or the Colorant Identification List tag is specified, but is missing.

EC-50150F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-501510 Invalid Value

The specified number of named colorants is zero or greater than 15.

Color Palette Gray

Tag ID X’5020’

Field Type X’05’ (BYTE)

Count 9 * the number of color entries

The tag translates 2-byte indexed color values to the monochrome color space. It consists of 9-byte repeating groups in the following format. Each repeating group maps an indexed color value to a gray value. Repeating groups must be sorted in ascending order of indexed color value.

Table 44. Color Palette Gray Tag Syntax

Offset	Type	Name	Range	Meaning
0–1	2-byte UBIN	IndexedColorValue	X'0100' – X'FFFF'	2-byte indexed color value specified in data stream
2–7	2-byte UBIN	CIELABValue	X'0000' – X'FFFF'	L* component a* component b* component
8	1-byte UBIN	Component_1	X'00'–X'FF'	Intensity of gray

All values must be specified. The CIELABValue is defined as a 2-byte CIELAB value with the D50 illuminant. The CIELAB value from byte 2–7 is used for substitution if gray is not the output space of the device.

Note: The actual CIELAB ranges are: L* component : 0.0 to 100.0, a* and b* components: -128.0 to +127.0. All these ranges need to be mapped to X'0000'–X'FFFF'.

There is no default.

Exception Conditions:

EC-502005 Invalid Count Value

The specified Count field value is invalid for the tag. It is less than 9 or it is not a multiple of 9.

EC-502006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-50200F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-502010 Invalid Value

The IndexedColorValue is not valid or the offset caused some portion of the tag data to be outside of the CMRdata.

| **EC-502012** Incorrect order of repeating groups

| Repeating groups are not sorted in ascending order of
| IndexedColorValue.

Color Palette CMYK

Tag ID X’5025’

Field Type X’05’ (BYTE)

Count 12 * the number of color entries

The tag translates 2-byte indexed color values to the CMYK color space. It consists of 12-byte repeating groups in the following format. Each repeating group maps an indexed color value to a CMYK value. Repeating groups must be sorted in ascending order of indexed color value.

Table 45. Color Palette CMYK Tag Syntax

Offset	Type	Name	Range	Meaning
0–1	2-byte UBIN	IndexedColorValue	X'0100' – X'FFFF'	2-byte indexed color value specified in data stream
2–7	2-byte UBIN	CIELABValue	X'0000' – X'FFFF'	L* component a* component b* component
8	1-byte UBIN	Component_1	X'00'–X'FF'	Intensity of cyan
9	1-byte UBIN	Component_2	X'00'–X'FF'	Intensity of magenta
10	1-byte UBIN	Component_3	X'00'–X'FF'	Intensity of yellow
11	1-byte UBIN	Component_4	X'00'–X'FF'	Intensity of black

All values must be specified. The CIELABValue is defined as a 2-byte CIELAB value with the D50 illuminant. The CIELAB value from byte 2–7 is used for substitution if CMYK is not the output space of the device.

Note: The actual CIELAB ranges are: L* component : 0.0 to 100.0, a* and b* components: -128.0 to +127.0. All these ranges need to be mapped to X'0000'–X'FFFF'.

There is no default.

Exception Conditions:

EC-502505 Invalid Count Value

The specified Count field value is invalid for the tag. It is less than 12 or it is not a multiple of 12.

EC-502506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-50250F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-502510 Invalid Value

The IndexedColorValue is not valid or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-502512	Incorrect order of repeating groups Repeating groups are not sorted in ascending order of IndexedColorValue.

Color Palette RGB

Tag ID X'5030'

Field Type X'05' (BYTE)

Count 11 * the number of color entries

The tag translates 2-byte indexed color values to the RGB color space. It consists of 11-byte repeating groups in the following format. Each repeating group maps an indexed color value to a RGB value. Repeating groups must be sorted in ascending order of indexed color value.

Table 46. Color Palette RGB Tag Syntax

Offset	Type	Name	Range	Meaning
0–1	2-byte UBIN	IndexedColorValue	X'0100' – X'FFFF'	2-byte indexed color value specified in data stream
2–7	2-byte UBIN	CIELABValue	X'0000' – X'FFFF'	L* component a* component b* component
8	1-byte UBIN	Component_1	X'00'–X'FF'	Intensity of red
9	1-byte UBIN	Component_2	X'00'–X'FF'	Intensity of green
10	1-byte UBIN	Component_3	X'00'–X'FF'	Intensity of blue

All values must be specified. The CIELABValue is defined as a 2-byte CIELAB value with the D50 illuminant. The CIELAB value from byte 2–7 is used for substitution if RGB is not the output space of the device.

Note: The actual CIELAB ranges are: L* component : 0.0 to 100.0, a* and b* components: -128.0 to +127.0. All these ranges need to be mapped to X'0000'–X'FFFF'.

There is no default.

Exception Conditions:

EC-503005 Invalid Count Value

The specified Count field value is invalid for the tag. It is less than 11 or it is not a multiple of 11.

EC-503006 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-50300F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-503010 Invalid Value

The IndexedColorValue is not valid or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-503012 Incorrect order of repeating groups

Repeating groups are not sorted in ascending order of IndexedColorValue.

Color Palette CIELAB

Tag ID X’5035’

Field Type X’05’ (BYTE)

Count 8 * the number of color entries

The tag translates 2-byte indexed color values to D50 CIELAB color space. The precision of the CIELAB values is 2-byte. This tag consists of 8-byte repeating groups in the following format. Each repeating group maps an indexed color value to a CIELAB value. Repeating groups must be sorted in ascending order of indexed color value.

Table 47. Color Palette CIELAB Tag Syntax

Offset	Type	Name	Range	Meaning
0–1	2-byte UBIN	IndexedColorValue	X'0100' – X'FFFF'	2-byte indexed color value specified in data stream
2–7	2-byte UBIN	CIELABValue	X'0000' – X'FFFF'	L* component a* component b* component

All values must be specified. The CIELABValue is defined as a 2-byte CIELAB value with the D50 illuminant.

Note: The actual CIELAB ranges are: L* component : 0.0 to 100.0, a* and b* components: -128.0 to +127.0. All these ranges need to be mapped to X'0000'–X'FFFF'.

There is no default.

Exception Conditions:

EC-503505 Invalid Count Value

The specified Count field value is invalid for the tag. It is less than 8 or it is not a multiple of 8.

EC-503506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-50350F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-503510 Invalid Value

The IndexedColorValue is not valid or the offset caused some portion of the tag data to be outside of the CMRdata.

| **EC-503512** Incorrect order of repeating groups

| Repeating groups are not sorted in ascending order of
| IndexedColorValue.

Color Palette Named Colorants

Tag ID X’5040’

Field Type X’05’ (BYTE)

Count (number of named colorants+8) * (the number of color entries)

The tag translates 2-byte indexed color values to the named colorants color space. It consists of (n+8)-byte repeating groups in the following format, where n is the number of named colorants specified in the Number of Named Colorants tag. Each repeating group maps an indexed color value to a mixture of n named colorants. Repeating groups must be sorted in ascending order of indexed color value. Each field from byte 8 to 7+n corresponds directly and in the same order to a ColorantName specified in the Colorant Identification List tag, that is, byte 8 corresponds to the first ColorantName, byte 9 corresponds to the second ColorantName, and so on.

Table 48. Color Palette Named Colorants Tag Syntax

Offset	Type	Name	Range	Meaning
0–1	2-byte UBIN	IndexedColorValue	X'0100' – X'FFFF'	2-byte indexed color value specified in data stream
2–7	2-byte UBIN	CIELABValue	X'0000' – X'FFFF'	L* component a* component b* component
8	1-byte UBIN	Component_1	X'00'–X'FF'	Intensity of the first colorant
9	1-byte UBIN	Component_2	X'00'–X'FF'	Intensity of the second colorant
...				
7+n	1-byte UBIN	Component_n	X'00'–X'FF'	Intensity of the nth colorant

All values must be specified. The CIELABValue is defined as a 2-byte CIELAB value with the D50 illuminant. The CIELAB value from byte 2–7 is used for substitution if all the colorants required for this index are not available in the device. Note that, if the intensity of a certain component is X'00' for a particular IndexedColorValue, then that colorant is not required for that index.

Note: The actual CIELAB ranges are: L* component : 0.0 to 100.0, a* and b* components: -128.0 to +127.0. All these ranges need to be mapped to X'0000'–X'FFFF'.

There is no default.

Exception Conditions:

EC-504005 Invalid Count Value

The specified Count field value is invalid for the tag. It is less than (number of named colorants+8) or it is not a multiple of (number of named colorants+8).

EC-504006 Invalid Field Type

The specified Field Type is invalid for the tag.

Color Palette Named Colorants tag

EC-50400E Missing Required Tag

At least one Color Palette Tag is required but none were specified.

EC-50400F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-504010 Invalid Value

The IndexedColorValue is not valid or the offset caused some portion of the tag data to be outside of the CMRdata.

EC-504012 Incorrect order of repeating groups

Repeating groups are not sorted in ascending order of IndexedColorValue.

Colorant Identification List

Tag ID X’5045’

Field Type X’05’ (BYTE)

Count sum of the length over the number of named colorants

The tag specifies colorant names in free format UTF-16BE. The colorant name is used to identify the required colorant, for example: Pantone xyz. Each colorant name is from 1–125 characters (2–250 bytes) long. Names starting with @SPECIAL-COLORANT@ are reserved for special use such as “mark color”. The tag consists of n repeating groups for the named colorants in the following format, where n is the number of named colorants specified in the Number of Named Colorants tag.

Table 49. The Colorant Identification List Tag Syntax

Offset	Type	Name	Range	Meaning
0	1-byte UBIN	Length	X'03'–X'FB'	length of this repeating group, including length field
1-end	UTF-16	Colorant Name		Colorant name in free format UTF-16BE

There is no default.

Exception Conditions:

EC-504505 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-504506 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-50450E Missing Required Tag

The tag is required when the Color Palette Named Colorants tag is specified, but is missing.

EC-50450F Invalid Sequence

The tag has been encountered out of sequence or more than once.

EC-504510 Invalid Value

The Length in one of the repeating groups is not valid, the number of repeating groups specified does not match the Number of Named Colorants tag value, or the offset caused some portion of the tag data to be outside of the CMRdata.

| **EC-504513** Duplicate value

| A ColorantName appears in more than one repeating group.

End Data

Tag ID X'FFFF'

Field Type X'05' (BYTE)

Count 0

This tag signifies the end of the tag list. This tag must be present for every CMR type, or exception EC-FFFF0F exists.

Exception Conditions:

EC-FFFF05 Invalid Count Value

The specified Count field value is invalid for the tag.

