INTERNATIONAL STANDARD

ISO 15076-1

Second edition
2010-12-01

Image technology colour management — Architecture, profile format and data structure —

Part 1: **Based on ICC.1:2010**

Gestion de couleur en technologie d'image — Architecture, format de profil et structure de données —

Partie 1: Fondé sur l'ICC.1:2010

Reference number
ISO 15076-1:2010(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 15076-1 was prepared by the *International Color Consortium*, in cooperation with Technical Committees ISO/TC 130 *Graphic technology* and ISO/TC 42 *Photography*, under the provisions of the Cooperative Agreement between ISO/TC130 and the *International Color Consortium* dated 2003-07-11.

This second edition cancels and replaces the first edition (ISO 15076-1:2005), which has been technically revised to incorporate changes made in the profile specification. These include the addition of the perceptual intent reference medium colour gamut, new technology signatures, a floating-point device encoding range, a colorimetric intent image state tag, and a profile sequence identifier tag. In addition, the mediaBlackPointTag has been deleted and PCSXYZ is no longer limited to the PCS illuminant.

ISO 15076-1 is technically identical to ICC.1:2010, *Image technology colour management — Architecture, profile format, and data structure (Profile version 4.3.0.0)*.

ISO 15076 consists of the following parts, under the general title *Image technology colour management — Architecture, profile format and data structure*:

— *Part 1: Based on ICC.1:2010*

Introduction

0.1 General

This part of ISO 15076 specifies the profile format defined by the International Color Consortium® (ICC). The intent of this format is to provide a cross-platform profile format for the creation and interpretation of colour data. Such profiles can be used to translate between different colour encodings, and to transform colour data created using one device into another device's native colour encoding. The acceptance of this format by application and operating system vendors allows end users to transparently move profiles, and images with embedded profiles, between different systems. For example, this allows a printer manufacturer to create a single profile for multiple applications and operating systems.

It is assumed that the reader of this part of ISO 15076 has a good understanding of colour science and imaging, such as familiarity with CIE, ISO and IEC colour standards, general knowledge of device measurement and characterization, and familiarity with at least one operating system level colour management system.

0.2 International Color Consortium

The International Color Consortium was formed with the primary intent of developing and administering a colour profile format standard, and for the registration of the associated tag signatures and descriptions. The founding members of this consortium were Adobe Systems Inc., Agfa-Gevaert N.V., Apple Computer, Inc., Eastman Kodak Company, FOGRA (Honorary), Microsoft Corporation, Silicon Graphics, Inc., Sun Microsystems, Inc., and Taligent, Inc. These companies committed to fully support the standard in their operating systems, platforms and applications. The consortium has since been expanded and now has over 60 members.

The initial version of the standard developed by the ICC has undergone various revisions, and it was agreed by the ICC that its revision 4.2 first be proposed as an International Standard. It is that revision which formed the basis of first edition of this part of ISO 15076 (ISO 15076-1:2005). This second edition is based on ICC revision 4.3, which is a minor ICC revision and is therefore fully backward compatible with 4.2. All the technical specifications contained in the first edition (ISO 15076-1:2005) are also given in this second edition, and new specifications exclusive to this second edition are clearly identified. Informative material has also been updated and clarified. The ICC will continue to administer its own version of ICC.1:2010 and, if enhancements are made, will be seriously considered for future revisions of this part of ISO 15076. ISO/TC 130 will work to ensure that there are no significant differences between the ICC and ISO versions of this part of ISO 15076.

The ICC web site (www.color.org) provides supplementary information relevant to this part of ISO 15076 and additional resources for developers and users. It also provides information on how to become a member of ICC.

0.3 Colour management architecture and profile connection space

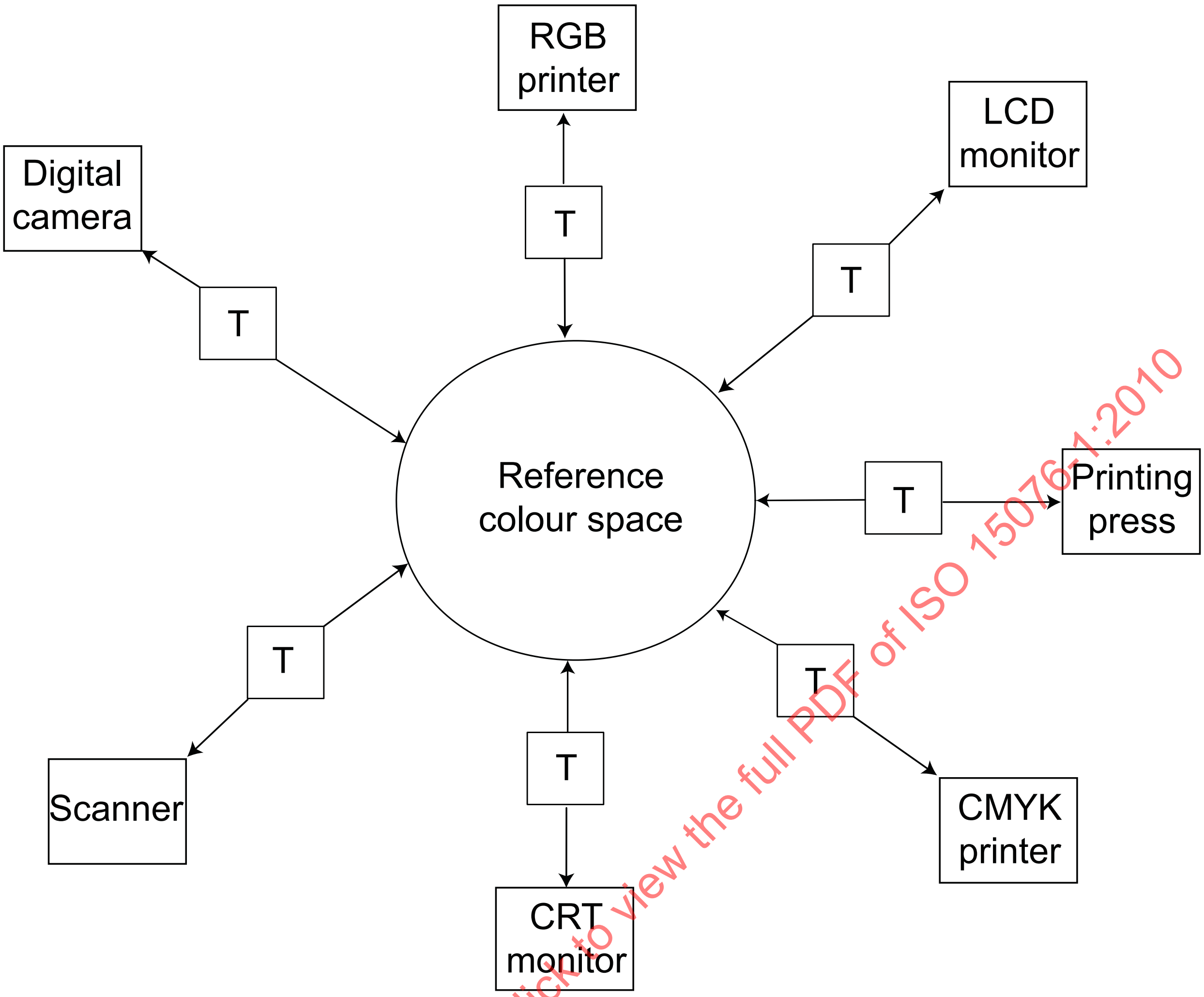
The underlying architecture assumed in this part of ISO 15076 is based around a reference colour space that is unambiguously defined. The colour specification method selected was that defined by CIE which is internationally accepted. The CIE system enables a set of tristimulus values (CIEXYZ) to be specified for a coloured stimulus. These tristimulus values enable a user to determine whether colours match in appearance when viewed by a typical observer in a specific viewing environment. It follows that it is possible to define the colour appearance of a sample by these tristimulus values (or some defined transformation of them) for a specified state of viewer adaptation. The colour appearance is simply the appearance of the colour to a typical human observer, as opposed to the physical characteristics of the colour stimulus, which is not fully specified using tristimulus values.

Calculation of the CIEXYZ values for transmitting or reflecting media is achieved from the spectral sum-product of the reflectance or transmittance of the sample, the relative spectral power distribution of the illumination source used to view it and the spectral 'sensitivity' of the standard observer. However, as CIE defines two standard observers, two measurement geometries (for reflecting media) and a large number of standard illuminants, it is necessary to restrict these options in order to have a colour specification system that is not ambiguous for a particular application. For this part of ISO 15076, the ICC has defined such a restriction, based on ISO 13655, and the resultant colour spaces are known as PCSXYZ and PCSLAB. Furthermore, the simple CIE system (whether CIEXYZ or the CIELAB values derived from them) does not accommodate the effect of surrounding stimuli to the sample being measured (which can be different for various types of media) or the illumination. Both of these affect appearance so the PCS values do not by themselves specify appearance. To overcome this problem, the PCS is used in two different ways. The first accounts only for the assumed state of chromatic adaptation of the viewer, and describes the colorimetry of actual originals and their reproductions, chromatically adapted to the PCS adopted white chromaticity, through the colorimetric rendering intents. The second, which describes the colorimetry of an image colour rendered to a standard reference medium under a specified viewing condition, is employed for the perceptual rendering intent and optionally for the saturation rendering intent. Thus, it can incorporate corrections for different states of viewer adaptation and other desired rendering effects, as well as accommodating differences between actual colour encoding and device dynamic ranges and colour gamuts, and those of the perceptual intent reference medium. When required, the viewing conditions can be specified to allow colour appearance to be determined for the colorimetric rendering intents.

So, in summary, the PCS is based on CIEXYZ (or CIELAB) determined for a specific observer (CIE Standard 1931 Colorimetric Observer, often known colloquially as the 2 degree observer), relative to a specific illuminant chromaticity (that of CIE D50), and measured with a specified measurement geometry (0°:45° or 45°:0°), for reflecting media. Measurement procedures are also defined for transmitting and self-luminous media. Since the conversion from CIEXYZ to CIELAB is quite unambiguous, profile builders can use either colour space for the PCS; the colour management system is able to determine which has been used from a tag in the header.

For colorimetric renderings where the measured data were not obtained relative to a D50 adopted white chromaticity, the profile builder is expected to adapt the data to achieve this. Therefore, a mechanism for identifying the chromatic adaptation used in such situations is provided. For the perceptual rendering intent the viewing conditions and reference medium are specified in order to provide a clear target for colour rendering and re-rendering (including gamut mapping). In the following paragraphs, the reference colour space, to which reference is made, needs to include the viewing conditions and reference medium when the perceptual intent is being considered. For the perceptual rendering intent, profile builders are expected to undertake any corrections for appearance effects if the viewing conditions used for monitors and transmitting media (such as dark surrounds) differ from those typical for reflecting media, and to account for differences between actual media and the reference medium.

Figure 1 shows how a reference colour space can be used to provide the common interface for transformations between different colour encodings, as used by different devices, or even different operational modes of the same device. Without it, a separate transformation would be required for each pair of device modes. If there are n device modes to be supported in a system, and it is necessary to provide a transformation between each pair of device modes, n^2 transforms would need to be defined and n new transforms would need to be defined every time a new device mode was added. As a new printer device mode can consist simply of a new paper type, this is not a practical solution. By using a reference colour space, only n transforms need be defined and only one new transform needs to be defined each time a new device mode is added; whatever device-to-device transforms are needed can be constructed by linking the source and destination profiles using the reference colour space as the interface.



Key

T colour management transform

Figure 1 — Use of a reference colour space

While images can be encoded directly in PCSXYZ or PCSLAB, this will not generally be the case. A number of colour encodings for open exchange have been standardized to meet a variety of needs. Depending on the use case, different bit depths, image states, reference media and colour gamuts are needed. Devices also have different characteristics resulting in different native encodings. Except for a few cases where default encodings for key system devices are used for exchange (like the sRGB encoding), it is not practical or productive to attempt to restrict system colour encoding support.

For reasons of precision, it is usually desirable to define the transformation between the colour data encoding and the profile connection space (PCS) at a high precision. If the transformation between a colour data encoding and the PCS is provided with an image file, it can be utilized when images are reproduced. In order that the transformation between the colour data encoding and the PCS can be interpreted by all applications it is important that it be defined in an open specification. The profile format defined in this part of ISO 15076 provides that specification.

0.4 Rendering intents

In general, actual device colour gamuts will fail to match each other, and that of the perceptual intent reference medium, to varying degrees. Because of this mismatch, and because of the needs of different applications, four rendering intents (colour rendering styles) are defined in this part of ISO 15076. Each one represents a different colour reproduction objective. The colorimetric rendering intents operate directly on measured colorimetric values, with correction for chromatic adaptation when the measured values were not obtained relative to the PCS adopted white chromaticity. The other rendering intents (perceptual and saturation) operate on colorimetric values which are adjusted in an as-needed fashion to account for any differences between devices, media, and viewing conditions.

Two colorimetric rendering intents are specified in this part of ISO 15076, though only one is included fully constructed in the profile. The included media relative colorimetric intent is based on media-relative colorimetry, which is normalized relative to the unprinted media white for reflecting, transmitting, and self luminous media, or, in the case of colour encodings and capture, to the colour encoding values that correspond to the highest perceived brightness. Thus the media white will have the values of 100, 0, 0 in PCSLAB. This ensures that highlight clipping will not occur when the media-relative colorimetric intent is used. The use of media-relative colorimetry enables colour reproductions to be defined which maintain highlight detail, while keeping the medium ‘white’, even when the original and reproduction media differ in colour. However, this rendering intent introduces some change in all colours in the reproduction when the media whites of the source and destination do not match.

The PCS adopted white is defined to be the radiance of a perfect reflecting diffuser illuminated by a source with a spectral power distribution matching that of CIE Illuminant D50. ICC profiles contain the values of the media white, adapted to be relative to the chromaticity of the PCS adopted white. For the ICC-absolute colorimetric rendering intent, all of the colorimetric values are re-calculated to be relative to the tristimulus values of the PCS adopted white. When source and destination viewing conditions are identical and an exact colour match is required for all within-gamut colours (including the source medium colour), the ICC-absolute colorimetric rendering intent should be used. This rendering intent can also be useful in other situations.