EC-FFFF06 Invalid Field Type

The specified Field Type is invalid for the tag.

EC-FFFF0E Missing Required Tag

The tag is required for the resource, but is missing.

Chapter 6. CMR Processing

Overview of CMR Processing

CMRs are used to describe a process that:

- Takes presentation data specified in an input color space
- Converts it to the output color space of the presentation device
- Modifies the colors to create the desired output for a particular device
- Halftones the output data

The actual input and output device color spaces constrain which CMRs are applicable. There can be multiple CMRs that are invoked, but only some of them are usable for given data.

Further, CMRs can be invoked at multiple levels of the AFP document hierarchy and it is possible to have more than one CMR that is applicable for a given task at one time. For instance, there can be two audit color conversions from RGB to CIELAB; one is SMPTE-C and the other is sRGB. One can be invoked at the object level and the other at the page level. IPDS hierarchical rules and last-received-wins (when multiple conflicting CMRs exist at the same level) are used to resolve conflicts.

Media matching also affects the hierarchical search. If the media attributes specified in the CMR header do not match the media currently in use by the device, the hierarchical search may continue, looking for a CMR that better matches the media. This is described in "Matching Media Type of CMR with Media Type of Device" on page 117.

Note that there might be multiple CMRs of a given type invoked at one particular level. For instance, there could be two audit Color Conversion CMRs attached to a GOCA object, one for CMYK input data and the other for RGB data. Colors within the GOCA object might be specified using both color spaces and the appropriate CMR would be used each time.

Taken as a whole, the CMR system can seem complex. But a typical situation will be simple. Some complicated explanations will be included later to clearly define what must be done in complex situations, but they will rarely be encountered.

When a CMR is needed, the device searches the hierarchy for an applicable CMR that applies to the current color space. The AFP architecture hierarchy for CMRs is as follows:

1. CMR invoked with an object. Note that CMRs attached to an object received in home state are ignored and that CMRs can be attached to an object when it is included using an IDO. (See "Color Conversion Profiles within TIFF, JFIF and GIF" on page 118 for a discussion of profiles embedded within the object.)
2. CMR invoked with a page or overlay
3. CMR invoked in home state
4. Device default CMR

If two applicable CMRs that both apply to the current color space are invoked at the same level, the last one invoked is used. If no applicable CMR is explicitly invoked, a device default is used.

For color conversions, Link CMRs are normally used to improve throughput. The following discussion assumes that there are no active Link CMRs. The next section, “Link Color Conversion CMRs” on page 105, discusses Link Color Conversion CMRs.

1. Presentation data specifies an input color space. Knowing that color space, a search is done of the invoked audit Tone Transfer Curve CMRs to find the first one that has the same number of components. If one is not found, the identity tone transfer curve (that is, the printer default) is used.
2. Next, knowing the input color space, a search is done of the invoked audit Color Conversion CMRs to find the first one that has that input color space. This is done by examining the Color Space Signature field within the ICC profile header. In cases where the name of the input color space is unknown, the number of components in the input data will be used to select a Color Conversion CMR.
3. A search of the invoked instruction Color Conversion CMRs is done to find the first one with the required output color space. In most cases, this will be a device default CMR. Note that the audit and instruction PCS's do not need to be the same. The device has the ability to convert between CIELAB and CIEXYZ, the available PCSs.
4. A search is done of the invoked instruction Tone Transfer Curve CMRs to find one with the correct number of components.
5. A search is done of the invoked Halftone CMRs to find one with the right number of components.
6. The colored data is converted and halftoned using these CMRs. Note that it is possible to combine some of the operations to improve performance.

"Applicable", "Selected" CMRs

CMRs must be **applicable** in order to be used. If a CMR is not applicable, it may be ignored and no NACK is issued. The following examples explain the meaning of **applicable**.

- An instruction Tone Transfer Curve CMR with three components is not applicable if the device is monochrome.
- An instruction Halftone CMR with three components is not applicable on a CMYK device.
- An audit Color Conversion CMR that converts from a three-component space is not applicable if the input image has four components.
- An output Color Conversion CMR that converts into a four-component color space is not applicable if the device is a CMYK printer that is running in a monochrome-appearance mode.

In order to **select** a CMR, the hierarchy is searched as discussed above, looking for an **applicable** CMR. The first applicable CMR found in the hierarchy is selected and used.

Link Color Conversion CMRs

Link Color Conversion CMRs provide an efficient method for converting directly from the input color space to the output color space. This is useful for optimizing performance. The next section, "Link Color Conversion CMRs based on audit/instruction Color Conversion CMRs" on page 105, discusses Link Color Conversion CMRs that link audit and instruction CMRs. The section after that, "Link Color Conversions: ICC DeviceLink Subset" on page 106, shows how ICC DeviceLink CMRs are used in the hierarchy search.

Link Color Conversion CMRs based on audit/instruction Color Conversion CMRs

Link Color Conversion CMRs with subset types of X'01' (LinkColorConversion LUT) and X'02' (LinkColorConversion Identity) are architected to require two special tags:
- Audit CMR identifier: A Link Audit CMR OID Tag
- Instruction CMR Identifier: A Link Instruction CMR OID Tag

When Color Conversion CMRs are sent to the device, an OID must be attached. The OID uniquely identifies the CMR and is used to connect the audit and instruction Color Conversion CMRs with the matching Link CMR. The OID was computed from the CMR contents, using an architected algorithm that includes an MD-5 checksum. The OID format is described in the MO:DCA Registry Appendix of the *Mixed Object Document Content Architecture Reference*.

The audit and instruction CMRs are identified as described above. Then a search is done of Link Color Conversion CMRs to find a link CMR that combines the audit and instruction CMRs. This is done by comparing the CMR OIDs with those specified in the link CMR. Note that any activated link CMR can be used. It does not need to be invoked in order to be eligible for use.

Note, further, that an audit CMR must be identified to use a downloaded link CMR. If the audit-type information from within an object (for example, a TIFF image) is used, there is no way to identify the link that can be substituted. So if it is desirable to use a link CMR, an audit CMR must be attached to the TIFF object. The audit CMR takes precedence over the ICC profile specified within the TIFF.

Link Color Conversion CMRs also require tags containing the CMR names of the audit and instruction CMRs. These two tags are required but are for informational purposes only and are not used while selecting the link CMR.

Link Color Conversions: ICC DeviceLink Subset

In some situations, it might be desirable to use a conversion that goes direct from input color space to output color space without using an intermediate conversion into the PCS. For instance, a direct conversion can be used to avoid losing information when conversions go in and out of the PCS. This might preserve information about how much black to use, or information about a spot color. The ICC DeviceLink subset of the Link CMR can be used in this situation. Typically, the ICC DeviceLink Profile would be created in some special customized or hand-crafted manner and be targeted at a particular device.

A Link CMR with subset type of ICC DeviceLink must be invoked in order to be used. Note that other Link CMRs do not need to be invoked. The ICC DeviceLink subset can be easily identified by checking the CMRType field in the CMR header: it will be "DL". Since an ICC DeviceLink can be invoked at various levels of the hierarchy, a hierarchical search must occur in order to select an ICC DeviceLink for use.

This special Link CMR has precedence over audit/instruction color conversions. In the following, "Look for an ICC DeviceLink" means that the input color space matches that of the data and the output space matches that of the device. If multiple device links are invoked at a given level of the hierarchy, the last one invoked will be selected.

The existence of an audit CC at some level has precedence over an ICC DeviceLink at a lower level. The search algorithm is shown here and the figure below diagrammatically shows how to search.

- Search at the object level
  - Look for an ICC DeviceLink. If found, stop and use it.
  - Else look for an audit CC CMR. If found, stop, find an instruction CC CMR by searching all levels. Use the selected pair.
- If an ICC Profile exists within the object (e.g. TIFF) use it and find an instruction CC CMR by searching all levels.
- Else search at the page/overlay level
  - Look for an ICC DeviceLink. If found, stop and use it.
  - Else look for an audit CC CMR. If found, stop, find an instruction CC CMR by searching all levels. Use the selected pair.
- Else search at higher levels
  - Look for an ICC DeviceLink. If found, stop and use it.
  - Else look for an audit CC CMR. If found, stop, find an instruction CC CMR by searching all levels. Use the selected pair.
- Else use the default audit CC CMR
  - Note that there is no default ICC DeviceLink CMR.
  - Find an instruction CC CMR by searching all levels. Use the selected pair.

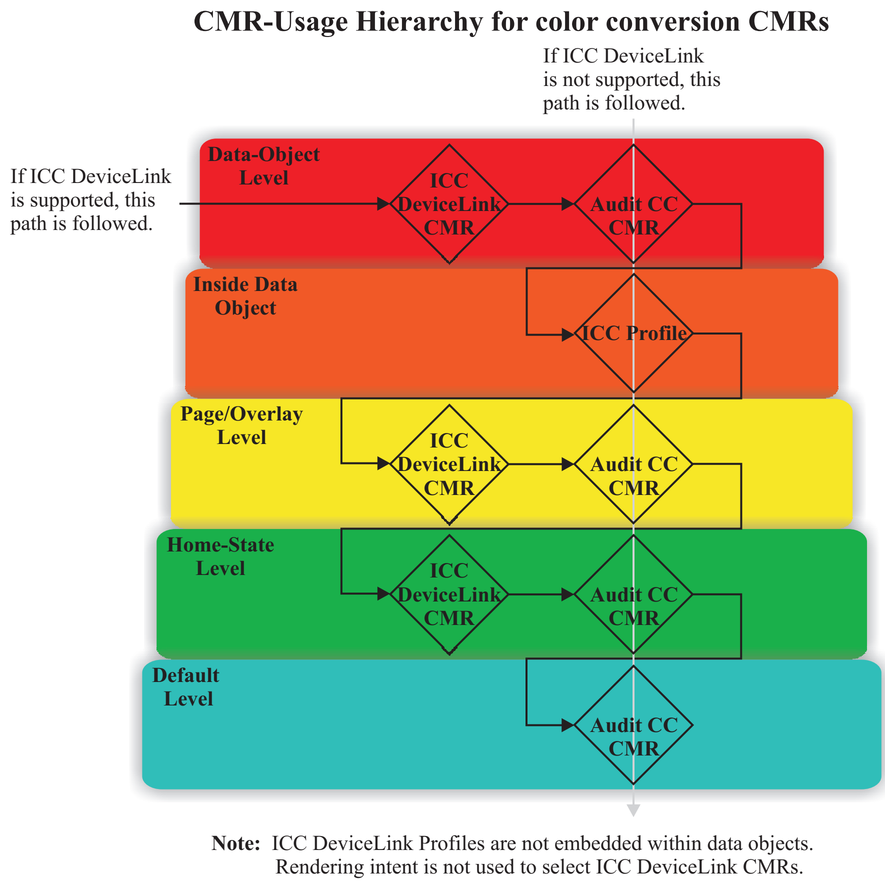


Figure 11. Selecting Appropriate Color-Conversion CMRs

Note that the CMR media attributes of an ICC DeviceLink must match the media attributes of the device. This means that the process defined in "Matching Media Type of CMR with Media Type of Device" on page 117 should be followed. If some of the attributes do not match and the hierarchy search continues, the hierarchy search will continue normally as if the CMR that does not match were never found.

Rendering Intent is not used when selecting an ICC DeviceLink for use. An ICC DeviceLink represents exactly one rendering intent that is specified in the header of the ICC Profile. If an ICC DeviceLink is selected for use during a hierarchical search, it is used regardless of whether the currently active rendering intent matches the rendering intent of the profile.

Audit/Instruction/Link Color Conversion CMRs

An audit Color Conversion CMR describes device-dependent colors or non-device colors (for example, sRGB) in the presentation data. They provide a way of converting from the input color space to a profile connection space (PCS).

An instruction Color Conversion CMR provides a way of converting from the PCS to the device output color space. Note that only certain instruction CMRs are applicable for a given device. For instance:

- If the device is a CMYK printer, only tone transfer curves with 4 components are applicable.
- Some devices support only bilevel halftone screens, not multilevel screens.
- If the device is an RGB display, it requires color conversions that convert into the RGB of the display.

The device may ignore any instruction CMRs that are not applicable, thus making the search path shorter.

Similarly, an ICC DeviceLink CMR must have an output color space that matches the color space of the device. If it does not match, the device may ignore it.

Table 50 shows which ICC profiles may be used for audit CMRs and which may be used for instruction CMRs. Table 13 on page 28 gives more information about the profiles.

Input type profiles describe colors coming from a scanner or digital camera and therefore they are not used as instruction CMRs.

Table 50. Profile Subsets in Audit & Instruction Color Conversion CMRs

Type	Audit	Instruction
Monochrome input profile	Yes	No
Monochrome display profile	Yes	Yes
Monochrome output profile	Yes	Yes
Three-component matrix-based input profile	Yes	No
Three-component matrix-based display profile	Yes	Yes
N-component LUT-based input profile	Yes	No
N-component LUT-based display profile	Yes	Yes
N-Component LUT-based output profiles	Yes	Yes
ColorSpace conversion profile	Yes	No

Creating Link Color Conversion CMRs – LinkColorConversion LUT subset

A link CMR describes how to convert directly from an input color space to an output device color space. It links an audit and an instruction CMR. It will define four color conversions, one for each of the possible rendering intents. Both the audit and the instruction CMR have rendering intents specified in their ICC header. The rendering intent in the ICC profile header of the instruction CMR becomes the Default Rendering Intent of the link CMR.

The link creation process combines the audit and instruction CMRs and creates four LUTs. Each LUT of the link CMR collapses all the steps of the two (audit and instruction) color conversions into a single multidimensional lookup table. It is possible that some of the LUTs will be identical if not enough information exists to create all versions separately. In this case, the offset for the LUTs could be the same.

To create each link LUT, the appropriate color conversion based on rendering intent must be selected from the ICC profiles for both the audit and the instruction Color Conversion CMRs. To create each one of the four link LUTs:

1. The rendering intent of the particular link LUT being created is identified.
2. The color conversion corresponding to that rendering intent is selected from the audit CMR.
3. The color conversion corresponding to that rendering intent is selected from the instruction CMR.
4. The selected audit and instruction color conversions are combined to produce the link LUT for that rendering intent.

The selection is based on the rendering intent of the link LUT currently being created and the goal is to use the color conversion from both the audit and the instruction CMR that is for that particular rendering intent.

When the color conversion rule for one of the rendering intents is not available, another color conversion must be used. The substitution methods are discussed below.

The basic rules for selecting a color conversion that corresponds to a given rendering intent for a given ICC profile are as follows:

1. If the appropriate AToBxTag (or BToAxTag) for that rendering intent exists, it is used. (AToB0Tag is used for perceptual, AToB1Tag is used for media-relative colorimetric, AToB2Tag is used for saturation.)
2. If that tag does not exist, the rendering intent in the header of this ICC profile is noted. The tag corresponding to this rendering intent is used.
3. If that tag does not exist, the implementation is device-dependent.

For audit CMRs, only the AToBxTags are considered, not the BToAxTags. If no AToBxTags exist:

- For monochrome profiles, the grayTRCTag is used for all rendering intents.
- For three-component matrix-based profiles, the matrix/TRC combinations are used for all rendering intents.

For instruction CMRs, only the BToAxTags are considered, not the AToBxTags. If no BToAxTags exist:

- For monochrome profiles, the inverse of the grayTRCTag is used for all rendering intents.
- For three-component matrix-based profiles, the inverse of the matrix/TRC transformation, as described in the ICC specification, is used for all rendering intents.

In order to create the link LUT for ICC-absolute colorimetric rendering intent, the following are combined:

- From the audit CMR, use the same tag(s) as were used when creating the link LUT for media-relative colorimetric. Apply a white point conversion based on the audit CMR's ICC mediaWhitePointTag.
- From the instruction CMR, use the same tag(s) as were used when creating the link LUT for media-relative colorimetric. Apply a white point conversion based on the instruction CMR's ICC mediaWhitePointTag.

Tone Transfer Curve and Printer Calibration

Processing of color information involves:

1. Converting color into the device color space (CMYK on normal printers)
2. Modifying the color for each plane
3. Halftoning each plane

The second step, modifying the color for each plane, is typically a one-dimensional conversion and is represented as a 1-D array called a curve. The curve would actually be a set of curves, one for each of the color planes in the color space. This document will use the term "curve" to represent the whole set of curves.

Actually, there can be multiple curves, each performing a different function. The curves are used sequentially although, in practice, they might be concatenated to form one curve for improved performance. The effect of each curve is a "delta" to the previous curve.

There are two basic uses of these modifying curves:

- There must be some way to put the device into a well-known state and maintain it in that state. This well-known state should be the state it was in when the default instruction Color Conversion and Halftone CMRs were created.
- The user might want to control the look-and-feel of the output.

Each device may handle these processes in different ways, but the following describes one way of dealing with this complexity. Note that many devices will not have all these options available or will describe them with different terms. There are four curves to be applied sequentially in this example:

Tone Transfer Curve (TTC) – This curve is contained in a CMR and can be specified by the user to modify the behavior of a device so that a desired relationship between input and output is achieved. This could include ink limiting, linearization, lightness, contrast, the relationship between the highlights, midtones and shadow regions or even reverse-video. Only one Tone Transfer Curve is used during the processing of a color object. The Tone Transfer Curve that is used can vary with each color object. It is selected using the rules of the CMOCA hierarchy. If no Tone Transfer Curve is specified, the printer default (the identity) will be used.

Operator Requested Curve – This curve allows the user the same control of the look-and-feel of the output as the Tone Transfer Curve. However, this curve is controlled by input from the printer console and is not specified in the data stream. The Operator Requested Curve will be constant for a complete print job or larger boundary.

Tone Reproduction Curve – This curve is used to put the printer into a known state. It compensates for dot gain, printer characteristics, typical humidity, paper characteristics, ink/toner characteristics, speed, etc. There might be different Tone Reproduction Curves in one printer for printing on different sides, different engines or different media. The assumption is that, after applying the Tone Reproduction Curve, the device is in an optimal state.

Calibration Curve – This curve is used to modify the behavior of a device, returning it to a known state. The assumption is that this known state is the optimal state. The Calibration Curve might be something controlled by the operator or might be automatically controlled within the printer. Changes to the Calibration Curve might need to occur relatively frequently due to changing ink/toner characteristics and changing humidity. The Calibration Curve might be different for each Tone Reproduction Curve in the printer.

The Tone Transfer Curve (TTC) and the Operator Requested Curve perform essentially the same function, but the first is transmitted in the data stream and the second is controlled via the printer's user interface. Normally, the two curves would be applied sequentially. However, in some cases, the device might want the Operator Requested Curve to override the TTC, effectively ignoring the TTC. If an applicable TTC is ignored, an error condition exists that is governed by exception ID X'025E..05'. This exception signifies that an "invoked, selected CMR was not used". The effect of this exception is controlled by the Color Fidelity Triplet and by Error Handling Control (EHC). The Color Fidelity Triplet or EHC can allow processing to continue by using a device default (identity) TTC or can force processing to stop. Reporting of the NACK is also controlled by the Color Fidelity Triplet or the EHC. Thus, by using the Color Fidelity Triplet or EHC, the user can control whether the Operator Requested Curve is allowed to override the TTC.

Note that the above discussion assumes that the printer calibration is done digitally, in software, before the color is halftoned. It is also possible to mechanically control the output after it is halftoned. For instance, it might be possible to regulate the amount of ink emitted by an inkjet. This control could be used instead of the Calibration Curve.

Use of Indexed CMRs

Indexed CMRs provide rules about how to render indexed colors. Indexed CMRs apply to indexed colors that are specified using the Highlight color space. They do not apply to indexed colors found within PostScript or other non-IPDS data objects.

For Indexed CMRs, both instruction and audit processing modes are valid. However, only Indexed CMRs with a processing mode of instruction will be used. Indexed CMRs that have an audit processing mode specified are ignored.

The tags in the Indexed CMR allow the CMR to use various color spaces in the descriptions. These color spaces can be grayscale (monochrome), named colorants, RGB, CMYK or CIELAB. A conversion from the index into CIELAB must always be provided. If a conversion into another color space is provided, it is used when applicable. For instance, if a conversion into CMYK is provided and the device is a CMYK printer, the conversion is used. The CMYK is assumed to be the device's CMYK and no color conversion CMRs are used. However, if a conversion into RGB is specified for that same CMYK printer, it is not applicable and the conversion into CIELAB will be used instead. In this case, the instruction Color Conversion selected from the hierarchy is used to convert the CIELAB into the output space of the device.

If the color palette is given in terms of named colorants, and some of the colorants required to produce a particular index are not available in the device, then the CIELAB palette information will be used instead of the named colorant information.

Indexed colors will ultimately be rendered in one of two color spaces:

1. The output color space of the device (typically CMYK for printers and RGB for displays)

2. A named colorant space when spot colors are available in printer.

In the first case, the Tone Transfer Curve CMR and the Halftone CMR selected from the hierarchy for the output color space are used. In the second case, the number of component named-colorants is noted. The hierarchy is searched for a Tone Transfer Curve and a Halftone that have this same number of components and the CMRs that are found are used.