The colour rendering of the perceptual and saturation rendering intents is vendor specific. The former, which is useful for general reproduction of pictorial images, typically includes tone scale adjustments to map the dynamic range of one medium to that of another, and gamut reshaping and mapping to deal with gamut mismatches. The latter has historically been useful for images which contain objects such as charts or diagrams, and usually involves compromises such as trading off preservation of hue in order to preserve the vividness of pure colours. As the saturation rendering intent is neither required to contain colorimetric characterization information or to use the perceptual intent reference medium, it is the only option, in proprietary systems, for providing colour rendering and re-rendering transforms to and from custom reference media represented in the PCS. For broader interoperability when using the saturation rendering intent, the perceptual reference medium can be used, and its use indicated.

For perceptual transforms it is necessary, in order to optimize colour rendering, to provide a realistic target for the colour rendering. For this reason, a reference medium and reference viewing condition have been defined which apply only to the perceptual rendering. The reference medium is defined as a hypothetical reflection print on a substrate with a white having a neutral reflectance of 89 %, and a density range of 2,459 3. The reference viewing condition is the P2 condition specified in ISO 3664, i.e. D50 at 500 lx for viewing reflecting media. A neutral surround of 20 % reflectance is assumed. The colour gamut of the reference medium is qualitatively specified as that of a reflection print, and whatever colour gamut is used in the PCS is required to be consistent with the specified dynamic range of the perceptual reference medium. It is recommended that the reference gamut specified in ISO 12640-3 be used as a more explicit target gamut for improved interoperability. Profile creators should consider this gamut to be the target for perceptual intent colour rendering and re-rendering to the PCS. Likewise, the perceptual intent colour re-rendering from the PCS needs to assume this gamut as the starting gamut for colour re-rendering to the destination medium. However, even when the use of this gamut is indicated, perceptual intent transforms need to be designed to produce the best visual results, and need not conform exactly to this gamut in the PCS.

The choice of a reference medium with a realistic black point for the perceptual intent provides a well-defined aim when colour rendering or re-rendering are required. Inputs with a dynamic range greater than a reflection print (e.g. a slide film image, or the colorimetry of high-range scenes) can have their highlights and shadows smoothly compressed to the range of the reference medium in such a way that these regions can be expanded again without undue loss of detail on output to wide-range media. Likewise, images from original media with limited dynamic range can be colour re-rendered to the expanded dynamic range of the reference medium, in order to produce better quality in subsequent reproductions. Bi-directional transform pairs (e.g. data-to-PCS and PCS-to-data for each rendering intent) in the profiles can be used to undo prior PCS-to-data colour re-rendering so that a differently optimized reproduction can be produced for a different reproduction medium.

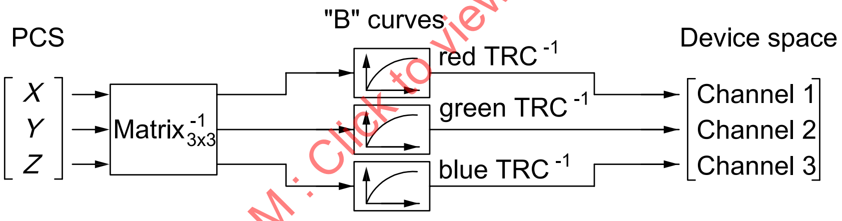
Profiles generally offer different transformations for different rendering intents. When the rendering intent is selected the corresponding transformation is selected by the colour management system. The choice of rendering intent is highly dependent upon the intended use. In general, the perceptual rendering intent is most applicable for the colour re-rendering of natural images, to make pleasing and aesthetically similar, but not exactly matching, reproductions on different media. The ICC-absolute colorimetric rendering intent is most appropriate for a proofing environment, where the colour reproduction obtained on one device is simulated on another. The media-relative colorimetric rendering intent is appropriate when mapping of the source medium white to the destination medium white is desired, but a full colour re-rendering is not.

For those requiring further information, an extended discussion of many of the issues described above is provided in Annex D.

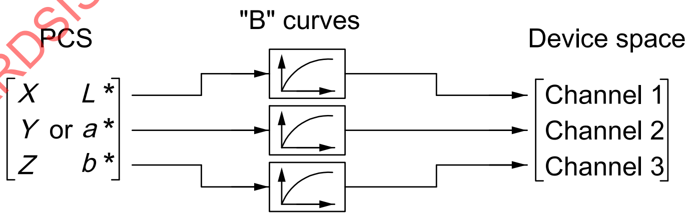
0.5 Colour profiles

Colour profiles provide colour management systems with the information necessary to convert colour data between different colour encodings, including device encodings. This part of ISO 15076 divides colour devices into three broad classifications, i.e. input devices, display devices and output devices. For each device class, a series of base algorithmic models are described which perform the transformation between colour encodings. Figures 2 and 3 show examples of these models, which provide different trade-offs in memory footprint, colour quality and performance results. The matrix tone reproduction curve (TRC) model is explained in detail in 8.3.3 and 8.4.3, the lutAToBType and lutBToAType in 10.10 and 10.11, and the multiProccessElementsType in 10.14. The necessary parameter data to implement these models is described in the appropriate tag type descriptions in Clause 10. This required data provides the information for the colour management framework default colour management module (CMM) to transform colour information between colour encodings. A representative architecture using these components is illustrated in Figure 4.

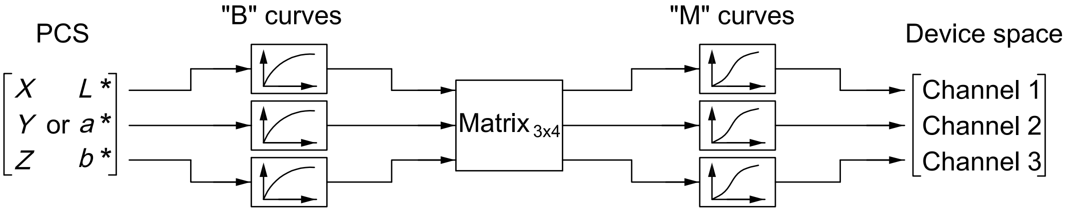
NOTE Only the models shown in Figures 2d), 2e), 2f), 3d), 3e) and 3f) can be used if the device space has more than three components/colours.



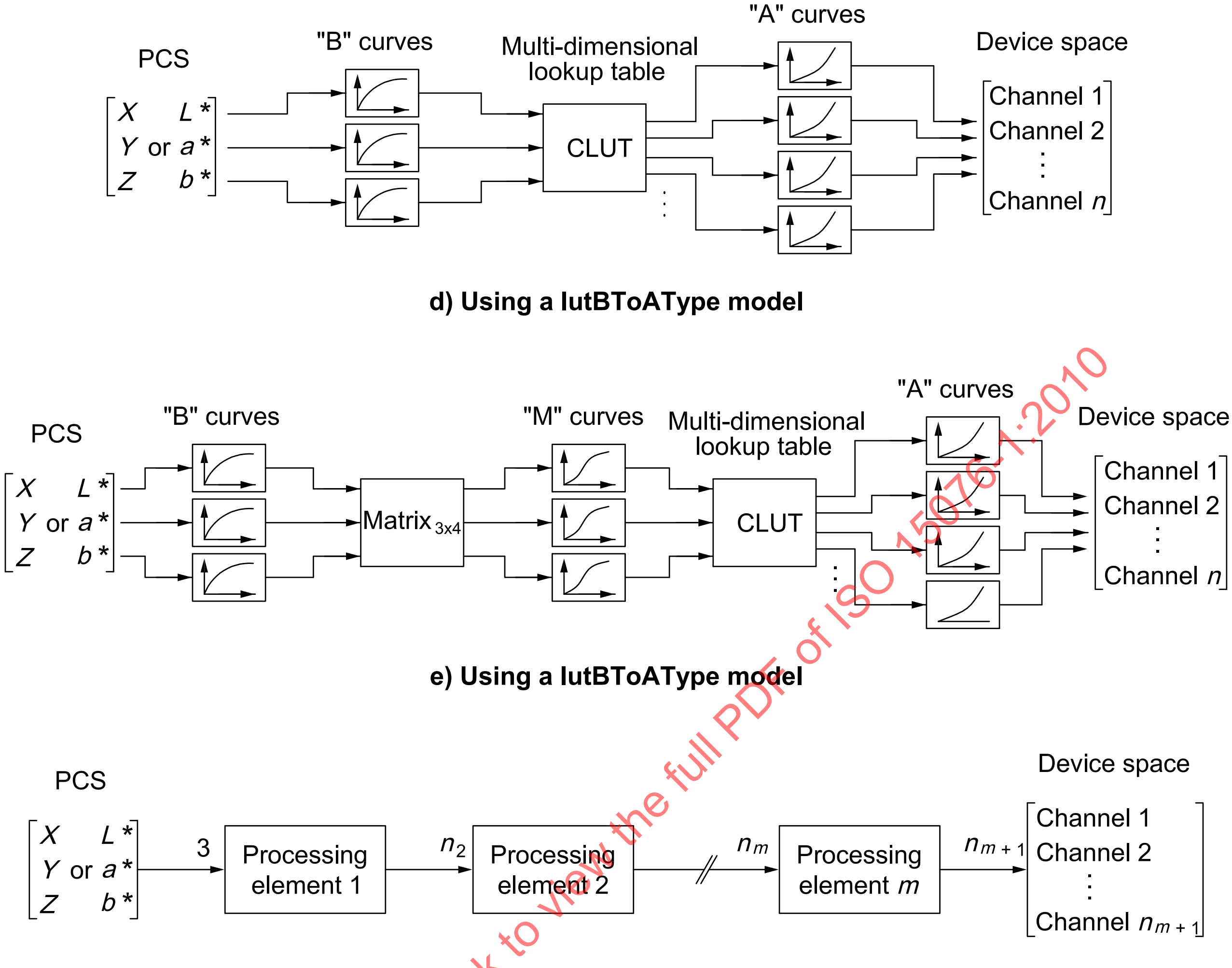
a) Using a matrix/TRC model



b) Using a lutBToAType model



c) Using a lutBToAType model

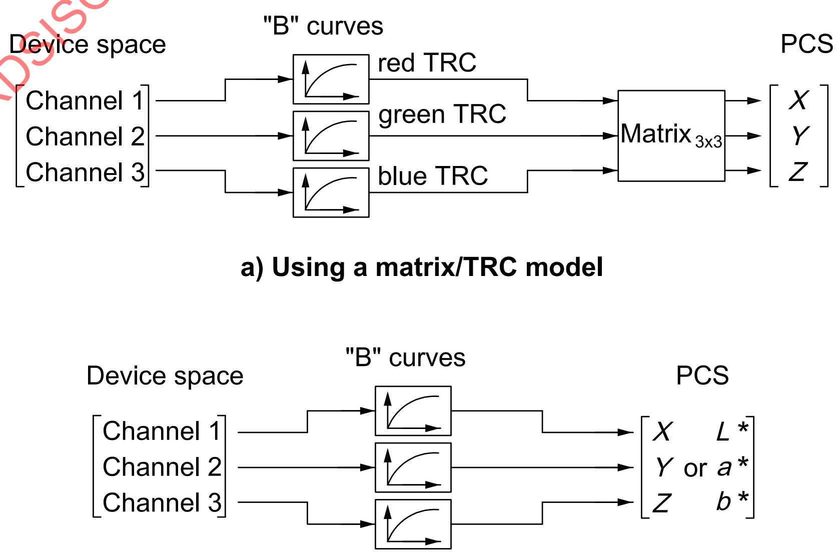


d) Using a lutBToAType model

e) Using a lutBToAType model

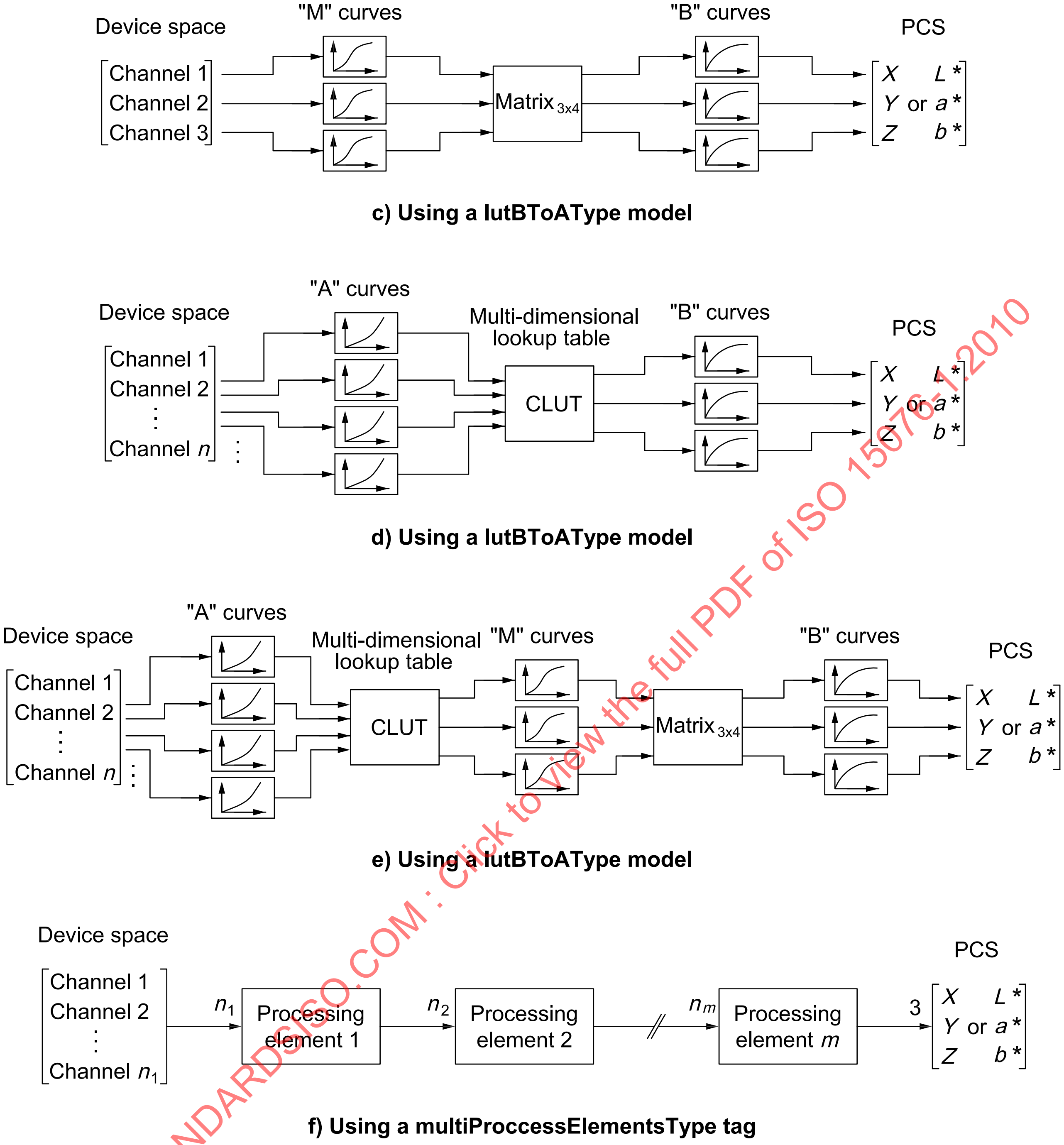
f) Using a multiProcessElementsType tag

Figure 2 — Examples of different ways of converting a colour from PCS to device space