The Indexed CMR to be used is selected using the normal hierarchical rules. Media-matching rules also apply. When the Indexed CMR is selected, its palettes are searched for the index in question. If the index is not found, IPDS exception processing is performed. No attempt is made to look for the index in any other Indexed CMR.

If a highlight color index is specified in the data stream, but cannot be resolved by an Indexed CMR, IPDS exception ID X'020E..03' is used. This can occur in two situations:

- No host-invoked Indexed CMR is found in the hierarchy.
- The required index is not found in the Indexed CMR that was selected.

Allowed Processing Modes

There are three possible processing modes: audit, instruction and link. Only certain processing modes are allowed with each specific type of CMR. An exception occurs if the processing mode is not valid for the CMR type. The following table shows which processing modes are valid. In addition, the device should ignore (without causing an exception) certain types of CMRs. That is also shown in the table.

A CMR is **generic** if the CMRVersion in the CMR Header is “generic”. Only Tone Transfer Curve CMRs and Halftone CMRs have registered generic versions, so if the type is anything else, an exception occurs.

Table 51. Allowed Processing Modes

CMR Type	Processing Mode					
	Non-Generic CMRs			Generic CMRs		
	Audit	Instruction	Link	Audit	Instruction	Link
Color Conversion	valid	valid	invalid - error	invalid - error		
Tone Transfer Curve	valid	valid	invalid - error	valid - ignored[1]	valid	invalid - error
Halftone	valid - ignored	valid	invalid - error	valid - ignored[1]	valid	invalid - error
Link Color Conversion	invalid - error	invalid - error	valid	invalid - error		
Indexed	valid - ignored	valid	invalid - error	invalid - error		

Notes:

1. Generic Tone Transfer Curves and generic Halftones are ignored because, in order to replace a generic CMR with a specific CMR, you must know the targeted device. This is unknown for an audit CMR.

A CMR is passthrough if the CMRVersion in the CMR Header is “pasthru”. Only Color Conversions can be passthrough so, if the CMR type is anything else, an exception occurs. The mode of a passthrough CC CMR must be audit. The CMR is ignored if the mode is specified as instruction. If the mode is link, an exception is generated since that is not valid for any Color Conversion CMRs.

Device Default CMRs

Every device must supply both audit and instruction default CMRs.

Default Instruction CMRs

The required default instruction CMRs are used to process colors in the color space of the device. If the device has multiple appearances - for instance, a printer that can function as both a full-color printer and a monochrome printer - then defaults for both identities must be available in the device.

For optimum quality, the device should have default instruction CMRs that are tuned for each type of media that it supports. For instance, a printer that officially supports three different paper types would have three different Color Conversion instruction CMRs available. It is possible that some media have different characteristics on the two sides. In this case, the device would have a default Color Conversion CMR for each side.

In some cases, a printer controller might be ripping pages that are sent to one of several print engines. It might be doing load-balancing among the engines. In this case, the printer controller would have a default Color Conversion CMR for each engine, and possibly for each media type on each engine.

The default Halftone and Tone Transfer Curve CMRs might also differ depending on the media or engine.

The device is responsible for supplying the following device default **instruction** CMRs:

- An instruction Halftone CMR that takes the device color space as input.
- An instruction Tone Transfer Curve CMR.
- An instruction Color Conversion CMR from an ICC Profile Connection Space (CIEXYZ or CIELAB) to the device color space. The device must have the ability to convert between CIEXYZ and CIELAB or else it must supply color conversions into the device output space from both. This CMR should have profiles for all rendering intents.
- There are no default Indexed CMRs.

The default instruction Color Conversion CMRs can be used when constructing Link Color Conversion CMRs. Therefore, the device manufacturer “publishes” these CMRs. OIDs for these CMRs are created by both the device and the host system wishing to create link CMRs using them. The OIDs are the same in both situations. Thus, the Link CMR successfully matches the device's default CMR.

Note that printer default CMRs are used only when no other applicable CMR is invoked in the hierarchy. In some cases it might be desirable to ensure that the device uses its defaults since it can have knowledge that the application does not have. For instance, if the sides of the paper have different characteristics, then the device might have different default Color Conversion CMRs for each side. The device will know which side it is printing on and should select the appropriate color conversion.

To force the printer to choose an applicable default instruction CMR, no applicable CMR of that type should be invoked within the IPDS hierarchy. The printer may not use its default to override an applicable CMR that is invoked at any level of the hierarchy.

Default Audit CMRs

Audit CMRs are not dependent on the output device or the media, so only one default in each category is required.

The device is responsible for supplying the following device default audit CMRs:

- An audit Color Conversion CMR from input RGB to CIEXYZ or CIELAB.
  - A display may assume that the RGB is its device RGB
- An audit Color Conversion CMR from input CMYK to CIEXYZ or CIELAB.
  - A full-color printer may assume that the CMYK is its device CMYK
- An audit Color Conversion CMR from input grayscale to CIEXYZ or CIELAB.
- An audit Color Conversion CMR from YCrCb or YCbCr to CIEXYZ or CIELAB.
- Audit Tone Transfer Curve CMRs that apply for varying number of components.

If any other color spaces such as luv, hsv, hls, Yxy or cmy are used within the data stream, the application is responsible for providing a color conversion CMR. The device may generate an exception if an applicable Color Conversion CMR is not supplied.

The following audit CMR defaults have been architected:

- RGB:
  - In an AFP printer, this is SMPTE-C RGB
  - In a display, this represents its device RGB.
- CMYK:
  - In an AFP full-color printer, this represents the CMYK of the presentation device.
  - In an AFP monochrome printer, this is SWOP CMYK.
  - In an AFP full-color printer working in grayscale mode, the CMYK represents the CMYK of the device when it is in full-color mode.
  - In a display, this is SWOP CMYK.
- Grayscale:
  - The Grayscale default applies when the color space has one component and more than 1 bit per pixel. This includes color spaces that are specified as YCbCr where only the Y component is present.
  - In an AFP full-color printer, C = M = Y = 0; K = 1-gray_value
  - In an AFP monochrome printer, the grayscale is the device's grayscale
  - In a display, R = G = B = gray_value
- YCbCr or YCrCb:
  - The default YCbCr is based on CCIR Recommendation 601-1 but components are normalized so as to occupy the full 256 levels of an 8-bit encoding. This version of YCbCr is used by TIFF and JFIF as their default. The equations below are consistent with the TIFF[1] and JFIF[2] specifications. YCbCr is first converted to RGB using the procedure below and the resulting RGB is then converted to the profile connection space. The audit Color Conversion CMR selected from the hierarchy is used to convert the RGB into the Profile Connection Space.

1. TIFF 6.0 Specification Section 21: YCbCr Images

2. JPEG File Interchange Format V1.02 (Sept. 1992)

- Y, Cb and Cr are in the range 0 to 255 as they would be in a digital image. Cr and Cb are shifted from the range 0 to 255 of a digital image into the range -128 to 127 in the equations below by subtracting 128.
- The procedure for converting a YCbCr value (256 levels) to a gamma-corrected RGB value is:

```
R = Y                      + 1.402   (Cr – 128)
G = Y - 0.34414 (Cb – 128) - 0.71414 (Cr – 128)
B = Y + 1.772   (Cb – 128)
```

R,G,B values should be clipped to the range [0,255] or [0.0,1.0] depending on the number system being used.

- Note that the default white point for both CIELAB and CIEXYZ is D50.
- Halftone: no architected default
- Tone Transfer Curves: identity
- Indexed: no architected default

Default CMRs to replace Generic CMRs

Instruction Halftone CMRs and instruction Tone Transfer Curve CMRs can be generic. Generic CMRs must be replaced by device-specific CMRs. The device is required to have its own device-specific versions of all the generic CMR's that are registered in the Color Management Object Content Architecture. (see Appendix B, “Generic CMR Name Registry,” on page 127). If the device does not recognize a generic CMR name, it NACKs using exception ID X'025E..04'.

Passthrough Audit Color Conversion CMRs

An audit Color Conversion CMR can specify the version to be “pasthru”. Passthrough CMRs are defined only for Color Conversion CMRs. Prop4, the color space for a CC CMR, should be specified. When this passthrough CMR is invoked and Prop4 is the same as the color space of the device, then the color values will be passed through without color conversion. If Prop4 is not the same as the device color space, or not specified, then the passthrough CC CMR is ignored.

The CMRVersion in the CMR header indicates whether a CMR is passthrough. Prop4 in the header indicates the color space. Other Prop fields are unspecified. A passthrough Color Conversion is valid only as an audit CMR. It is ignored if its mode is instruction. An error is returned if the mode is “Link”.

A passthrough CC CMR is treated like other audit CC CMRs in terms of selecting an audit CC CMR from the hierarchy. A passthrough CC CMR has no data. There is no device-specific CMR that can be substituted for the passthrough CC CMR. It merely instructs the device to avoid doing any color conversion.

Matching Media Type of CMR with Media Type of Device

In some cases, it is important to know if a CMR is **media-specific**. It is media-specific if the media is completely specified. It is not media-specific in the following cases:

- If the target device is a display screen and the MediaBrightness is not specified or is not a valid entry. (Z is a capitalized letter; x and y are digits.)
- If the target device is a printer and one or more of the four media fields in the header are not specified or are invalid.

When an instruction CMR is needed, the hierarchy is searched to find an applicable CMR. There are two possible outcomes:

- An applicable CMR is found in the hierarchy and is selected.
- No CMR is selected from the hierarchy and a default CMR must be used.

| Note that the following discussion also applies to Link CMRs with subset being
| ICC DeviceLink.

In the latter case, the device should attempt to select a default CMR whose media specification matches the media being used. If there is no default CMR found that exactly matches the media being used, the device selects the best existing default CMR.

In the former case, when an applicable CMR is selected from the hierarchy, then its media characteristics are examined to determine whether it should be used.

1. Assume the selected CMR is media specific.
   - If all the media field values in the CMR match the media values current in the device, the CMR is used. (For printers, all four media fields must match. For displays, only the MediaBrightness is considered.)
   - If one or more of the media fields do not match the current media, the device searches the hierarchy for a media-specific CMR that matches the current media. Multiple applicable CMRs might exist at each level of the hierarchy and are included in the search, and each level of the hierarchy is searched in the normal order, except for the device default level that is not part of the search. If no matching media-specific CMR is found, then IPDS exception handling rules for CMR exceptions (ID X'025E..03') should be followed.
2. If all of the media fields in the CMR are unspecified, then the CMR is used regardless of the media in the device. This provides the user with a means of ensuring the use of his chosen CMR.
3. If one or more of the media fields is invalid, then IPDS exception handling rules for CMR exceptions should be followed.
4. Assume some, but not all, of the media fields are specified and the specified fields are valid.
   - If all the media fields that are specified in the CMR match the media that is currently in the device, the device searches the hierarchy for a CMR whose media attributes are a better match with the current media. Multiple applicable CMRs might exist at each level of the hierarchy and are included in the search, and each level of the hierarchy is searched in the normal order, except for the device default level that is not part of the search. If a better matching CMR is not found, the original CMR is used.
   - If any of the media fields that are specified in the CMR do NOT match the device, then the device searches the hierarchy for a CMR whose media

attributes do match the current media. Multiple applicable CMRs might exist at each level of the hierarchy and are included in the search, and each level of the hierarchy is searched in the normal order, except for the device default level that is not part of the search. If no CMR is found whose attributes match the current media, then IPDS exception handling rules for CMR exceptions (ID X'025E..03') should be followed.

Treatment of Named and Highlight Colors

The AFP architecture supports OCA named colors such as Blue, Red, and Brown that are inherently device-dependent. As such, these named colors should not be used when an exact color is required. AFP architecture has recommended RGB values for each OCA named color. It is recommended that these RGB values be interpreted as SMPTE-C values and mapped to a device's output color space. Note that there is no architected method for associating an audit color conversion CMR with the named OCA colors via the CMR hierarchy.

It is recommended that OCA Black (X'0008'), presented on a CMYK output device, be rendered as C=M=Y=X'00' and K=X'FF'.

The AFP architecture also supports a Highlight color space. It can be used in two ways:

- The highlight color number specifies the spot color to use. The range is X'0000'–X'00FF'.
- The highlight color number is interpreted as an index into a palette. The range is X'0100'–X'FFFF'. The Indexed CMR describes which colorants are used to render that color.

When the highlight color is interpreted as a spot color, no mechanism for converting this highlight color to the device's output color space is provided by the Color Management Object Content Architecture. However, if an application wants to assign a particular mix of colors to some highlight color, it can use a Color Mapping Table.

The section "Use of Indexed CMRs" on page 111 describes how other CMRs are used in conjunction with an indexed color.

Color Conversion Profiles within TIFF, JFIF and GIF

Some presentation data objects contain internal color management information. ICC profiles can be embedded within TIFF, JFIF and GIF.

When presentation data objects contain internal color management information, the device will use internal audit-like color management information, if any, when no applicable audit CMR is invoked with the object. The object-level audit CMR has priority over internal color management information. However, the internal color management information has priority over any audit CMRs at the page, home state or device-default levels. So the hierarchy is:

1. Object-level audit CMR
2. ICC profile within the object
3. Page-level audit CMR
4. Homestate-level audit CMR
5. Printer default audit CMR

The same rules hold for tone transfer curves that can be found within the object.

All internal instruction-like color management information is ignored.

An ICC DeviceLink profile cannot be specified within the object, according to the ICC color profile specification. Link Color Conversion CMRs should be applied when processing TIFF, JFIF and GIF if the device reports support for Link CC CMRs (IPDS STM property pair X'E001').

In some cases, some color conversion controls can be specified within the TIFF, JFIF, GIF, EPS or PDF. For instance, a white point or a grayscale correction curve can be specified. Where these are part of the definition of the input color space, they are used prior to applying any other color conversion, regardless of whether that color conversion is specified within the object or within a CMR.

CMR Usage within EPS, PDF

Audit Color Conversions

PostScript supports the following input color spaces:

- Device-Independent Color Spaces
  - CIEBasedABC
  - CIEBasedA
  - CIEBasedDEF
  - CIEBasedDEFG
- Device-Dependent Color Spaces
  - DeviceGray
  - DeviceRGB
  - DeviceCMYK
- Special Color Spaces (They add features or properties to an underlying color space.)
  - Pattern
  - Indexed
  - Separation
  - DeviceN

Device-independent color spaces include a specification of how to convert from that input space to the CIEXYZ connection space. An Audit Color Conversion invoked at the object level should override device-independent color spaces that specify the same color space if a device reports that it can reliably apply CMRs to EPS/PDF (IPDS STM property pair X'E100'). *[Architecture note: It might not be absolutely clear if the color space specified in the CMR is the same as that being specified by the device-independent color since only the number of components is known about the PostScript color space. Other information in the data stream might make it clear whether the color spaces match.]*

Device-dependent color spaces are not clearly specified. There are default simple rules that can be used to convert from one to another. They sometimes do not produce optimal output quality. These color space definitions will be overridden by audit Color Conversion CMRs. However, PostScript LanguageLevel3 supports the UseCIEColor parameter. This parameter allows selected device-dependent color spaces to be systematically transformed into a device-independent CIE-based color space. If this parameter is used within the EPS/PDF object to define a particular color space, the audit Color Conversion CMR will NOT override the definition of that color space.

Special color spaces are not directly affected by audit Color Conversion CMRs. However, if a device-dependent color space is modified by a CMR as in the preceding paragraph, and the special color space uses that as the underlying color space, then the color of the special color space will be modified.

ICC profiles can be embedded in EPS/PDF. An audit Color Conversion CMR invoked at the object level should override the ICC profiles that specify the same color space if a device reports that it can reliably apply CMRs to EPS/PDF (IPDS STM property pair X'E100').

Instruction Color Conversions

The appropriate printer-default instruction Color Conversion CMR will be used for the default Color Rendering Dictionary (CRD).

- If a CRD is specified within the PostScript data stream, it will be used instead of the default CRD.
- If an instruction Color Conversion CMR is associated directly with the EPS/PDF object, it will override both the default CRD and any CRD specified within the EPS/PDF object.

Link Color Conversion CMRs

Link Color Conversion CMRs should be used when processing EPS/PDF if a device reports support for Link CC CMRs (IPDS STM property pair X'E001') and it reports that it can reliably apply CMRs to EPS/PDF (IPDS STM property pair X'E100').

Halftones

The appropriate printer-default instruction Halftone CMR will be used for the default EPS/PDF halftone.

- If a halftone is specified within the EPS/PDF object, it will be used instead of the default EPS/PDF halftone.
- If an instruction Halftone CMR is associated directly with the EPS/PDF object, it will override both the default halftone and any halftone specified within the EPS/PDF object.

Note that these rules will be applied only if the PostScript RIP runs in bilevel mode. When the RIP runs in multilevel mode, that means RIP output specifies 8-bit intensity values, any halftone operations in the PostScript data stream will be ignored.

Audit Halftone CMRs have no effect on EPS/PDF processing.

Tone Transfer Curves

The printer default instruction Tone Transfer Curve is the identity so it will have no effect on EPS/PDF.

- If a Transfer Function is specified within the EPS/PDF object, it will be used.
- If an instruction Tone Transfer Curve CMR is associated directly with the EPS/PDF object, it will override any Transfer Function specified within the EPS/PDF object.

Audit Tone Transfer Curve CMRs have no effect on EPS/PDF processing.

Caveat

Some devices cannot completely, reliably ensure that a selected CMR is actually applied. This is because some EPS/PDF objects can be created in such a way that it is not possible to override the parameters. Other devices can reliably, predictably apply the CMRs during EPS/PDF processing.

Property pair X'E100' in the IPDS STM reply indicates that CMRs can be reliably applied to all EPS/PDF objects. If a device cannot guarantee that CMRs will always be predictably applied, it does not report X'E100' in the STM reply. When using such a device, it is recommended that an application that uses CMRs with EPS/PDF objects should test the output to verify that the CMRs are applied as expected.

Implementation notes

The following are some suggestions of possible ways to apply CMRs to a EPS/PDF object. A PostScript preamble could be used to set up parameters for the PS RIP and/or post-processing work could be done.

In order to specify CMR color conversions for CMYK and RGB, UseCIEColor can be used. This requires a PS RIP that is level 3. Note that any UseCIEColor operations that come later in the PostScript data stream will override the operations in the preamble. This is correct implementation.

The settransfer or setcolortransfer operators can be used to implement the Tone Transfer Curve CMR.

PostScript Type 3 Halftone Dictionaries can be used to implement Halftone CMRs that are bilevel threshold arrays. Halftone CMRs that are multilevel or error-diffusion have no direct counterpart in standard PostScript Language. If it is necessary to implement these types of halftones, the PostScript RIP could output 8-bit color values and the product RIP could perform the screening as a post-processing activity. It might also be possible to redefine some of the standard PostScript dictionary entries or operators to enable halftoning within the PostScript RIP.

For both halftones and TTCs, the PostScript operator should be redefined at the end of the preamble so that no halftones or transfer functions specified within the PostScript data stream will override those specified in the preamble.

Different Encodings of CIELAB

CIELAB is defined as follows:
- L has a value between 0.0 and 100.0
- a has a value between -128.0 and 127.0
- b has a value between -128.0 and 127.0

The encoding for L consistently maps the range [0.0, 100.0] to [X'0000', X'FFFF'] or to [X'00', X'FF'] depending on the number of bytes in the representation. The only exception to this is ICC profile legacy 16-bit encoding that maps to [X'0000', X'FF00']. Consult the ICC profile specification for more information about this.

Different architectures convert the range for a and b into 1-byte or 2-byte values differently.

- CMRs encode a and b by mapping [-128.0, 127.0] into the range [X'00', X'FF'] or [X'0000', X'FFFF']. Thus, -128.0 is represented by X'0000'. The hex values are treated as unsigned integers. These values are used in the Indexed CMR and also when using a Link CMR that converts from an input space of CIELAB.
- AFP treats the hex values as signed integers. Further, it specifies that the mapping depends on the number of bits. If eight bits are used, the range -128 to +127 is mapped to the range X'80' to X'7F', i.e. -128 is represented by X'80', and +127 is represented by X'7F'. If fewer than 8 bits are used, treatment of the most negative binary endpoint for the a and b components is device-dependent. More than 8 bits are not allowed.
- The ICC profile specification specifies that [-128.0, +127.0] maps into the range [X'00', X'FF'] or [X'0000', X'FFFF'], that is the same encoding used by CMRs. In addition, there is a special case with 16-bit ICC encoding where ‘legacy’ encoding is used. Consult the ICC profile specification for more information about this.
- In TIFF, [-127.0, +127.0] maps into [X'80', X'7F']. Thus, TIFF is treating the hex values as signed integers, but not exactly the same as IPDS does.

Care must be taken when using values from the different architectures to ensure that they are converted to a common encoding.

Implications for Drivers

It is preferable for the host to suppress the downloading of CMRs that are not applicable to the device, but this is not required. For instance, instruction Halftone CMRs for a three-component color space are not useful on a CMYK printer.