a) Using a matrix/TRC model

b) Using a lutAToBType model



c) Using a lutBToAType model

d) Using a lutBToAType model

e) Using a lutBToAType model

f) Using a multiProccessElementsType tag

Figure 3 — Examples of different ways of converting a colour from device to PCS

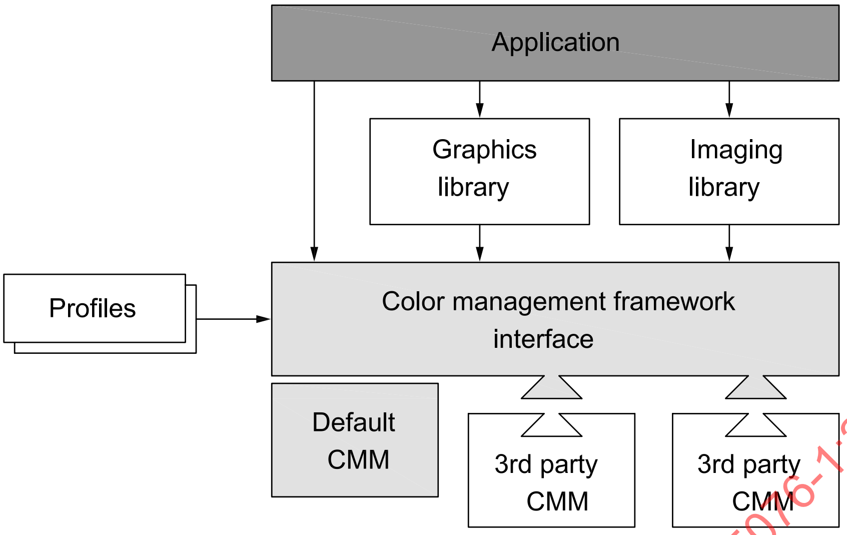


Figure 4 — Colour management architecture

0.6 Profile element structure

The profile structure is defined as a header followed by a tag table followed by a series of tagged elements that can be accessed randomly and individually. This collection of tagged elements provides three levels of information for developers: required data, optional data and private data. An element tag table provides a table of contents for the tagging information in each individual profile. This table includes a tag signature, the beginning address offset and size of the data for each individual tagged element. Signatures in this part of ISO 15076 are defined as a 4-byte hexadecimal number. This tagging scheme allows developers to read in the element tag table and then randomly access and load into memory only the information necessary to their particular software application. Since some instances of profiles can be quite large, this provides significant savings in performance and memory. The detailed descriptions of the tags, along with their intent, are included later in this part of ISO 15076.

The required tags provide the complete set of information necessary for the CMM to translate colour information between the PCS and the data colour encoding. Each profile class determines which combination of tags is required.

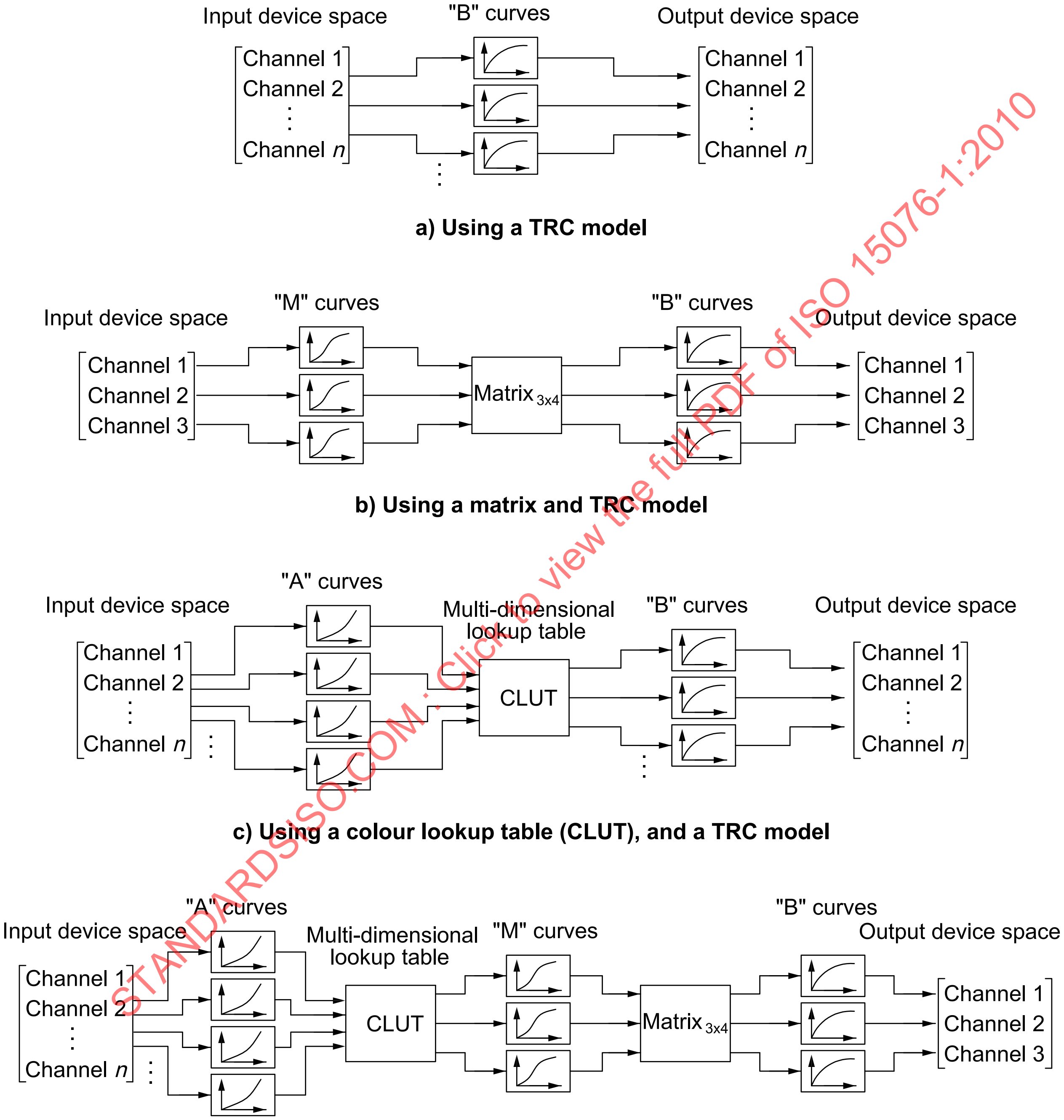
In addition to the required tags for each colour profile, a number of optional tags are defined that can be used for enhanced capabilities. In the case of required and optional tags, all of the signatures, an algorithmic description (where appropriate), and intent are registered with the International Color Consortium. Private data tags allow CMM developers to add proprietary value to their profiles. By registering just the tag signature and tag type signature, developers are assured of maintaining their proprietary advantages while maintaining compatibility with this part of ISO 15076. However, since the overall philosophy of this format is to maintain an open, cross-platform standard, developers are encouraged to keep the use of private tags to an absolute minimum.

0.7 Embedded profiles

In addition to providing a cross-platform standard for the colour profile format, this part of ISO 15076 also describes the convention for embedding these profiles within graphics documents and images. Embedded profiles allow users to transparently move colour data between different computers, networks and even operating systems without having to worry if the necessary profiles are present on the destination systems. The intention of embedded profiles is to allow the interpretation of the associated colour data. Embedding profiles are described in Annex B of this part of ISO 15076.

0.8 Other profiles

Four profile types, in addition to the device profile types described above, are defined in this part of ISO 15076. DeviceLink profiles provide a dedicated transformation from one device encoding to another, which can be useful in situations where such a transformation is used frequently or has required optimisation to achieve specific objectives. (Figure 5 shows the various algorithmic models which can be used to construct a DeviceLink profile.)

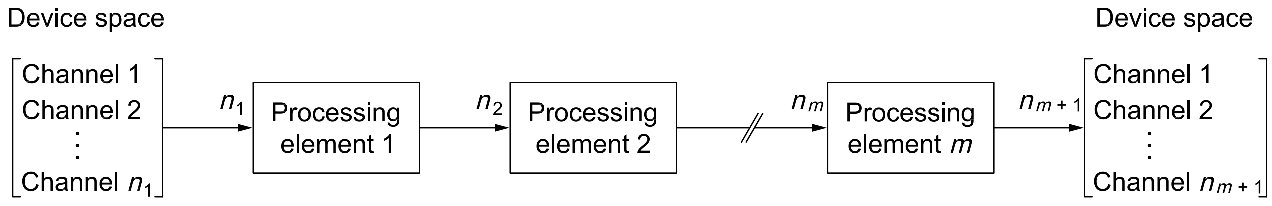


a) Using a TRC model

b) Using a matrix and TRC model

c) Using a colour lookup table (CLUT), and a TRC model

d) Using a CLUT, a matrix, and a TRC model



e) Using a multiProccessElementsType tag

Figure 5 — Examples of converting a colour from device to device using a DeviceLink profile

ColorSpace profiles provide transformations between standard colour encodings and the PCS, providing a means for supporting existing and future colour encodings with backward compatibility. Abstract profiles are defined from PCS to PCS and enable colour transformations to be defined that provide some specific colour effects. NamedColor profiles provide a mechanism for specifying the relationship between device values and the PCS for specific colours, rather than for general images.

0.9 Organizational description of this part of ISO 15076

This part of ISO 15076 addresses a very complex set of issues and it has been organized to provide a clear, clean, and unambiguous explanation of the entire format. To accomplish this, the overall presentation is from a top-down perspective, beginning with the summary overview presented above, followed by the necessary background information and definitions needed for unambiguous interpretation of the text. A description of the PCS and rendering intents is then provided before continuing down at increasing levels of detail into a byte stream description of the format. Clause 6 describes the PCS and rendering intents; Clause 7 describes the structure of the various fields required in a profile; and Clause 8 describes the content of the required tags for each profile class. Clause 9 lists the various tags (optional and required) and briefly summarizes the function of the tags as well as listing the signature and permitted tag types for each. The tag types are defined in Clause 10. Annex A provides additional information pertaining to the data colour encodings and rendering intents used in this part of ISO 15076 while Annex B provides details for embedding profiles into EPS, TIFF, and JPEG files. Annex C provides a general description of the PostScript Level 2 tags used in this part of ISO 15076 while Annex D provides some background material on the PCS. Annex E provides additional information pertaining to chromatic adaptation and the chromaticAdaptationTag while Annex F describes some computational models assumed in this part of ISO 15076. Annex G summarizes in tabular form the required tags for each profile class as specified in Clause 8.

0.10 Patent statement

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this part of ISO 15076 can involve the use of a patent concerning the outputResponseTag, (support of the outputResponseTag is optional), given in 9.2.36.

ISO takes no position concerning the evidence, validity and scope of this patent right. The holder of this patent right has assured the ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Intellectual Property Standards and Transactions
Eastman Kodak Company
343 State Street,
Rochester, NY 14650
USA

Attention is drawn to the possibility that some of the elements of this part of ISO 15076 may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

Image technology colour management — Architecture, profile format and data structure —

Part 1: **Based on ICC.1:2010**

1 Scope

This part of ISO 15076 specifies a colour profile format and describes the architecture within which it can operate. This architecture supports the exchange of information which specifies the intended colour image processing of digital data. The required reference colour spaces and the data structures (tags) are also specified.

NOTE The technical content of this part of ISO 15076 is identical to that of ICC.1:2010.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 5-3, *Photography and graphic technology — Density measurements — Part 3: Spectral conditions*

ISO 639-1, *Codes for the representation of names of languages — Part 1: Alpha-2 code*

ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 3664, *Graphic technology and photography — Viewing conditions*

ISO 13655, *Graphic technology — Spectral measurement and colorimetric computation for graphic arts images*

DIN 16536-2, *Testing of prints and printing inks in graphic technology — Colour density measurements on on-press or off-press prints — Part 2: Instrument specifications for reflection densitometers and their calibration*

EBU Tech. 3213-E, *EBU standard for chromaticity tolerances for studio monitors*

ITU-R BT.709-2, *Parameter values for the HDTV standards for production and international programme exchange*

SMPTE RP 145, *SMPTE C Color Monitor Colorimetry. Available from:* <http://store.smpte.org/category-s/22.htm>

Internet RFC 1321, *The MD5 Message-Digest Algorithm*, R. Rivest, April 1992, Available from Internet <ftp://www.ietf.org/rfc/rfc1321.txt>

IEEE 754, *Standard for Binary Floating-Point Arithmetic. Available from:* <http://ieeexplore.ieee.org>

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
adopted white
spectral radiance distribution as seen by an image capture or measurement device and converted to colour signals that are considered to be perfectly achromatic and to have an observer adaptive luminance factor of unity; i.e. colour signals that are considered to correspond to a perfect white diffuser

NOTE 1 The adopted white can vary within a scene.

NOTE 2 No assumptions should be made concerning the relationship between the adapted or adopted white and measurements of near perfectly reflecting diffusers in a scene, because measurements of such diffusers will depend on the illumination and viewing geometry, and other elements in the scene that can affect perception. It is easy to arrange conditions for which a near perfectly reflecting diffuser will appear to be grey or coloured.

[ISO 22028-1]

3.1.2
ASCII text string
sequence of bytes, each containing a graphic character specified in ISO/IEC 646, the last character in the string being a NULL (character 0/0)

3.1.3
big-endian
addressing the bytes within a 16-bit, 32-bit or 64-bit value from the most significant to the least significant, as the byte address increases

3.1.4
bit position
bits are numbered such that bit 0 is the least significant bit

3.1.5
byte
8-bit unsigned binary integer

3.1.6
byte offset
number of bytes from the beginning of a field

3.1.7
CIELAB
CIE 1976 L^*, a^* and b^* values calculated from nCIEXYZ according to CIE 15

3.1.8
CIEXYZ
XYZ tristimulus values based on the CIE 1931 Standard Colorimetric Observer as defined in CIE 15

NOTE Y is expressed in candelas per square meter.

3.1.9
nCIEXYZ
CIEXYZ values that have been uniformly scaled so that $Y = 1{,}0$ for the adopted white

NOTE In this part of ISO 15076, this is referred to as ICC-absolute colorimetry.

3.1.10
colour encoding
generic term for a quantized digital encoding of a colour space, encompassing both colour space encodings and colour image encodings

[ISO 22028-1]

NOTE Values specified by an encoding are the closest representation to the colour space or image values permitted by the encoding precision.

3.1.11
colour management
communication of the associated data required for unambiguous interpretation of colour content data, and application of colour data conversions, as required, to produce the intended reproductions

NOTE 1 Colour content can consist of text, line art, graphics, and pictorial images, in raster or vector form, all of which can be colour managed.

NOTE 2 Colour management considers the characteristics of input and output devices in determining colour data conversions for these devices.

3.1.12
colour rendering
mapping of image data representing the colour-space coordinates of the elements of a scene to output-referred image data representing the colour-space coordinates of the elements of a reproduction

NOTE Colour rendering generally consists of one or more of the following:

— compensating for differences in the input and output viewing conditions,

— tone scale and gamut mapping to map the scene colours onto the dynamic range and colour gamut of the reproduction, and

— applying preference adjustments.

[ISO 22028-1]

3.1.13
encoding maximum white
highest luminance achromatic colour that can be represented using a specified colour encoding

NOTE For the purpose of this definition, a colour is achromatic if it has the same chromaticity as the adopted white. The choice of the adopted white is an user decision.

3.1.14
fixed point
method of encoding a real number into binary by putting an implied binary point at a fixed bit position

NOTE Many of the tag types defined in this part of ISO 15076 contain fixed point numbers. Several references can be found (MetaFonts, etc.) illustrating the benefit of fixed point representation when compared to pure floating point representation in very structured circumstances.

3.1.15
hexadecimal
numeral system with a radix, or base, of 16, written using the symbols 0–9 and A–F, or a–f

NOTE The notation used to indicate hexadecimal numbers in this part of ISO 15076 is xxh.

3.1.16
image state
attribute of a colour image encoding indicating the rendering state of the image data

[ISO 22028-1]

3.1.17
media white point
reference colour that is used as the basis for scaling of media relative transforms

NOTE In a reproduction process, this is usually the colour with the highest luminance that can be produced by an imaging medium, measured using the specified measurement geometry. In a capture process this is usually the colour associated with the device side encoding maximum white.

3.1.18
NULL
character coded in position 0/0 as specified in ISO/IEC 646

3.1.19
PCS
profile connection space
colour space used to connect the source and destination profiles

NOTE See Annex D for a full description.

3.1.20
PCS adopted white
adopted white where the spectral radiance distribution is that of the PCS illuminant

3.1.21
PCS illuminant
illuminant with the spectral radiance distribution of CIE illuminant D50 and nCIEXYZ $X = 0{,}964\ 2$, nCIEXYZ $Y = 1{,}0$, nCIEXYZ $Z = 0{,}824\ 9$

3.1.22
PCSLAB
CIELAB values calculated from PCSXYZ

3.1.23
PCSLAB encoding
PCSLAB values that have been encoded as 8-bit or 16-bit numbers, or as floating point numbers

3.1.24
PCSXYZ
nCIEXYZ values that have been linearly scaled so that PCSXYZ $X = 0{,}964\ 2$, PCSXYZ $Y = 1{,}0$, PCSXYZ $Z = 0{,}824\ 9$ for media white

3.1.25
PCSXYZ encoding
PCSXYZ values that have been encoded as 16-bit numbers, or as floating point numbers

3.1.26
picture-referred image state
image state associated with image data that represents the colour-space coordinates of the elements of a hardcopy or softcopy image, encompassing both original-referred image data and output-referred image data

NOTE 1 When the phrase "picture-referred" is used as a qualifier to an object, it implies that the object is in a picture-referred image state. For example, picture-referred image data are image data in a picture-referred image state.

NOTE 2 Picture-referred image data will generally be colour-rendered for a specific real or virtual imaging medium and viewing condition.

NOTE 3 Picture-referred image data can include image data that do not originate from an original scene, such as text, line art, vector graphics and other forms of original artwork.

[ISO 22028-1]

3.1.27
rendering intent
style of mapping colour values from one image description to another

NOTE See Clause 6 and Annexes A and D for a description of the four rendering intents (ICC-absolute colorimetric, relative colorimetric, perceptual and saturation) used in ICC profiles

3.1.28
scene
spectral radiances of a view of the natural world as measured from a specified vantage point in space and at a specified time

NOTE A scene can correspond to an actual view of the natural world or to a computer-generated virtual scene simulating such a view.

[ISO 22028-1]

3.1.29
spot colour
single colorant, identified by name, whose printing tone-values are specified independently from the colour values specified in a colour coordinate system

3.1.30
signature
alphanumerical 4-byte value, registered with the ICC

NOTE Shorter values are padded at the end with 20h bytes.

3.1.31
viewing flare
veiling glare that is observed in a viewing environment but not accounted for in radiometric measurements made using a prescribed measurement geometry

[ISO 22028-1]

NOTE The viewing flare is expressed as a percentage of the luminance of adapted white.

3.1.32
veiling glare
light, reflected from an imaging medium, that has not been modulated by the means used to produce the image

[ISO 22028-1]

3.2 Abbreviated terms

ANSI	American National Standards Institute
CIE	*Commission Internationale de l'eclairage* (International Commission on Illumination)
CLUT	Colour lookup table (multi-dimensional)
CMM	Colour management module
CMY	Cyan, magenta, yellow
CMYK	Cyan, magenta, yellow, key (black)

CRD	Colour rendering dictionary
CRT	Cathode-ray tube
EPS	Encapsulated PostScript
ICC	International Color Consortium
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LCD	Liquid crystal display
LUT	Lookup table
PCS	Profile connection space
RGB	Red, green, blue
TIFF	Tagged Image File Format
TRC	Tone reproduction curve

4 Basic number types

4.1 General

The basic numeric types used in this part of ISO 15076 are defined in 4.2 to 4.15.

NOTE As defined in 7.1, all profile data is encoded as big-endian.

4.2 dateTimeNumber

A dateTimeNumber is a 12-byte value representation of the time and date, where the byte usage is assigned as specified in Table 1. The actual values are encoded as 16-bit unsigned integers (uInt16Number, see 4.10).

Table 1 — dateTimeNumber

Byte position	Field length bytes	Content	Encoded as
0 to 1	2	Number of the year (actual year, e.g. 1994)	uInt16Number
2 to 3	2	Number of the month (1 to 12)	uInt16Number
4 to 5	2	Number of the day of the month (1 to 31)	uInt16Number
6 to 7	2	Number of hours (0 to 23)	uInt16Number
8 to 9	2	Number of minutes (0 to 59)	uInt16Number
10 to 11	2	Number of seconds (0 to 59)	uInt16Number

All the dateTimeNumber values in a profile shall be in Coordinated Universal Time (UTC, also known as GMT or ZULU Time). Profile writers are required to convert local time to UTC when setting these values. Programs that display these values may show the dateTimeNumber as UTC, show the equivalent local time (at current locale), or display both UTC and local versions of the dateTimeNumber.

4.3 float32Number

A float32Number shall be single-precision 32-bit floating-point number as specified in IEEE 754, excluding un-normalized numbers, infinities, and “not a number” (NaN) values.

NOTE 1 A 32-bit IEEE 754 floating-point number has an 8-bit exponent and a 23-bit mantissa.

NOTE 2 Although un-normalized numbers, infinities and NaN values are not stored in the ICC Profile, such values can occur as a result of CMM computations.

4.4 positionNumber

Positions of some data elements are indicated using a position offset with the data element’s size. This data type allows this information to be stored as a single entity. Table 2 shows the positionNumber encoding.

Table 2 — positionNumber encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	Offset to data element in bytes	uInt32Number
4 to 7	4	Size of data element in bytes	uInt32Number

4.5 response16Number

A response16Number is an 8-byte value, used to associate a normalized device code with a measurement value, where byte usage shall be assigned as specified in Table 3.

Table 3 — response16Number

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 1	2	16-bit number in the interval [DeviceMin to DeviceMax][a]	uInt16Number
2 to 3	2	Reserved, shall be zero	
4 to 7	4	Measurement value	s15Fixed16Number

[a] DeviceMin is encoded as 0000h and DeviceMax is encoded as FFFFh.

4.6 s15Fixed16Number

An s15Fixed16Number is a fixed signed 4-byte (32-bit) quantity which has 16 fractional bits as shown in Table 4.

Table 4 — s15Fixed16Number

Number	**Encoded as**
−32 768,0	80000000h
0	00000000h
1,0	00010000h
32 767 + (65 535/65 536)	7FFFFFFFh

4.7 u16Fixed16Number

A u16Fixed16Number is a fixed unsigned 4-byte (32-bit) quantity having 16 fractional bits as shown in Table 5.

Table 5 — u16Fixed16Number

Number	Encoded as
0	00000000h
1,0	00010000h
65 535 + (65 535/65 536)	FFFFFFFFh

4.8 u1Fixed15Number

A u1Fixed15Number is a fixed unsigned 2-byte (16-bit) quantity having15 fractional bits as shown in Table 6.

Table 6 — u1Fixed15Number

Number	Encoded as
0	0000h
1,0	8000h
1 + (32 767/32 768)	FFFFh

4.9 u8Fixed8Number

A u8Fixed8Numberfixed is an unsigned 2-byte (16-bit) quantity having 8 fractional bits as shown in Table 7.

Table 7 — u8Fixed8Number

Number	Encoded as
0	0000h
1,0	0100h
255 + (255/256)	FFFFh

4.10 uInt16Number

A uInt16Number is an unsigned 2-byte (16-bit) integer.

4.11 uInt32Number

A uInt32Number is an unsigned 4-byte (32-bit) integer.

4.12 uInt64Number

A uInt64Number is an unsigned 8-byte (64-bit) integer.

4.13 uInt8Number

A uInt8Number is an unsigned 1-byte (8-bit) integer.

4.14 XYZNumber

An XYZNumber is a set of three fixed signed 4-byte (32-bit) quantities used to encode CIEXYZ, nCIEXYZ, and PCSXYZ tristimulus values where byte usage is assigned as specified in Table 8. Although the CIE specifies that for reflecting and transmitting media Y should be normalized such that it has the value 100,0 for the perfect diffusing reflector or transmitter, in this part of ISO 15076, for reasons of coding efficiency, Y is specified such that it has the value 1,0 for the perfect diffusing reflector or transmitter for nCIEXYZ.

NOTE 1 Signed numbers are employed for this type to accommodate negative values arising during calculations.

Table 8 — XYZNumber

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	CIEXYZ X, nCIEXYZ X, or PCSXYZ X	s15Fixed16Number
4 to 7	4	CIEXYZ Y, nCIEXYZ Y, or PCSXYZ Y	s15Fixed16Number
8 to 11	4	CIEXYZ Z, nCIEXYZ Z, or PCSXYZ Z	s15Fixed16Number

4.15 Seven-bit ASCII

Alpha-numeric values, and other input and output codes, shall conform to the American Standard Code for Information Interchange (ASCII) specified in ISO/IEC 646.

5 Conformance

Any colour management system, application, utility or device driver that claims conformance with this part of ISO 15076 shall have the ability to read the profiles as they are defined in this part of ISO 15076. Any profile-generating software and/or hardware that claims conformance with this part of ISO 15076 shall have the ability to create profiles as they are defined in this part of ISO 15076. ICC conforming software shall use the ICC profiles in an appropriate manner.

This part of ISO 15076 requires that signatures for CMM type, device manufacturer, device model, profile tags and profile tag types shall be registered to ensure that all profile data is uniquely defined. The registration authority for these data is the ICC Technical Secretary.

NOTE See the ICC Web Site (www.color.org) for contact information.

6 Profile connection space, rendering intents, and device encoding

6.1 General considerations

The PCS is the reference colour space in which colours are encoded in order to provide an interface for connecting source and destination transforms. The PCS values constitute an encoding of a CIE colorimetric specification.

Four rendering intents are specified in this part of ISO 15076:

a) ICC-absolute colorimetric;

b) media-relative colorimetric;

c) perceptual;

d) saturation.

Each represents a type of colour rendering (mapping of colour values) that is useful for various imaging workflows. The colorimetric intents preserve the colorimetry of in-gamut colours at the expense of out-of-gamut colours. The mapping of out-of-gamut colours is not specified but should be consistent with the intended use of the transform. The perceptual and saturation rendering intents modify colorimetric values to account for any differences between devices, media, and viewing conditions.

The requirements for these rendering intents are given in 6.2 and discussed further in Annex D.

Profiles are required to contain transformations for one, or more, of these rendering intents. Clause 6 specifies which rendering intents are required, and which are optional, for the various classes of profiles.

NOTE When present, the colorimetricIntentImageStateTag indicates the image state of the colorimetry produced by the colorimetric intent transforms.

6.2 Rendering intents

6.2.1 General

The colorimetric rendering intents operate on measurement-based colorimetric values as adapted to the PCS adopted white chromaticity. This adaptation, when required, shall be indicated in the chromaticAdaptationTag. For the purposes of this part of ISO 15076, chromatic adaptation should be calculated using the linear Bradford model. Details of this model are provided in Annex E.