Generic CMRs versus Device-Specific CMRs

Certain *instruction* CMRs can be generic. Generic CMRs are defined for only two CMR types: Halftone CMRs and Tone Transfer Curve CMRs. The generic CMR must always be replaced by a device-specific CMR by either the server or the device. Appendix B, “Generic CMR Name Registry,” on page 127 lists the registered generic CMR names for the Halftone CMR type and the Tone Transfer Curve CMR type, with brief descriptions of the intended appearance of each.

An example of a generic CMR is a Tone Transfer Curve CMR that specifies a “highlight-midtone” output appearance. When the reference to the generic CMR is processed at print time, the generic CMR is mapped to a specific highlight-midtone Tone Transfer Curve CMR for the target device. The print server then downloads and activates the device-specific CMR in place of the generic CMR.

In cases where the mapping from generic to device-specific was not done by the print server, the conversion must be done in the presentation device. The device is required to have device-specific CMRs for each of the registered generic CMRs.

The CMRVersion in the CMR header indicates whether a CMR is generic so that a device can recognize when a mapping must be done. Appendix B, “Generic CMR Name Registry,” on page 127 indicates which fields of the CMR header are used to represent the generic information needed to map from generic to specific. Other fields are unspecified.

If a generic CMR is received and the device is unable to map it to a device-specific CMR, an error condition exists (exception ID X'025E..04').

Partial Support of TTC & HT CMRs

In some cases, a device provides only partial support for certain types of CMRs. The device needs to declare support in the IPDS STM reply to enable the host to transmit the CMR to the device. The device might receive a CMR that is not supported and, if so, it should use the X'025E..05' exception (invoked, selected CMR was not used). This exception is controlled by Error Handling Control and by the Color Fidelity triplet. Some situations where a CMR is not used include:

1. A HT or TTC is supported at the job level but not supported at the page or object level. For instance, the HT and/or TTC is applied in hardware or in software after a complete sheet is created.
2. A device does not support all subsets of halftones. For instance, a laser printer might be able to use a threshold array but not an error diffusion halftone.
3. A limited number of HT or TTC are supported on one sheet. If a CMR is invoked after the maximum number is reached, exception X'025E..05' should be used.

Note that exception handling should not be used with generic CMRs in cases where partial support affects the selection of the device specific CMR that is used. The device is substituting as best it can and no exception handling is required.

Appendix A. Tag Registry

This table defines the CMR tags in the base level of the Color Management Object Content Architecture. Support for some CMR types is optional, see Appendix C, "Compliance with Color Management Object Content Architecture," on page 133. When a presentation device supports a given CMR type, it must support the tags used by that CMR type, as defined in this table.

Table 52. Tag Registry

TagIDs	Tag Name	CMR Type
X'0004'	"Comment" on page 41	Halftone Tone Transfer Curve Color Conversion Link Color Conversion Indexed
X'0008'	"Date and Time Stamp" on page 42	Halftone Tone Transfer Curve Color Conversion Link Color Conversion Indexed
X'0011'	"Number of Components" on page 44	Halftone Tone Transfer Curve
X'1011'	"Halftone Subset" on page 45	Halftone
X'1021'	"Array Width" on page 46	Halftone
X'1025'	"Array Height" on page 47	Halftone
X'1030'	"Max Image Value" on page 48	Halftone
X'1035'	"Number of Device Levels" on page 49	Halftone
X'1040'	"Offset Tiling" on page 50	Halftone
X'1045'	"Bilevel Point-Operation Screen Data" on page 51	Halftone
X'1050'	"Multilevel Point-Operation Screen Data" on page 52	Halftone
X'1055'	"Error Diffusion Filter" on page 53	Halftone
X'1060'	"Location of Current Pixel" on page 55	Halftone
X'1065'	"Raster Direction" on page 56	Halftone
X'1070'	"Boundary Condition" on page 58	Halftone
X'1075'	"Threshold Value" on page 60	Halftone
X'1080'	"Quantization Boundary Table" on page 61	Halftone
X'2004'	"Tone Transfer Curve Subset" on page 63	Tone Transfer Curve
X'2011'	"Tone Transfer Curve Length" on page 64	Tone Transfer Curve
X'2015'	"Tone Transfer Curve Data" on page 65	Tone Transfer Curve
X'2020'	"Inverse Tone Transfer Curve Data" on page 67	Tone Transfer Curve
X'3011'	"ICC Profile Subset" on page 69	Color Conversion
X'3015'	"ICC Profile Data" on page 70	Color Conversion

Table 52. Tag Registry (continued)

TagIDs	Tag Name	CMR Type
X'3025'	"ICC Profile Filename" on page 72	Color Conversion
X'4011'	"Link Color Conversion Subset" on page 73	Link Color Conversion
X'4015'	"Link Audit CMR OID" on page 74	Link Color Conversion
X'4020'	"Link Instruction CMR OID" on page 75	Link Color Conversion
X'4025'	"Link Audit CMR Name" on page 76	Link Color Conversion
X'4030'	"Link Instruction CMR Name" on page 77	Link Color Conversion
X'4035'	"Default Rendering Intent" on page 78	Link Color Conversion
X'4040'	"Link LUT Perceptual" on page 79	Link Color Conversion
X'4045'	"Link LUT Media-Relative Colorimetric" on page 82	Link Color Conversion
X'4050'	"Link LUT Saturation" on page 85	Link Color Conversion
X'4055'	"Link LUT ICC-Absolute Colorimetric" on page 88	Link Color Conversion
X'4090'	"Link CMRE Identifier" on page 90	Link Color Conversion
X'5011'	"Indexed Subset" on page 91	Indexed
X'5015'	"Number of Named Colorants" on page 92	Indexed
X'5020'	"Color Palette Gray" on page 93	Indexed
X'5025'	"Color Palette CMYK" on page 94	Indexed
X'5030'	"Color Palette RGB" on page 96	Indexed
X'5035'	"Color Palette CIELAB" on page 98	Indexed
X'5040'	"Color Palette Named Colorants" on page 99	Indexed
X'5045'	"Colorant Identification List" on page 101	Indexed
X'FFFF'	"End Data" on page 102	Halftone Tone Transfer Curve Color Conversion Link Color Conversion Indexed

Notes:

1. For an Indexed CMR, at least one of the Color Palette tags must be present.
2. Tags X'F000'–X'FFFE' are private tags.

Appendix B. Generic CMR Name Registry

Generic CMRs are allowed for instruction Halftone CMRs and instruction Tone Transfer Curve CMRs. This appendix defines the currently registered generic CMR names. All presentation devices that support downloaded Halftone and Tone Transfer Curve CMRs must support all names defined in this registry. The device must substitute a device-specific CMR that fits as best it can. The device will not always have an accurate match but should not NACK if it recognizes the name. If the device does not recognize the generic CMR name, it uses exception ID X'025E..04' to indicate that this name is not supported.

No device or media information may be included in generic names. Only the fields describing the generic property are filled in. Other fields are not specified. Note that, for improved readability, spaces are left between fields of the CMR name in the examples below.

The CMR names specified in Figure 12 and Figure 13 on page 128 are the only valid generic CMRs.

Registered Generic Halftone CMRs

The registered generic Halftone CMRs are shown in Figure 12. Halftone Property 3 (Line Screen Frequency) or Property 2 (Halftone Type) are used to describe the generic quality of the halftone.

```
Prop # ...............................................1......2.....3....4....5..........
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 71@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 85@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 106@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 120@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 141@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 150@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 170@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 190@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 202@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 300@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 600@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ sto@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ dsp@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ erd@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ f-d@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ jjn@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ stu@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ brk@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ sra@@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ s-a@@@ @@@@ @@@@ @@@@ @@@@@@@@
```

Figure 12. Generic Halftone CMRs

The first eleven generic Halftone CMRs are used for clustered-dot halftones and indicate the line screen frequency of the halftone. The next three generic Halftone CMRs are used to specify a halftone of type stochastic, dispersed or error-diffusion. The last six are specific error diffusion halftones

Registered Generic Tone Transfer Curve CMRs

Four generic Tone Transfer Curve (TTC) CMRs are registered. The generic TTC CMRs control the "look-and-feel" or appearance (Property 2) of the image that is output. The four appearances that are supported are:

- Dark
- Accutone
- Highlight Midtone
- Standard

The registered generic Tone Transfer Curve CMR names are shown in the figure below.

```
Prop # ............................................1......2.....3....4....5..........
@@@@@@@@ TC generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ dark@@ @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ accutn @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ hilmid @@@@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ standd @@@@ @@@@ @@@@ @@@@@@@@
```

Figure 13. Generic Tone Transfer Curve CMRs

These appearances were designed for black/white printers and allow the user to specify the general look of the output. When one of these generic TTC CMRs is targeted for a color printer, the expected result is not clearly defined and the default TTC curve (the identity) can be substituted.

Generic Tone Transfer Curve Appearance Definition

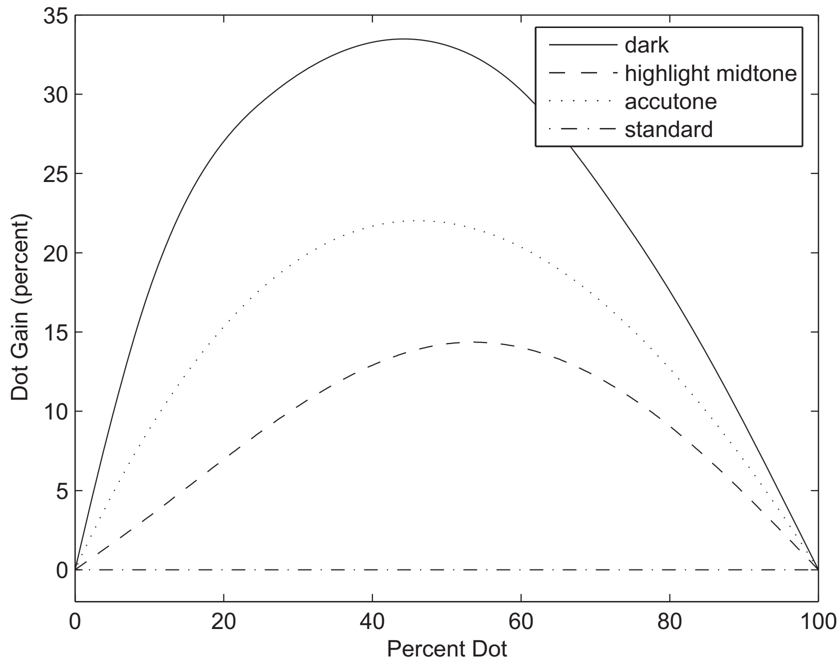
A Tone Transfer Curve (color calibration) alters the darkness of image data, accounting for the dot gain characteristics of the printer. A TTC might be used to allow one device to emulate the output of another device. For example, to emulate an offset press, it might be desired that a 50% tint should be printed as a 65% tint after calibration. To accomplish this, a Tone Transfer Curve compensates for the printer's dot gain so that when a 50% tint is specified in the image, a 65% (Murray-Davies apparent dot area) tint is printed.