NOTE 1 The original measurement values can be determined from PCS values by applying the inverse chromatic adaptation transformation. This inversion is not usually performed. Since the PCS values are already adapted to the PCS adopted white chromaticity when constructing the profiles, neither the forward nor the inverse chromatic adaptation transforms need to be applied by the CMM in normal use of the profiles.

For the other intents transformations shall be assumed to be specified relative to the PCS illuminant. However, for these transformations profiles are not required to specify any chromatic adaptation that may have been employed in the calculation of the transformation data.

In transforms for the media-relative and ICC-absolute colorimetric intents, for Input profiles the PCS values may represent a colour rendering of the actual original captured. Likewise for Output profiles, the PCS values may be colour rendered by the output device to the actual medium. However, wherever ICC profiles are used, unless the colorimetric intent image state tag is present, the PCS values resulting from such transforms shall be interpreted as the colorimetry of the original and reproduction, regardless of whether such colorimetry is the actual colorimetry. If the colorimetric intent image state tag is present, then the interpretation of the colorimetry of the resulting PCS values shall be according to the value recorded in that tag.

NOTE 2 All PCS colorimetry is picture-referred unless otherwise indicated in the colorimetric intent image state tag.

6.2.2 Media-relative colorimetric intents

Transformations for this intent shall re-scale 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) as defined in 6.3.2.

NOTE Transforms for the media-relative colorimetric intent represent media-relative measurements of the captured original (for Input profiles), or media-relative colour reproductions produced by the output device (for Output profiles) unless otherwise indicated in the colorimetricIntentImageStateTag.

6.2.3 ICC-absolute colorimetric intent

Transformations for this intent shall leave the chromatically adapted nCIEXYZ tristimulus values of the in-gamut colours unchanged.

The DToB3 and BToD3 tags contain separate transforms for the ICC-absolute colorimetric intent. This allows for an absolute expression of colorimetric data, limited only by the range of float32Number values. The mediaWhitePointTag is *not* used in processing DToB3 and BToD3 tags.

When DToB3 and BToD3 tags are not present (or not used), profiles do not contain a separate transform for the ICC-absolute colorimetric intent. When this intent is needed, it shall be generated, as described in 6.3.2, using the mediaWhitePointTag, which specifies the CIE 1931 XYZ tristimulus values of the white point of the actual medium, as represented in the PCS. In this way, ICC-absolute colorimetric rendering may be obtained by using the media-relative colorimetric intent transformations (AToB1, BToA1) for the source and destination profiles and scaling the PCS values by the ratio of the destination profile mediaWhitePointTag to the source profile mediaWhitePointTag (see Annex D for more information). As specified in 9.2.34, for monitor profiles the mediaWhitePointTag shall be set to the PCS white point (defined in 6.3.4.3). Likewise, if the viewer is assumed to completely adapt to the white point of the medium for any other media (i.e. the media white looks like a perfect reflecting diffuser) the mediaWhitePointTag should be set to the PCS white point.

NOTE 1 Transforms for the ICC-absolute colorimetric intent represent measurements of the captured original relative to a hypothetical perfectly reflecting or transmitting diffuser (for Input profiles), or colour reproductions produced by the output device relative to a hypothetical perfectly reflecting or transmitting diffuser (for Output profiles), unless otherwise indicated in the in the colorimetricIntentImageStateTag.

NOTE 2 This definition of ICC-absolute colorimetry is sometimes 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.

6.2.4 Perceptual intent

In perceptual transforms the PCS values represent hypothetical measurements of a colour reproduction on the reference reflective medium. By extension, for the perceptual intent, the PCS represents the appearance of that reproduction as viewed in the reference viewing environment by a human observer adapted to that environment. The exact colour rendering of the perceptual intent is vendor specific.

NOTE 1 The reference medium and viewing environment are defined in 6.3.3.

NOTE 2 The perceptual intent is useful when it is not required to exactly maintain image colorimetry (such as with natural images), with input and output media that are substantially different.

NOTE 3 When using the perceptual intent, the colour rendering to the reference medium serves to ensure that Input and Output profiles from different manufacturers will work reasonably well together, although the results from different combinations of profiles will likely be different due to the proprietary nature of the colour rendering contained in this intent.

6.2.5 Saturation intent

The exact colour rendering 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 colours.

6.3 Profile connection space

6.3.1 Chromatic adaptation

The PCS adopted white chromaticity shall be the chromaticity of the D50 illuminant defined in ISO 3664.

6.3.2 Colorimetric specification

6.3.2.1 General

The measurement parameters for the PCS, and all other colour spaces defined in this part of ISO 15076, shall be based on ISO 13655. The colorimetry shall be assumed not to contain any flare or other defect caused by inadequacies in the optical system of the instrument and illumination used to make the measurements, but shall be assumed to include the surface reflection component normally associated with the prescribed measurement geometry.

The PCS colour space encodings shall be based on media-relative colorimetry in which tristimulus values are relative to the values in the mediaWhitePointTag. For the perceptual rendering intent the medium for calculation of the media-relative colorimetry shall be the reference medium defined in 6.3.3.

6.3.2.2 Translation between media-relative colorimetric data and ICC-absolute colorimetric data

The translation from ICC-absolute colorimetric data to media-relative colorimetry data is given by Equations (1) to (3).

$$X_\mathrm{r} = \left[\frac{X_\mathrm{i}}{X_\mathrm{mw}}\right] X_\mathrm{a} \tag{1}$$

$$Y_\mathrm{r} = \left[\frac{Y_\mathrm{i}}{Y_\mathrm{mw}}\right] Y_\mathrm{a} \tag{2}$$

$$Z_\mathrm{r} = \left[\frac{Z_\mathrm{i}}{Z_\mathrm{mw}}\right] Z_\mathrm{a} \tag{3}$$

where

X_r, Y_r, Z_r are the media-relative colorimetric data (i.e. PCSXYZ);

X_a, Y_a, Z_a are the ICC-absolute colorimetric data (i.e. nCIEXYZ);

X_mw, Y_mw, Z_mw are the nCIEXYZ values of the media white point as specified in the mediaWhitePointTag;

X_i, Y_i, Z_i are the PCSXYZ values of the PCS white point defined in 6.3.4.3.

The translation from media-relative colorimetry data to ICC-absolute colorimetric data is given by Equations (4) to (6):

$$X_\mathrm{a} = \left[\frac{X_\mathrm{mw}}{X_\mathrm{i}}\right] X_\mathrm{r} \tag{4}$$

$$Y_\mathrm{a} = \left[\frac{Y_\mathrm{mw}}{Y_\mathrm{i}}\right] Y_\mathrm{r} \tag{5}$$

$$Z_\mathrm{a} = \left[\frac{Z_\mathrm{mw}}{Z_\mathrm{i}}\right] Z_\mathrm{r} \tag{6}$$

where

X, Y, Z are the CIE tristimulus values;

X_n, Y_n, Z_n are the CIE values of the illuminant white point.

6.3.2.3 Computation of PCSLAB

When values are encoded as PCSLAB, these shall be computed from the PCSXYZ tristimulus values as specified in Annex A, noting that:

$$\frac{X}{X_\mathrm{n}} \text{ is replaced by } \frac{X_\mathrm{r}}{X_\mathrm{i}} \text{ (or } \frac{X_\mathrm{a}}{X_\mathrm{mw}}\text{)} \tag{7}$$

$$\frac{Y}{Y_\mathrm{n}} \quad \text{is replaced by} \quad \frac{Y_\mathrm{r}}{Y_\mathrm{i}} \quad \left(\text{or } \frac{Y_\mathrm{a}}{Y_\mathrm{mw}}\right) \tag{8}$$

$$\frac{Z}{Z_\mathrm{n}} \quad \text{is replaced by} \quad \frac{Z_\mathrm{r}}{Z_\mathrm{i}} \quad \left(\text{or } \frac{Z_\mathrm{a}}{Z_\mathrm{mw}}\right) \tag{9}$$

6.3.3 Reference viewing environment and medium for the perceptual rendering intent

6.3.3.1 General

Because perceptual rendering generally involves mapping the colours of a source to be well-suited for a destination medium (i.e. colour rendering and/or colour re-rendering), it is desirable that the perceptual intent PCS reference medium (PRM) and associated viewing conditions be well-defined. Then, the source profile can perceptually render from the source to the PRM, and the destination profile can perceptually render from the PRM to the destination medium. The PRM, in the PCS, serves as the common intermediate representation. Well-defined viewing conditions are required because they will affect the appearance of colour content represented on the PRM.

Perceptual rendering remains a proprietary art, due both to the current state of perceptual rendering algorithms, and also to the fact that viewer and application specific preferences can affect the nature of a desired reproduction (when exact colour matching is not the objective). It is not practical or desirable to specify standard perceptual rendering algorithms. Consequently, it is also not practical or desirable to require that perceptual rendering intents match an exact perceptual intent reference medium gamut (PRMG). Gamut mapping could be applied to clip the results of a perceptual rendering algorithm to a specific target gamut, but that would result in a loss of information and invertiblility. Therefore, the reference medium white point, black point, and viewing conditions attributes of the PRM are defined precisely, and the PRM gamut is defined to be a fuzzy target that can be used as the aim of perceptual rendering transforms, but does not have to be exactly matched.

6.3.3.2 Perceptual intent reference medium (PRM) characteristics

The reference medium is defined as a hypothetical print on a substrate specified to have a neutral reflectance of 89 %. The darkest printable colour on this medium is assumed to have a neutral reflectance of 0,309 11 %, which is 0,347 31 % of the substrate reflectance. These shall be assumed to be the white point and black point of the reference medium respectively.

NOTE The reference medium therefore has a linear dynamic range of 287,9 : 1 and a density range of 2,459 3.

6.3.3.3 Perceptual intent reference medium gamut (PRMG)

Perceptual rendering intent and saturation rendering intent transforms may optionally use the reflection colour gamut specified in ISO 12640-3, and provided in Table 9, as the PRMG. This colour gamut boundary description is intended only as an optional target for perceptual and saturation rendering intents, and these transforms should not clip to it. It is provided to enable improved interoperability of perceptual and saturation transforms.

Table 9 — Maximum chroma values C^*_{ab}

h_{ab} [°]	Maximum C^*_{ab} for $L^*=$																				
	3,1373	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
0	0	11	26	39	52	64	74	83	91	92	91	87	82	75	67	57	47	37	25	13	0
10	0	10	24	38	50	62	73	82	90	92	91	87	82	75	67	58	48	37	26	13	0
20	0	10	23	37	50	62	73	84	93	94	94	90	85	78	70	60	50	39	27	14	0
30	0	9	22	35	48	61	74	86	98	100	101	96	90	83	75	65	54	42	30	15	0
40	0	8	21	34	47	60	73	83	93	97	101	99	97	90	83	73	61	47	34	17	0
50	0	8	20	32	43	55	66	77	88	95	99	101	100	98	92	85	72	56	40	20	0
60	0	7	17	27	37	47	57	67	76	84	91	96	100	102	103	98	90	72	51	26	0
70	0	6	16	25	34	43	52	60	68	76	83	90	96	100	104	107	109	100	74	37	0
80	0	6	15	23	32	40	48	57	64	71	78	85	91	97	103	107	110	113	110	70	0
90	0	6	14	22	30	39	47	55	62	68	75	82	88	95	101	106	112	117	120	123	0
100	0	6	14	22	30	38	46	54	61	68	74	81	88	94	100	106	109	112	112	92	0
110	0	6	14	22	31	39	47	55	63	69	76	83	89	96	100	103	106	107	102	75	0
120	0	6	15	24	32	41	49	58	66	73	80	87	93	98	101	102	99	91	73	50	0
130	0	6	16	25	35	44	54	63	72	80	87	93	97	101	99	94	86	73	56	34	0
140	0	7	18	28	38	48	57	67	77	86	95	98	101	97	93	85	75	61	44	26	0
150	0	7	19	30	40	51	62	72	83	92	97	99	96	91	85	76	66	52	37	22	0
160	0	7	20	32	44	56	68	80	92	96	99	97	92	87	79	70	59	46	33	19	0
170	0	8	20	32	43	53	64	75	85	91	96	93	89	82	75	65	55	42	30	17	0
180	0	8	20	31	41	52	62	72	81	87	92	90	86	79	71	61	52	40	28	15	0
190	0	8	20	30	40	50	60	68	76	82	87	85	82	76	69	60	50	39	27	14	0
200	0	8	20	30	38	47	56	63	70	76	82	81	77	72	66	58	49	38	27	14	0
210	0	8	20	29	37	46	53	60	66	73	79	80	75	70	64	57	49	38	27	14	0
220	0	8	20	29	37	45	52	59	65	71	76	75	72	68	63	56	48	38	27	14	0
230	0	9	20	29	38	46	53	59	65	70	75	73	71	66	61	54	46	36	26	13	0
240	0	10	22	31	40	48	55	61	67	71	74	70	66	61	56	49	41	32	23	12	0
250	0	11	24	34	43	51	59	65	70	73	71	68	63	58	52	45	38	30	21	11	0
260	0	14	27	38	48	57	64	69	73	73	70	66	61	56	50	43	35	28	20	10	0
270	0	17	32	45	55	65	70	75	75	73	70	66	61	55	49	42	34	27	19	10	0
280	0	21	42	55	68	75	81	80	79	76	72	67	61	55	49	41	34	26	18	9	0
290	0	26	52	68	83	86	89	87	84	80	75	69	63	57	50	42	35	27	18	10	0
300	0	25	69	82	95	94	93	91	88	85	79	73	66	59	52	44	36	28	19	10	0
310	0	21	51	74	91	97	100	98	95	90	84	77	70	63	55	47	39	30	20	10	0
320	0	18	41	62	79	91	102	101	98	95	89	83	76	68	60	51	42	32	22	11	0
330	0	16	35	53	71	82	91	100	104	102	98	91	84	76	67	57	47	36	24	12	0
340	0	14	31	46	61	73	83	92	101	103	99	95	89	80	71	61	50	38	26	13	0
350	0	12	28	42	55	68	77	86	94	96	93	90	85	77	68	58	48	37	25	13	0

The L^*, C^* and h values specified in this table are relative to the reference medium white with a reflectance factor of 0,89. To calculate values relative to a perfect reflecting diffuser, it is necessary to convert the table values to X, Y and Z values, scale the resulting X, Y and Z values by a factor of 0,89, and convert back to L^*, C^* and h relative to the perfect reflecting diffuser.