The actual appearance depends on a combination of the printer model, the halftone screen, and the appearance selected (the TTC).

The dot gain curves for the final output appearance when using the generic TTC CMRs are shown in Figure 14 on page 129. They represent the effect of the combined TTC, the halftone and the dot gain from the printer. This means, for instance, that the Standard generic TTC is exactly undoing the dot gain of the printer. For a given generic TTC, each printer model must supply device-specific TTCs for each of the standard halftones that it supports.

Dot gain is frequently quoted as a single number, for example "15 percent dot gain". If the dot gain is specified without a corresponding percent dot where it is measured, a 50% dot is assumed. The appearances Dark, Accutone, Highlight Midtone, and Standard have dot gains of 33, 22, 14 and 0 percent respectively, measured at the 50% dot. Accutone approximates linear L* tone characteristics.

Figure 15 on page 129 shows the lightness as a function of percent dot for each of the generic Tone Transfer Curves. The lightness curve is the primary reference.



The dot gain represents what the output looks like in terms commonly understood. It is not necessarily an accurate representation.

Figure 14. Dot Gain as a Function of Percent Dot

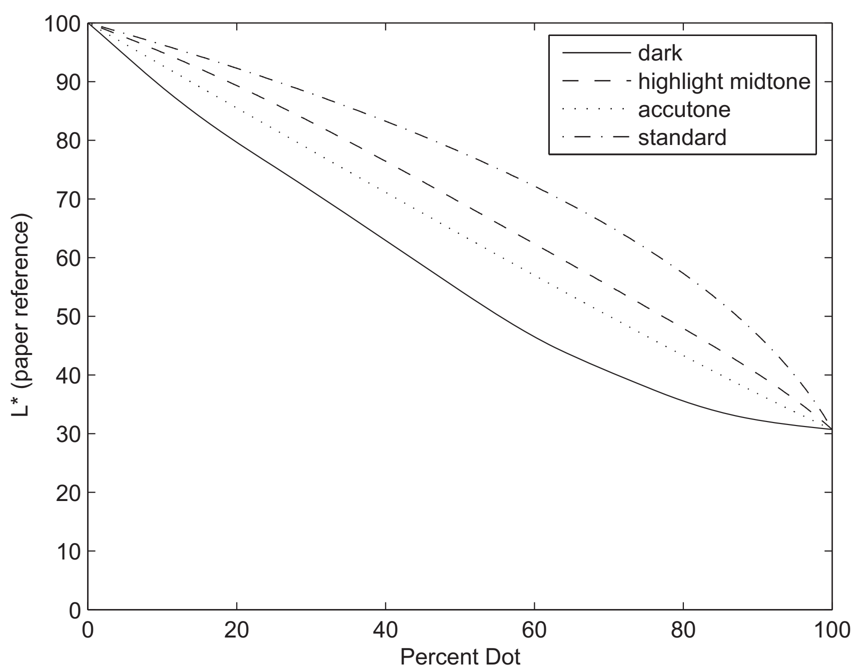


Figure 15. Lightness as a Function of Percent Dot

Using Generic Halftones and TTCs Together

The Tone Transfer Curve (TTC) and Halftone are interrelated. The TTC depends on the Halftone that is selected. So, when mapping generic to device-specific, the Halftone is mapped first and then the TTC mapping is done, taking into consideration the Halftone that is being used.

A device-specific TTC would specify both the Look-and-Feel and the Halftone's Characterization. Note that, in the following example, other @@@ fields would actually be filled in since this is a device-specific CMR.

```
Prop # ...........................................1......2.....3....4....5..........
@@@@@@@@ TC 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ accutn 141@ @@@@ @@@@ @@@@@@@@
```

Assume that a particular printer has the following 3 device-specific halftones available: [Other @@@ fields would actually be filled in since these CMRs are device specific.]

```
Prop # ...........................................1......2.....3....4....5..........
@@@@@@@@ HT 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 750@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 104@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ HT 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 141@ @@@@ @@@@ @@@@@@@@
```

Then it would need to have 12 device-specific TTCs available to cover all combinations. Note that some of the TTCs could be duplicates. Here are five of them.

```
Prop # ...........................................1......2.....3....4....5..........
@@@@@@@@ TC 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ hilmid 750@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ hilmid 104@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ hilmid 141@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ standd 750@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ standd 104@ @@@@ @@@@ @@@@@@@@
```

Now suppose the user selects a generic Halftone of 106 lpi and a generic TTC for Highlight Midtone by specifying the following:

```
Prop # ...........................................1......2.....3....4....5..........
@@@@@@@@ HT generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 106@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC generic @@@@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ hilmid @@@@ @@@@ @@@@ @@@@@@@@
```

So, given the printer assumed above, when the device-specific 104 lpi halftone is selected, the device-specific TTC for hilmid / 104 would be selected. These are the two device-specific CMRs that would be used:

```
Prop # ...........................................1......2.....3....4....5..........
@@@@@@@@ HT 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ @@@@@@ 104@ @@@@ @@@@ @@@@@@@@
@@@@@@@@ TC 001.200 IBM@@ @@@@@@ @@@ @@@ @@@ @@ @@@ @@@@@ hilmid 104@ @@@@ @@@@ @@@@@@@@
```

Now, that was simple because both the Halftone and TTC were generic. If the halftone is not one of the device-specific halftones that the printer has available for conversion from generic, then the printer might not have an exact device-specific match for the generic TTC. In this case, the device will select the device-specific TTC that matches as best it can.

Appendix C. Compliance with Color Management Object Content Architecture

In order to claim that the baseline Color Management Object Content Architecture is supported, a device is required to support certain CMRs as indicated in the following table.

Table 53. CMR Architecture Compliance Requirements

CMR Type	Required for Baseline Support?
Halftone	No
Tone Transfer Curve	No
Color Conversion	Yes
Link Color Conversion	No
Indexed	No

In order to claim that the baseline Color Management Object Content Architecture is supported, a device is required to support generic CMRs as follows, regardless of whether it reports support for these CMR types in the IPDS STM reply:

- It must support all of the Registered Generic Halftone CMRs shown in Figure 12 on page 127.
- It must support all of the Registered Generic Tone Transfer Curve CMRs shown in Figure 13 on page 128.

In order to claim that the baseline Color Management Object Content Architecture is supported, a device is required to supply device defaults as follows, regardless of what downloaded CMRs it supports:

- The device must supply the following default instruction CMRs:
  – An instruction Halftone CMR that takes the device color space as input.
  – An instruction Tone Transfer Curve CMR.
  – An instruction Color Conversion CMR from an ICC Profile Connection Space (CIEXYZ or CIELAB) to the device color space. The device must have the ability to convert between CIEXYZ and CIELAB or else it must supply color conversions into the device output space from both.
  – There is no default Indexed CMR.
- The device must supply the following default audit CMRs:
  – An audit Color Conversion CMR from input RGB to CIEXYZ or CIELAB.
    - A display may assume that the RGB is its device RGB.
  – An audit Color Conversion CMR from input CMYK to CIEXYZ or CIELAB.
    - A full-color printer may assume that the CMYK is its device CMYK.
  – An audit Color Conversion CMR from input grayscale to CIEXYZ or CIELAB.
    - A monochrome printer may assume that the grayscale is its device grayscale.
  – An audit Color Conversion CMR from YCrCb or YCbCr to CIEXYZ or CIELAB.
  – Audit Tone Transfer Curve CMRs for all possible number of components.

- Since the default audit Tone Transfer Curve is the identity, it may be implicitly implemented by doing nothing.
– No default audit Halftone CMR is required since audit Halftone CMRs are ignored.
– No default audit Indexed CMR is required since audit Indexed CMRs are ignored.

- A device must be able to handle an input color space that is specified as CIELAB or CIEXYZ (i.e. a profile connection space). No audit CMR is required for CIELAB or CIEXYZ since an instruction CMR can be used directly.

Notices

The AFP Consortium or consortium member companies might have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents.

The following statement does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: AFP Consortium PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. The AFP Consortium might make improvements and/or changes in the architecture described in this publication at any time without notice.

Any references in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this architecture and use of those Web sites is at your own risk.

The AFP Consortium may use or distribute any information you supply in any way it believes appropriate without incurring any obligation to you.

This information contains examples of data and reports used in daily business operations. To illustrate them in a complete manner, some examples include the names of individuals, companies, brands, or products. These names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, other countries, or both:

Advanced Function Presentation
AFP
AFPCC
AFP Color Consortium
Bar Code Object Content Architecture
BCOCA
Color Management Object Content Architecture
CMOCA
InfoPrint
Intelligent Printer Data Stream
IPDS
Mixed Object Document Content Architecture
MO:DCA

These terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

IBM

Adobe®, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

AFPC and AFP Consortium are trademarks of the AFP Consortium.

UP^3I is a trademark of UP^3I Limited.

Other company, product, or service names may be trademarks or service marks of others.

Glossary

This glossary contains terms that apply to the CMOCA Architecture and also terms that apply to other related presentation architectures.

If you do not find the term that you are looking for, please refer to the *IBM Dictionary of Computing*, document number ZC20-1699 or the *InfoPrint Dictionary of Printing*.

The following definitions are provided as supporting information only, and are not intended to be used as a substitute for the semantics described in the body of this reference.

A

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.

additive color. Additive colors are transmitted light used in video monitors and televisions. Red, green and blue light are referred as the Additive Primary Colors. When used in various degrees of intensity and variation, they create all other colors of light; when superimposed equally, they create gray.

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.

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 (MO:DCA) Architecture; formerly referred to as AFPDS
- Intelligent Printer Data Stream (IPDS) Architecture
- 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)
- Presentation Text Object Content Architecture (PTOCA)

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 AFP Consortium web site.

application. (1) The use to which an information system is put. (2) The use to which an information system is put. A collection of software components used to perform specific types of work on a computer.

architected. Identifies data that is defined and controlled by an architecture.

array. A structure that contains an ordered group of data elements. All elements in an array have the same data type.

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.

attribute. A property or characteristic of one or more constructs.

audit CMR. A CMR that reflects processing that has been done on an object.

B

base level. Within the base-and-towers concept, the smallest portion of architected function that is allowed to be implemented.

big endian. A format for storage or transmission of binary data in which the most significant bit (or byte) is placed first.

bilevel. Every point in an image can only take the values 0 or 1, i.e. black or not black.

BITS. A data type for architecture syntax, indicating one or more bytes to be interpreted as bit string information.

brightness. Attribute of a visual sensation according to which an area appears to exhibit more or less light.

BYTE. 8 bits. A data type for architecture syntax indicating that each byte has no predefined interpretation. Therefore, 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.

Cartesian coordinate system. An 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.

character. A member of a set of elements used for the organization, control, or representation of data. A character can be either a graphic character or a control character.

CIE. Commission Internationale d'Eclairage. An association of international colour scientists who produced the standards that are used as the basis of the description of colour.

CIELAB. 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. The fundamental CIE-based color space that allows colors to be expressed as a mixture of the three tristimulus values X, Y, and Z.

cluster-dot screening. A halftone method that uses multiple pixels that vary from small to larger dots as the color gets darker. It is characterized by a polka-dot look.

CMOCA. See Color Management Object Content Architecture.

CMR. See Color Management Resource.

CMYK color space. The primary colors used together in printing to effectively create a multitude of other colors: cyan, magenta, yellow, and black. Based on the subtractive color theory; the primary colors used in four color printing processes.

CODE. A data type for architecture syntax that indicates an architected constant to be interpreted as defined by the architecture.

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.

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 calibration. The process of altering the behaviour of an input or output device to make it conform to an established state, specified by a manufacturer, user, or industrywide specification or standard.

color component. A dimension of a color value expressed as a numeric value. 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 management. The technology to calibrate the color of input devices (e.g., scanners or digital cameras), display devices and output devices (e.g., printers or offset presses). Parts of this technology are implemented in the operating system (OS), the API or directly in the application.

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. Objects that provide 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 palette. A system of designated colors that are used in conjunction with each other to achieve visual consistency.

Color Rendering Dictionary. PostScript language construct for converting colors from CIEXYZ to device color space. It is analogous to the "from PCS" part of an ICC printer profile with one rendering intent, i.e. the part used when the profile is a destination profile.

color space. The method by which a color is specified. A color space specifies how color information is represented. For example, the RGB color space specifies color in terms of three intensities for red (R), green (G), and blue (B).

ColorSpace Conversion Profile. An ICC profile that provides the relevant information to perform a color space transformation between the nondevice color spaces and the PCS. It does not represent any device model. ColorSpace Conversion profiles can be embedded in images.

column. A subarray consisting of all elements that have an identical position within the low dimension of a regular two-dimensional array.

coordinate system. See Cartesian coordinate system.

D

data object . An object that conveys information, such as text, graphics, audio, or video.

data stream. A continuous stream of data that has a defined format. An example of a defined format is a structured field.

default. A value, attribute, or option that is assumed when none has been specified and one is needed to continue processing.

device attribute . A property or characteristic of a device.

device-dependent. Dependent upon one or more device characteristics.

device-independent color space. CIE-based color spaces that allow color to be expressed in a device-independent way. It ensures colors to be predictably and accurately matched among various color devices.

device profile. A structure that provides a means of defining the color characteristics of a given device in a particular state.

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.

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 *clustered dot halftoning*.

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.

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.

dpi. See dots per inch.

dots per inch. (1) The number of dots that will fit in an inch. (2) The unit of measure for output resolution. (3) Dpi is also used to measure the quality of input when using a scanner. In this case, the 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.

E

embedded ICC profile. ICC profiles that are embedded within graphic documents and images. It allow 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.

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.

exception condition. The condition that exists when a product finds an invalid or unsupported construct.

F

format. The arrangement or layout of data on a physical medium or in a presentation space.

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.

generic. Relating to, or characteristic of, a whole group or class.

GIF. Graphic Interchange Format. 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.

GOCA. See *Graphics Object Content Architecture.*

Graphics Object Content Architecture (GOCA). An architected collection of constructs used to interchange and present graphics data.

grayscale. A means of specifying color using only one color component in shades of gray ranging from white to black.

H

halftone. A method of generating on press or on a 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.

HDTV. High Definition Television is a digital television format that combines high-resolution video and theater-like sound to create a movie theater quality TV viewing experience.

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*.

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.

HSV color space. (1) A transformation of RGB 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.

host.

1. In the IPDS architecture, a computer that drives a printer.
2. In IOCA, the host is the controlling environment.

I

ICC. Acronym for International Color Consortium. A group of companies chartered to develop, use, and promote cross-platform standards so that applications and devices can exchange color data without ambiguity.

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 nondevice 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. The International Color Consortium profile format. 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. The acceptance of this format by operating system vendors allows end users to transparently move profiles and images with embedded profiles between different operating systems.

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 Object Content Architecture (IOCA). An architected collection of constructs used to interchange and present images.

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.

input device profile. The ICC profile that is associated with the image and describes the characteristics of the device for which the image was created.

instruction CMR. A CMR that reflects 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.

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.

IOCA. See *Image Object Content Architecture*.

IPDS. See *Intelligent Printer Data Stream*.

ISO. See *International Organization for Standardization*.

J

JPEG. Joint Photographic Experts Group. JPEG is a standards committee that designed an image compression format. The compression format they designed is known as a lossy compression, 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.

JFIF. JPEG File Interchange Format, 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.

L

LID. See *local identifier*.

local identifier (LID). An identifier that is mapped by the environment to a named resource.

line art. An image that contains only black and white with no shades of gray.

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.

lookup table. A table used to map one or more input values to one or more output values.

LPI. Lines Per Inch. The number of lines per inch on a halftone screen. Units used when measuring line screen frequency.

LUT. See lookup table.

luv color space. CIELUV color space. The 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

media. A physical entity on which information is presented. Examples of physical media are a sheet of paper, a roll of paper, an envelope, and a display screen.

media-relative colorimetric. This rendering intent re-scales 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).

Mixed Object Document Content Architecture. An architected, device-independent data stream for interchanging documents.

MO:DCA. See *Mixed Object Document Content Architecture*.

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.

multilevel . Every point in an image can have values from 0 to n, where n is greater than 1.

multilevel device. A device that is used in a manner that permits it to process 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**.

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.

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 text, fonts, graphics, image, and formatted data objects.
2. Something that a user works with to perform a task.

object identifier (OID). 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).

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*.

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.

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.

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*.

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.

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.

PCL. A set of printer commands, developed by Hewlett-Packard, that provide access to printer features.

PCS. See Profile Connection Space.

PDF. An acronym standing 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.

perceptual rendering intent. The exact gamut mapping of the perceptual 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.

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*.

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.

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.

presentation data stream. A presentation data stream that is processed in AFP environments. MO:DCA is the AFP interchange data stream. IPDS is 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 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.

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*.

printfile. A file that is created for the purpose of printing data. A printfile includes information to be printed and, optionally, some of the data.

process color. A color that is specified as a combination of the components, or primaries, of a color space. A process color is rendered by mixing the specified amounts of the primaries. An example of a process color is C=.1, M=.8, Y=.2, K=.1 in the cyan/magenta/yellow/black (CMYK) color space. Contrast with *spot color*.

profile. See ICC profile.

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.

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.

R

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.

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.

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.

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:

1. Fields or values that have been used by a product in a manner not compliant with the architected definition.
2. Fields or values that have been removed from architecture.

resolution. The sharpness of text and graphics provided by any output device. On printed media, it is the number of dots per inch; on a video monitor, it is the number of pixels per unit of measurement. In general, the higher the dpi, the sharper the image.

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.

RGB color space. Red, green and blue, the additive primaries. RGB is the basic additive color model used for color video display, as on a computer monitor.

RIP. A raster image processor is a hardware or software tool that processes a presentation data stream and converts it - rasterizes it- to a printable format.

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.

S

saturation rendering intent. The exact gamut mapping of the saturation 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.

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.

semantics. The meaning of the parameters of a construct. See also *syntax*.

server. In a network, hardware or software that provides facilities to other stations. Examples of a server are a file server, a printer server, or a mail server.

sheet. A division of the physical medium; multiple sheets can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a sheet. Envelopes are an example of a physical medium that comprises only one sheet. The IPDS architecture defines four types of sheets: cut-sheets, continuous forms, envelopes, and computer output on microfilm. Each type of sheet has a top edge. A sheet has two sides, a front side and a back side.

signed integers. The integers consist of 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 (or Z in blackboard bold,), that stands for Zahlen (German for “numbers”).

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 monitor or printed paper or any other output device. sRGB was promoted by the ICC and submitted for standardization by the IEC.

stochastic. A method that uses a pseudo-random dot size and/or frequency to create halftoned images, but without the visible regularity in the dot patterns found in traditional screening.

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*.

subtractive color. As opposed to Additive Primary Colors. Colorants of cyan, magenta, and yellow are 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.

SWOP. Abbreviation for the Specifications for Web Offset. A standard set of specifications for color separations, proofs, and printing to ensure consistency of color printing.

syntax. The rules governing the structure of a construct. See also *semantics*.

T

tag. A data structure that is used within the data portion of the CMR. A CMR tag consists of TagID, FieldType, Count, and ValueOffset.

text. A graphic representation of information. Text can consist of alphanumeric characters and symbols arranged in paragraphs, tables, columns, and other shapes. An example of text is the data sent in an IPDS Write Text command.

text orientation. A description of the appearance of text as a combination of inline direction and baseline direction. See also *orientation* and *presentation space orientation*.

TIFF. Tagged Image File Format. A graphics file format using bitmaps.

triplet. A three-part self-defining variable-length parameter consisting of a length byte, an identifier byte, and one or more parameter-value bytes.

tone transfer curve. A mathematical representation of the relationship between the input and output of a system, subsystem, or equipment. The function is normally 1-dimensional, i.e., single channel of input corresponds 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.

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.

U

UBIN. A data type for architecture syntax, indicating one or more bytes to be interpreted as an unsigned binary number.

UTC. Coordinated Universal Time, the standard time reference for Earth and the human race. Knowing the UTC time and one's timezone offset from it, makes it possible to calculate the local time; for example, 1:00 PM UTC corresponds to 3:00 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.

UTF16. A datatype for architecture syntax, indicating UTF-16BE characters.

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).

V

vector graphics. A vector has a defined starting point, a designated direction and a specified distance. Vector graphics are line-based graphics where, vectors determine how straight and curved lines are shaped between specific points. The lines and the colors fill the areas enclosed by the lines make up the picture.

W

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.

Y

YCbCr. Three-component color space that approximately models how color is interpreted by the human visual system, with an intensity value and two color values.

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 HSV space.

YCrCb. Three-component color space that approximately models how color is interpreted by the human visual system, with an intensity value and two color values.

Index

Numerics

A

B

C

D

E

G

H

I

J

L

M

N

O

U

W

Y

Advanced Function Presentation Consortium

Color Management Object Content Architecture Reference

AFPC-0006-01