The values in this table are calculated for illuminant D50 and the CIE 1931, 2°, standard observer.

NOTE If the resolution of the data needs to be finer it would normally be adequate to obtain it by linear interpolation of the quoted data.

When the PRMG is used as the target colour gamut for perceptual and saturation transforms, this should be indicated with the corresponding profile tags as described in 9.2.37 and in 9.2.46.

Furthermore, as several popular source-to-destination colour re-rendering algorithms are defined via transformations of primaries and secondaries, approximate locations for red, green, blue, cyan, magenta and yellow in the PRMG have also been specified. These coordinates are provided in Table 10.

Table 10 — PRMG "primary and secondary" colours

Colour coordinate	Red	Yellow	Green	Cyan	Blue	Magenta
L^*_{ab}	41	95	60	50	21	42
C^*_{ab}	98	123	100	76	95	102
h_{ab}	29	90	140	220	300	340

6.3.3.4 Perceptual intent reference medium viewing conditions

The reference viewing environment shall be based on standard viewing condition P2, as specified for graphic arts and photography in ISO 3664, but extended to include an "average" surround, i.e. the illumination of the image shall be assumed to be similar to the illumination of the rest of the environment. The surfaces immediately surrounding the image shall be assumed to be a uniform matt grey with a reflectance of 20 %. The reference viewing environment shall also be assumed to have a level of viewing flare of 0,007 5 (3/4 %) of the luminance of the reference medium in the reference viewing environment (1,06 cd/m^2). If the actual viewing environment differs from the reference viewing environment perceptual transforms shall be used to compensate for the difference in viewing environments.

NOTE ISO 3664 describes the appropriate illumination level for practical appraisal of prints as 500 lx (P2), which is specified to be typical of the level found in actual home and office viewing environments. This was deemed to be most appropriate for the reference viewing environment.

6.3.4 Colour space encodings for the PCS

6.3.4.1 General

The colorimetric data defined in 6.3.2 and 6.3.3 may be specified either as PCSXYZ or PCSLAB data. When specified as PCSXYZ data they shall be encoded using 16 bits per component while when specified as PCSLAB data they shall be encoded as either 8 bits per component or 16 bits per component. Additionally, within the context of DToBx and BToDx tags PCSXYZ or PCSLAB data shall be encoded using float32Number values that directly express PCSXYZ or PCSLAB colorimetry.

When converting from float32Number-based to integer-based encoding, component-wise clipping shall be applied if the floating-point value is outside the range that can be encoded as integers.

NOTE 1 These alternative methods are provided in order to satisfy conflicting requirements for accuracy and storage space. The profile header specifies which encoding method has been used. While supporting multiple encodings increases the complexity of colour management, it provides flexibility in addressing different user requirements such as colour accuracy and memory footprint.

NOTE 2 It is important to understand that the PCS encodings do not represent a quantization of the connection space. The purpose of the encodings is to allow points within the space to be specified. Since the processing models benefit from interpolation between table entries, the interpolated AToB results need to be used as the inputs to the BToA transforms. The AToB results are not rounded to the nearest encoding value. (AToB and BToA transforms are defined in 10.10 and 10.11)

6.3.4.2 General PCS encoding

For the 16-bit integer based PCSXYZ encoding, each component (PCSXYZ X, PCSXYZ Y, and PCSXYZ Z) is encoded as a u1Fixed15Number. This encoding was chosen to allow for PCSXYZ values that have an X or Z greater than 1,0.

For the XYZNumber based PCSXYZ encoding, each component (PCSXYZ X, PCSXYZ Y, and PCSXYZ Z) is encoded as a s15Fixed16Number. For the float32Number-based PCS encodings the actual PCSXYZ or PCSLAB values are directly encoded. The relationship between PCSXYZ encodings is shown in Table 11.

Table 11 — PCSXYZ X, Y or Z encoding

Value (X, Y or Z)	u1Fixed15Number	s15Fixed16Number	float32Number
0,0	0000h	00000000h	**0,0**
1,0	8000h	00010000h	**1,0**
1,0 + (32 767,0 / 32 768,0)	FFFFh	0001FFFEh	1,999 969 482 421 875
NOTE 1,999 969 482 421 875 is 1,0 + 32 767,0 / 32 768,0 expressed as a float32Number.			

For the PCSLAB encodings, the PCSLAB L^* values have a different encoding than the PCSLAB a^* and PCSLAB b^* values. The PCSLAB L^* encoding is shown in Table 12.

Table 12 — PCSLAB L^* encoding

Value (PCSLAB L^*)	uInt8Number	uInt16Number	float32Number
0	00h	0000h	0,0
100,0	FFh	FFFFh	100,0

The PCSLAB a^* and PCSLAB b^* encoding is shown in Table 13.

Table 13 — PCSLAB a^* or PCSLAB b^* encoding

Value (PCSLAB a^* or PCSLAB b^*)	uInt8Number	uInt16Number	float32Number
−128,0	00h	0000h	−128,0
0	80h	8080h	0,0
127,0	FFh	FFFFh	127,0

NOTE 1 The integer encoding is not “two’s complement” encoding, but a linear scaling after an offset of 128. This encoding was chosen to prevent discontinuities in CLUTs when going from negative to positive values.

NOTE 2 It is possible to convert between the 8-bit and 16-bit encodings by multiplying or dividing by 257. (See A.4.)

NOTE 3 Both the lut16Type and the namedColor2Type tag types (and *only* those tag types) use a legacy 16-bit encoding of PCSLAB L^*, PCSLAB a^* and PCSLAB b^* which is retained for backwards compatibility with an earlier profile version (version 2). To avoid confusion this encoding is specified in 10.8 “lut16Type”.

6.3.4.3 PCS encodings for white and black

In transforms for the media-relative colorimetric, perceptual, and saturation rendering intents (all intents other than ICC-absolute colorimetric), the white point of the medium is represented in PCSXYZ and PCSLAB formats as shown in Table 14.

NOTE For the perceptual intent, the medium is the perceptual reference medium.

Table 14 — Encodings of PCS white point

Component	Value	8-bit encoding	16-bit encoding
PCSLAB L^*	100	FFh	FFFFh
PCSLAB a^*	0	80h	8080h
PCSLAB b^*	0	80h	8080h
PCSXYZ X	0,964 2	—	7B6Bh
PCSXYZ Y	1,000 0	—	8000h
PCSXYZ Z	0,824 9	—	6996h

In transforms for all intents the perfect absorber (a theoretical medium that reflects absolutely no light) is represented in PCSXYZ and PCSLAB formats as shown in Table 15. Other reflectance values are mapped linearly to PCSXYZ.

Table 15 — Perfect absorber encodings

Component	Value	8-bit encoding	16-bit encoding
PCSLAB L^*	0,0	00h	0000h
PCSLAB a^*	0,0	80h	8080h
PCSLAB b^*	0,0	80h	8080h
PCSXYZ X	0,0	—	0000h
PCSXYZ Y	0,0	—	0000h
PCSXYZ Z	0,0	—	0000h

In transforms for the perceptual intent, the black point of the perceptual reference medium is defined in Table 16 in PCSLAB and PCSXYZ.

Table 16 — Encodings of the perceptual reference medium black point

Component	Value	8-bit encoding	16-bit encoding
PCSLAB L^*	3,137 3	08h	0808h
PCSLAB a^*	0,0	80h	8080h
PCSLAB b^*	0,0	80h	8080h
PCSXYZ X	0,003 357	—	006Eh
PCSXYZ Y	0,003 479	—	0072h
PCSXYZ Z	0,002 869	—	005Eh
NOTE Due to limited numerical precision, PCSXYZ Y might not exactly match PCSLAB L^*.			

6.4 Converting between PCSXYZ and PCSLAB encodings

Conversions between the PCSXYZ and PCSLAB encodings shall use the equations of the form specified in ISO 13655 with appropriate adjustments for the range.

When converting to integer-based encodings, any colours in the PCSXYZ encoding range that are outside of the PCSLAB encoding range shall be clipped on a per-component basis to the outside limits of the range of PCSLAB when transforming from PCSXYZ into PCSLAB. Conversely, any colours that occur in the PCSLAB encoding range that are outside of the encoding range of PCSXYZ shall be clipped on a per-component basis to the PCSXYZ range when transforming from PCSLAB into PCSXYZ.

When converting to float32Number-based encodings, conversion between PCSXYZ and PCSLAB is performed and encoded using the float32Number encoding of PCS values as defined in 6.3.4.2. No clipping is performed. In order to calculate PCSLAB values from negative PCSXYZ values, the straight line portion of the PCSLAB colour component transfer function below 0,008 856 shall be extended linearly below zero.

6.5 Device encoding

The specification of device value encoding is determined by the device. Normally, device values in the range of 0,0 to 1,0 are encoded using a 0 to 255 (FFh) range when using 8 bits and are encoded using a 0 to 65 535 (FFFFh) range when using 16 bits. When encoding using float32Number values in DToBx and BToDx tags, device values may be outside the 0,0 to 1,0 range.

7 Profile requirements

7.1 General

7.1.1 An ICC profile shall include the following elements, in the order shown, as a single file:

a) a 128-byte profile header as defined in 7.2;

b) a profile tag table as defined in 7.3;

c) a profile tagged element data as defined in 7.4.

This is illustrated in Figure 6.

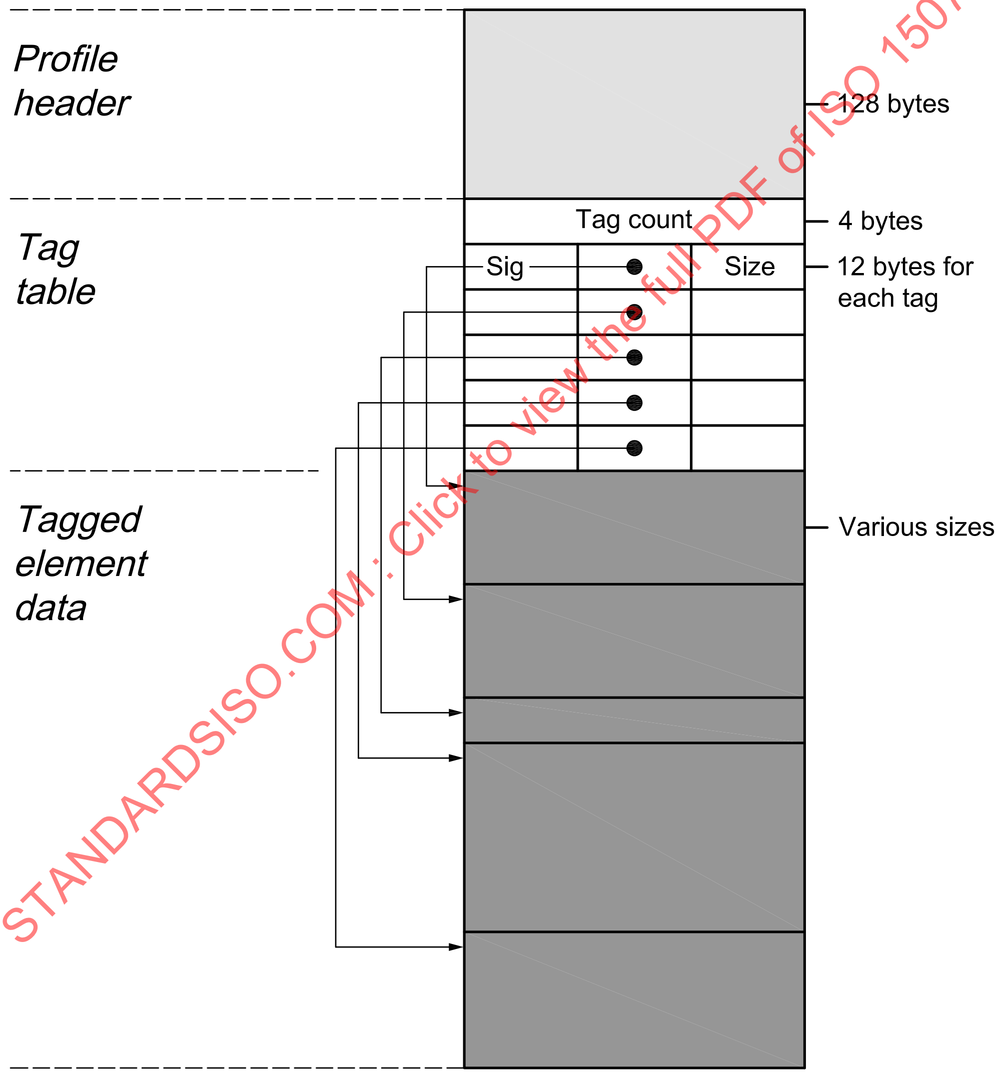


Figure 6 — Profile structure

7.1.2 The required tags for each profile type are tabulated in Clause 8. The definition of all publicly available tags and their signatures is contained in Clause 9 along with the permitted tag types for each tag. Tag types are defined in Clause 10.

Within the profile structure:

a) all profile data shall be encoded as big-endian;

b) the first set of tagged element data shall immediately follow the tag table;

c) all tagged element data, including the last, shall be padded by no more than three following pad bytes to reach a 4-byte boundary;

d) all pad bytes shall be NULL (as defined in ISO/IEC 646, character 0/0).

NOTE 1 This implies that the length is required to be a multiple of four.

NOTE 2 The above restrictions result in two key benefits. First, the likelihood of two profiles which contain the same tag data, yet have different checksum values, is reduced. Second, all profiles are reduced to a minimum size.

7.2 Profile header

7.2.1 General requirements

The profile header provides the necessary information to allow a receiving system to properly search and sort ICC profiles. The profile header is 128 bytes in length and contains 18 fields. Table 17 gives the byte position, field length, and content of each element in the profile header. The encoding of the field contents shall be as defined in 7.2.2 to 7.2.19.

NOTE 1 Having a fixed length header allows for performance enhancements in profile searching and sorting applications.

NOTE 2 For ColorSpace and Abstract profiles (see 8.7 and 8.8) some of these fields are not relevant and can be set to zero.

Table 17 — Profile header fields

Byte position	Field length bytes	Field contents	Encoded as
0 to 3	4	Profile size	UInt32Number
4 to 7	4	Preferred CMM type	See 7.2.3
8 to 11	4	Profile version number	See 7.2.4
12 to 15	4	Profile/Device class	See 7.2.5
16 to 19	4	Colour space of data (possibly a derived space)	See 7.2.6
20 to 23	4	PCS	See 7.2.7
24 to 35	12	Date and time this profile was first created	dateTimeNumber
36 to 39	4	‘acsp’ (61637370h) profile file signature	See 7.2.9
40 to 43	4	Primary platform signature	See 7.2.10
44 to 47	4	Profile flags to indicate various options for the CMM such as distributed processing and caching options	See 7.2.11
48 to 51	4	Device manufacturer of the device for which this profile is created	See 7.2.12
52 to 55	4	Device model of the device for which this profile is created	See 7.2.13
56 to 63	8	Device attributes unique to the particular device setup such as media type	See 7.2.14
64 to 67	4	Rendering Intent	See 7.2.15
68 to 79	12	The nCIEXYZ values of the illuminant of the PCS	XYZNumber
80 to 83	4	Profile creator signature	See 7.2.17
84 to 99	16	Profile ID	See 7.2.18
100 to 127	28	Bytes reserved for future expansion and shall be set to zero (00h)	

7.2.2 Profile size field (bytes 0 to 3)

The value in the profile size field shall be the exact size obtained by combining the profile header, the tag table, and the tagged element data, including the pad bytes for the last tag. It shall be encoded as a uInt32Number.

7.2.3 Preferred CMM type field (bytes 4 to 7)

This field may be used to identify the preferred CMM to be used. If used, it shall match a CMM type signature registered in the ICC registry (see Clause 5). If no preferred CMM is identified, this field shall be set to zero (00000000h).

7.2.4 Profile version field (bytes 8 to 11)

The profile version with which the profile is compliant shall be encoded as binary-coded decimal in the profile version field. The first byte (byte 8) shall identify the major version and byte 9 shall identify the minor version and bug fix version in each 4-bit half of the byte. Bytes 10 and 11 are reserved and shall be set to zero. The major and minor versions are set by the International Color Consortium. The profile version number consistent with this part of ISO 15076 is “4.3.0.0” (encoded as 04300000h).

NOTE A major version number change occurs only when changes made to the specification require that both CMMs and profile generating software be upgraded in order to correctly use or produce profiles conforming to the revised specification. A minor version number change will occur when profiles conforming to the revised specification can be processed by existing CMMs. For example, adding a required tag would require a major revision to the specification, whereas adding an optional tag would only require a minor revision.

7.2.5 Profile/device class field (bytes 12 to15)

This field shall contain one of the profile class signatures shown in Table 18.

There are three basic classes of device profiles, which are Input, Display and Output. In addition to the three basic device profile classes, four additional colour processing profiles are defined. These profiles provide a standard implementation for use by the CMM in general colour processing, or for the convenience of CMMs which may use these types to store calculated transforms. These four additional profile classes are DeviceLink, ColorSpace, Abstract and NamedColor.

Table 18 — Profile classes

Profile class	Signature	Hex encoding
Input device profile	‘scnr’	73636E72h
Display device profile	‘mntr’	6D6E7472h
Output device profile	‘prtr’	70727472h
DeviceLink profile	‘link’	6C696E6Bh
ColorSpace profile	‘spac’	73706163h
Abstract profile	‘abst’	61627374h
NamedColor profile	‘nmcl’	6E6D636Ch

7.2.6 Data colour space field (bytes 16 to 20)

This field shall contain the signature of the data colour space expected on the A side (device side) of the profile transforms. The names and signatures of the permitted data colour spaces are shown in Table 19. Signatures are left justified.

Table 19 — Data colour space signatures

Colour space type	Signature	Hex encoding
nCIEXYZ or PCSXYZ[a]	‘XYZ ’	58595A20h
CIELAB or PCSLAB[b]	‘Lab ’	4C616220h
CIELUV	‘Luv ’	4C757620h
YCbCr	‘YCbr’	59436272h
CIEYxy	‘Yxy ’	59787920h
RGB	‘RGB ’	52474220h
Gray	‘GRAY’	47524159h
HSV	‘HSV ’	48535620h
HLS	‘HLS ’	484C5320h
CMYK	‘CMYK’	434D594Bh
CMY	‘CMY ’	434D5920h
2 colour	‘2CLR’	32434C52h
3 colour (other than those listed above)	‘3CLR’	33434C52h
4 colour (other than CMYK)	‘4CLR’	34434C52h
5 colour	‘5CLR’	35434C52h
6 colour	‘6CLR’	36434C52h
7 colour	‘7CLR’	37434C52h
8 colour	‘8CLR’	38434C52h
9 colour	‘9CLR’	39434C52h
10 colour	‘ACLR’	41434C52h
11 colour	‘BCLR’	42434C52h
12 colour	‘CCLR’	43434C52h
13 colour	‘DCLR’	44434C52h
14 colour	‘ECLR’	45434C52h
15 colour	‘FCLR’	46434C52h

[a] The signature 'XYZ' refers to nCIEXYZ or PCSXYZ depending upon the context.

[b] The signature 'Lab' refers to CIELAB or PCSLAB depending upon the context.

7.2.7 PCS field (bytes 20 to 23)

For all profile classes (see Table 18), other than a DeviceLink profile, the PCS encoding shall be either PCSXYZ or PCSLAB and the signature shall be as defined in Table 19. When the profile/device class is a DeviceLink profile, the value of the PCS shall be the appropriate data colour space from Table 19. The field represents the colour space on the B side (PCS side) of the transform.

7.2.8 Date and time field (bytes 24 to 35)

This header field shall contain the date and time that the profile was first created, encoded as a dateTimeNumber.

7.2.9 Profile file signature field (bytes 36 to 39)

The profile file signature field shall contain the value “acsp” (61637379h) as a profile file signature.

7.2.10 Primary platform field (bytes 40 to 43)

This field may be used to identify the primary platform/operating system framework for which the profile was created. The primary platforms that have been identified, and the signatures that shall be used are shown in Table 20. If there is no primary platform identified, this field shall be set to zero (00000000h).

Table 20 — Primary platforms

Primary platform	Signature	Hex encoding
Apple Computer, Inc.	‘APPL’	4150504Ch
Microsoft Corporation	‘MSFT’	4D534654h
Silicon Graphics, Inc.	‘SGI ’	53474920h
Sun Microsystems, Inc.	‘SUNW’	53554E57h

7.2.11 Profile flags field (bytes 44 to 47)

The profile flags field shall contain flags to indicate various hints for the CMM such as distributed processing and caching options. The least-significant 16 bits are reserved for the ICC. Flags in bit positions 0 and 1 shall be used as indicated in Table 21. Annex B describes embedding device profiles within EPS, TIFF, JFIF and GIF image files.

Table 21 — Profile flags

Bit position	Field length bits	Field contents
0	1	Embedded profile (0 if not embedded, 1 if embedded in file)
1	1	Profile cannot be used independently of the embedded colour data (set to 1 if true, 0 if false)

7.2.12 Device manufacturer field (bytes 48 to 51)

This field may be used to identify a device manufacturer. If used the signature shall match the signature contained in the appropriate section of the ICC signature registry found at www.color.org (see Clause 5). If not used this field shall be set to zero (00000000h).

7.2.13 Device model field (bytes 52 to 55)

This field may be used to identify a device model. If used the signature shall match the signature contained in the appropriate section of the ICC signature registry found at www.color.org (see Clause 5). If not used this field shall be set to zero (00000000h).

7.2.14 Device attributes field (bytes 56 to 63)

The device attributes field shall contain flags used to identify attributes unique to the particular device setup for which the profile is applicable. The least-significant 32 bits of this 64-bit value are defined by the ICC. Bit usage shall be used as shown in Table 22.

Table 22 — Device attributes

Bit position	Field length bits	Attribute
0	1	Reflective (0) or transparency (1)
1	1	Glossy (0) or matte (1)
2	1	Media polarity, positive (0) or negative (1)
3	1	Colour media (0), black & white media (1)
4 to 31	28	Reserved (set to binary zero)
32 to 63	32	Use not defined by ICC (vendor specific)
NOTE Notice that bits 0, 1, 2, and 3 describe the media, not the device. For example, a profile for a colour scanner that has been loaded with black & white film will have bit 3 set on, regardless of the value in the data colour space field (see 7.2.6). If the media is not inherently "colour" or "black & white" (such as the paper in an inkjet printer), the reproduction takes on the property of the device. Thus, an inkjet printer loaded with a colour ink cartridge can be thought to have "colour" media.		

7.2.15 Rendering intent field (bytes 64 to 67)

The rendering intent field shall specify the rendering intent which should be used (or, in the case of a DeviceLink profile, was used) when this profile is (was) combined with another profile. In a sequence of more than two profiles, it applies to the combination of this profile and the next profile in the sequence and not to the entire sequence. Typically, the user or application will set the rendering intent dynamically at runtime or embedding time. Therefore, this flag may not have any meaning until the profile is used in some context, e.g. in a DeviceLink or an embedded source profile.

The field is a uInt32Number in which the least-significant 16 bits shall be used to encode the rendering intent. The most significant 16 bits shall be set to zero (0000h).

The defined rendering intents are perceptual, media-relative colorimetric, saturation and ICC-absolute colorimetric. These shall be identified using the values shown in Table 23.

Table 23 — Rendering intents

Rendering intent	Value
Perceptual	0
Media-relative colorimetric	1
Saturation	2
ICC-absolute colorimetric	3

7.2.16 PCS illuminant field (Bytes 68 to 79)

The PCS illuminant field shall contain the nCIEXYZ values $X = 0{,}964\ 2$, $Y = 1{,}0$ and $Z = 0{,}824\ 9$ encoded as an XYZNumber. See Annex A for further details.

NOTE These values are the nCIEXYZ values of CIE illuminant D50

7.2.17 Profile creator field (bytes 80 to 83)

This field may be used to identify the creator of the profile. If used the signature should match the signature contained in the device manufacturer section of the ICC signature registry found at www.color.org. If not used this field shall be set to zero (00000000h).

7.2.18 Profile ID field (bytes 84 to 99)

This field, if not zero (00h), shall hold the Profile ID. The Profile ID shall be calculated using the MD5 fingerprinting method as defined in Internet RFC 1321. The entire profile, whose length is given by the size field in the header, with the profile flags field (bytes 44 to 47, see 7.2.11), rendering intent field (bytes 64 to 67, see 7.2.15), and profile ID field (bytes 84 to 99) in the profile header temporarily set to zeros (00h), shall be used to calculate the ID. A profile ID field value of zero (00h) shall indicate that a profile ID has not been calculated.

It is suggested that profile creators compute and record a profile ID.

7.2.19 Reserved field (bytes 100 to 127)

This field of the profile header is reserved for future ICC definition and shall be set to zero.

7.3 Tag table

7.3.1 Overview

The tag table acts as a table of contents for the tags and an index into the tag data element in the profiles. It shall consist of a 4-byte entry that contains a count of the number of tags in the table followed by a series of 12-byte entries with one entry for each tag. The tag table therefore contains 4+12n bytes where n is the number of tags contained in the profile. The entries for the tags within the table are not required to be in any particular order nor are they required to match the sequence of tag data element within the profile.

Each 12-byte tag entry following the tag count shall consist of a 4-byte tag signature, a 4-byte offset to define the beginning of the tag data element, and a 4-byte entry identifying the length of the tag data element in bytes. Table 24 illustrates the structure for this tag table. 7.3.2 to 7.3.5 specify the position and content of the entries composing the tag table.

Table 24 — Tag table structure

Byte offset [a]	Field length bytes	Content	Encoded as
0 to 3	4	Tag count (n)	
4 to 7	4	Tag Signature	
8 to 11	4	Offset to beginning of tag data element	uInt32Number
12 to 15	4	Size of tag data element	uInt32Number
16 to (12n+3)	12(n–1)	Signature, offset and size respectively of subsequent n–1 tags	

n is the number of tags contained in the profile

[a] The byte offset shown in this table is with respect to the 128-byte header. Thus the tag table starts at byte position 128.

7.3.2 Tag count (byte position 0 to 3)

Byte positions 0 to 3 shall specify the number of tags contained in the tag table, encoded as a uInt32Number.

7.3.3 Tag signature (byte position 4 to 7 and repeating)

Byte positions 4 to 7 (and repeating at 12-byte intervals) shall specify the signature of a tag listed in Clause 10, or of a private tag. Signatures of private tags shall be registered with the ICC as defined in Clause 5.

7.3.4 Offset to beginning of tag data element (byte position 8 to 11 and repeating)

Byte positions 8 to 11 (and repeating at 12-byte intervals) shall specify the address of the beginning of the tag data element, with respect to the beginning of the profile data stream (which has an address of zero), encoded as a uInt32Number.

NOTE 1 For profiles that are not embedded, the number specified is the same as the file offset.

All tag data elements shall start on a 4-byte boundary (relative to the start of the profile data stream) and the two least-significant bits of each tag data offset shall be zero. This means that a tag starting with a 32-bit value will be properly aligned without the tag handler needing to know the contents of the tag.

NOTE 2 A data element is aligned with respect to a data type if the address of the data element is an integral multiple of the number of bytes in the data type.

7.3.5 Tag data element size (byte position 12 to 15 and repeating)

The tag data element size shall be the number of bytes in the tag data element encoded as a uInt32Number. The value of the tag data element size shall be the number of actual data bytes and shall not include any padding at the end of the tag data element.

7.4 Tag data

The first set of tag data elements shall immediately follow the tag table and all tag data elements, including the last tag data element, shall be padded by no more than three following pad bytes to reach a 4-byte boundary.

The size of individual tag data elements and the accumulated size of all tag data elements shall only be restricted by the limits imposed by the 32-bit tag data offset value and the 32-bit tag data element size value.

8 Required tags

8.1 General

8.2 to 8.9 identify the tags that are required, in addition to the header defined in 7.2, for each profile type. (These required tags are also given in tabular form in Annex G.)

NOTE Profiles can include additional tags beyond those listed as required. The explicitly listed tags are those which are required in order to comprise a legal profile of each type.

The intent of requiring certain tags with each type of profile is to provide a common base level of functionality. If a custom CMM is not present, then the required tags will have enough information to allow the default CMM to perform the requested colour transformations. The particular models implied by the required data are identified for each profile type and described in detail in Annex F. While the data provided by the required tags might not provide the level of quality obtainable with optional tags and private data, the data provided is adequate for sophisticated device modelling.

8.2 Common requirements

With the exception of DeviceLink profiles, all profiles shall contain the following tags:

- profileDescriptionTag (see 9.2.41);
- copyrightTag (see 9.2.21);
- mediaWhitePointTag (see 9.2.34);
- chromaticAdaptationTag, when the measurement data used to calculate the profile was specified for an adopted white with a chromaticity different from that of the PCS adopted white (see 9.2.15).

NOTE A DeviceLink profile is not required to have either a mediaWhitePointTag or a chromaticAdaptationTag.

If the image state produced in the PCS using the ICC-absolute or media-relative colorimetric rendering intents is not picture-referred, the colorimetricIntentImageStateTag should be included. If the colorimetricIntentImageStateTag is not included, colorimetry represented in the PCS should be interpreted to be picture-referred, with the media class as indicated in the profile header attributes.

For all profiles it is permissible to reference the same tag data for all rendering intents, and to use the media-relative colorimetric intent tag when ICC-absolute colorimetry is specified.

8.3 Input profiles

8.3.1 General

Input profiles are generally used with devices such as scanners and digital cameras. The types of profiles available for use as Input profiles are N-component LUT-based, Three-component matrix-based, and monochrome.

8.3.2 N-component LUT-based Input profiles

In addition to the tags listed in 8.2 an N-component LUT-based Input profile shall contain the following tag:

— AToB0Tag (see 9.2.1).

The AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA0Tag (see 9.2.6), BToA1Tag (see 9.2.7), BToA2Tag (see 9.2.8) may also be included in an N-component LUT-based Input profile. If these are present, their usage shall be as defined in Table 25 (see 9.1). The DToB0Tag (see 9.2.24), DToB1Tag (see 9.2.25), DToB2Tag (see 9.2.26), DToB3Tag (see 9.2.27), BToD0Tag (see 9.2.9), BToD1Tag (see 9.2.10), BToD2Tag (see 9.2.11), and BToD3Tag (see 9.2.12) may also be included.

A gamutTag (see 9.2.28) may be included. The usage of this tag is identical as in Output profiles.

8.3.3 Three-component matrix-based Input profiles

In addition to the tags listed in 8.2, a three-component matrix-based Input profile shall contain the following tags:

— redMatrixColumnTag (see 9.2.44);

— greenMatrixColumnTag (see 9.2.30);

— blueMatrixColumnTag (see 9.2.4);

— redTRCTag (see 9.2.45);

— greenTRCTag (see 9.2.31);

— blueTRCTag (see 9.2.5).

The AToB0Tag (see 9.2.1), AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA0Tag (see 9.2.6), BToA1Tag (see 9.2.7), and BToA2Tag (see 9.2.8) may also be included in a three-component matrix-based Input profile. If these are present, their usage shall be as defined in Table 25 (see 9.1).

In addition a gamutTag (see 9.2.28) may be included. The usage of this tag is identical as in Output profiles.

Only the PCSXYZ encoding can be used with matrix/TRC models. This profile may be used for any device which has a three-component colour space suitably related to PCSXYZ by this model.

NOTE If the PCSLAB encoding is to be used, the profile is required to be an N-component LUT-based Input profile, which includes an AToB0Tag (see 8.3.2), instead of the matrix-based profile.

The computational model supported by three-component matrix-based Input profiles shall be that defined in F.3.

8.3.4 Monochrome Input profiles

In addition to the tags listed in 8.2, a monochrome Input profile shall contain the following tag:

— grayTRCTag (see 9.2.29).

The AToB0Tag (see 9.2.1), AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA0Tag (see 9.2.6), BToA1Tag (see 9.2.7), and BToA2Tag (see 9.2.8) may also be included in monochrome Input profiles. If these are present, their usage shall be as defined in Table 25 (see 9.1).

The computational model supported by the grayTRCTag shall be that defined in F.2.

8.4 Display profiles

8.4.1 General

This class of profiles represents display devices such as monitors. The types of profiles available for use as Display profiles are N-component LUT-based, Three-component matrix-based, and monochrome.

8.4.2 N-Component LUT-based Display profiles

In addition to the tags listed in 8.2 an N-component LUT-based Input profile shall contain the following tags:

— AToB0Tag (see 9.2.1);

— BToA0Tag (see 9.2.6).

The AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA1Tag (see 9.2.7), and BToA2Tag (see 9.2.8) may also be included in an N-component LUT-based Display profile. If these are present, their usage shall be as defined in Table 25 (see 9.1).

The DToB0Tag (see 9.2.24), DToB1Tag (see 9.2.25), DToB2Tag (see 9.2.26), DToB3Tag (see 9.2.27), BToD0Tag (see 9.2.9), BToD1Tag (see 9.2.10), BToD2Tag (see 9.2.11), BToD3Tag (see 9.2.12) may also be included.

A gamutTag (see 9.2.28) may be included. The usage of this tag is identical as in Output profiles.

8.4.3 Three-component matrix-based Display profiles

In addition to the tags listed in 8.2, a three-component matrix-based Display profile shall contain the following tags:

— redMatrixColumnTag (see 9.2.44);

— greenMatrixColumnTag (see 9.2.30);

— blueMatrixColumnTag (see 9.2.4);

— redTRCTag (see 9.2.45);

— greenTRCTag (see 9.2.31);

— blueTRCTag (see 9.2.5).

The AToB0Tag (see 9.2.1), AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA0Tag (see 9.2.6), BToA1Tag (see 9.2.7), and BToA2Tag (see 9.2.8) may also be included in three-component matrix-based Display profiles. If these are present, their usage shall be as defined in Table 25 (see 9.1).

In addition a gamutTag (see 9.2.28) may be included. The usage of this tag is identical as in Output profiles.

Only the PCSXYZ encoding can be used with matrix/TRC models. This profile may be used for any device which has a three-component colour space suitably related to PCSXYZ by this model.

NOTE If the PCSLAB encoding is to be used the profile is required to be an N-component LUT-based Display profile instead of the matrix-based profile.

The computational model supported by three-component matrix-based Display profiles shall be that defined in F.3.

8.4.4 Monochrome Display profiles

In addition to the tags listed in 8.2 a monochrome Display profile shall contain the following tag:

— grayTRCTag (see 9.2.29).

The AToB0Tag (see 9.2.1), AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA0Tag (see 9.2.6), BToA1Tag (see 9.2.7), and BToA2Tag (see 9.2.8) may also be included in monochrome Display profiles. If these are present, their usage shall be as defined in Table 25 (see 9.1).

The computational model supported by the grayTRCTag shall be that defined in F.2.

8.5 Output profiles

8.5.1 General

Output profiles are used to support devices such as printers and film recorders. The types of profiles available for use as Output profiles are N-component LUT-based and Monochrome.

8.5.2 N-component LUT-based Output profiles

In addition to the tags listed in 8.2 an N-component LUT-based Output profile shall contain the following tags:

— AToB0Tag (see 9.2.1);

— AToB1Tag (see 9.2.2);

— AToB2Tag (see 9.2.3);

— BToA0Tag (see 9.2.6);

— BToA1Tag (see 9.2.7);

— BToA2Tag (see 9.2.8);

— gamutTag (see 9.2.28);

— colorantTableTag (see 9.2.18), for the xCLR colour spaces (see 7.2.6)

The colorantTableTag (9.2.18) is a required tag only for xCLR colour spaces. It enables the names and PCSXYZ or PCSLAB values of the colorants to be specified for these colour spaces (Table 19), as these names are not otherwise implicit in the choice of the colour space.

DToB0Tag (see 9.2.24), DToB1Tag (see 9.2.25), DToB2Tag (see 9.2.26), DToB3Tag (see 9.2.27), BToD0Tag (see 9.2.9), BToD1Tag (see 9.2.10), BToD2Tag (see 9.2.11), and BToD3Tag (see 9.2.12) may also be included in an N-component LUT-based Output profile.

8.5.3 Monochrome Output profiles

In addition to the tags listed in 8.2 a monochrome Output profile shall contain the following tag:

- grayTRCTag (see 9.2.29).

The AToB0Tag (see 9.2.1), AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA0Tag (see 9.2.6), BToA1Tag (see 9.2.7), and BToA2Tag (see 9.2.8) may also be included in a monochrome Output profile. If these are present, their usage shall be as defined in Table 25 (see 9.1).

The computational model supported by the grayTRCTag shall be that defined in F.2.

8.6 DeviceLink profile

A DeviceLink profile shall contain the following tags:

- profileDescriptionTag (see 9.2.41);
- copyrightTag (see 9.2.21);
- profileSequenceDescTag (see 9.2.42);
- AToB0Tag (see 9.2.1);
- colorantTableTag (see 9.2.18) which is required only if the data colour space field is xCLR, where x is hexadecimal 2 to F (see 7.2.6);
- colorantTableOutTag (see 9.2.19), required only if the PCS field is xCLR, where x is hexadecimal 2 to F (see 7.2.6)

This profile contains a pre-evaluated transform that cannot be undone, which represents a one-way link or connection between devices. It does not represent any device model nor can it be embedded into images.

The single AToB0Tag may contain data for any one of the four possible rendering intents. The rendering intent used is indicated in the header of the profile.

The data colour space field (see 7.2.6) in the DeviceLink profile will be the same as the data colour space field of the first profile in the sequence used to construct the device link. The PCS field (see 7.2.7) will be the same as the data colour space field of the last profile in the sequence.

If the data colour space field is set to xCLR, where x is hexadecimal 2 to F, the colorantTableTag (9.2.18) is a required tag to specify the names and PCSLAB values of the input colorants (Table 19), as these names are not otherwise implicit in the choice of the colour space. These colorants represent the input values of the profile.

Correspondingly, if the PCS field is set to xCLR where x is hexadecimal 2 to F, the colorantTableOutTag (9.2.19) is required, and represents the output colorants (Table 19). Only PCSLAB values are permitted.

A DeviceLink profile may contain a DToB0Tag (see 9.2.24).

NOTE The colorantOrderType tag ‘clro’ specifies the laydown order of the output colorants.

8.7 ColorSpace profile

In addition to the tags listed in 8.2, a ColorSpace profile shall contain the following tags:

- BToA0Tag (see 9.2.6);
- AToB0Tag (see 9.2.1).

This profile provides the relevant information to perform a transformation between colour encodings and the PCS. This type of profile is based on modelling rather than device measurement or characterization data. ColorSpace profiles may be embedded in images.

For ColorSpace profiles, the device profile dependent fields are set to zero if irrelevant.

The AToB1Tag (see 9.2.2), AToB2Tag (see 9.2.3), BToA1Tag (see 9.2.7), BToA2Tag (see 9.2.8) DToB0Tag (see 9.2.24), DToB1Tag (see 9.2.25), DToB2Tag (see 9.2.26), DToB3Tag (see 9.2.27), BToD0Tag (see 9.2.9), BToD1Tag (see 9.2.10), BToD2Tag (see 9.2.11), and BToD3Tag (see 9.2.12) may also be included in a ColorSpace profile. If these are present, their usage shall be as defined in Table 25 (see 9.1).

A gamutTag (see 9.2.28) may be included. The usage of this tag is identical as in Output profiles.

8.8 Abstract profile

In addition to the tags listed in 8.2, an Abstract profile shall contain the following tag:

— AToB0Tag (see 9.2.1).

This profile represents abstract transforms and does not represent any device model. Colour transformations using Abstract profiles are performed from PCS to PCS. Abstract profiles cannot be embedded in images.

An Abstract profile may contain a DToB0Tag (see 9.2.24).

8.9 NamedColor profile

In addition to the tags listed in 8.2, a NamedColor profile shall contain the following tag:

— namedColor2Tag (see 9.2.35).

NamedColor profiles can be thought of as sibling profiles to device profiles. For a given device there would be one or more device profiles to handle process colour conversions and one or more named colour profiles to handle named colours.

The namedColor2Tag provides a PCS and optional device representation for each named colour in a list of named colours. NamedColor profiles are device-specific in that their data is shaped for a particular device. There might be multiple NamedColor profiles to account for different consumables or multiple named colour vendors. The PCS representation is provided to support general colour management functionality. It is very useful for display and emulation of the named colours.

When using a NamedColor profile with the device for which it is intended, the device representation of the colour specifies the exact device coordinates for each named colour, if available. The PCS representation in conjunction with the device's Output profile can provide an approximation of these exact coordinates. The exactness of this approximation is a function of the accuracy of the Output profile and the colour management system performing the transformations.

The combination of the PCS and device representations provides for flexibility with respect to accuracy and portability.

8.10 Precedence order of tag usage

8.10.1 General

There are several methods of colour transformation that can function within a single CMM. If data for more than one method are included in the same profile, the following selection algorithm shall be used by the software implementation.

8.10.2 Input, display, output, or colour space profile types

For input, display, output, or colour space profile types, the precedence order of the tag usage for a designated rendering intent shall be the following.

a) Use the BToD0Tag, BToD1Tag, BToD2Tag, BToD3Tag, DToB0Tag, DToB1Tag, DToB2Tag, or DToB3Tag designated for the rendering intent if the tag is present, except where this tag is not needed or supported by the CMM (if a particular processing element within the tag is not supported the tag is not supported).

b) Use the BToA0Tag, BToA1Tag, BToA2Tag, AToB0Tag, AToB1Tag, or AToB2Tag designated for the rendering intent if present, when the tag in a) is not used.

c) Use the BToA0Tag or AToB0Tag if present, when the tags in a) and b) are not used.

d) Use TRCs (redTRCTag, greenTRCTag, blueTRCTag, or grayTRCTag) and colorants (redMatrixColumnTag, greenMatrixColumnTag, blueMatrixColumnTag) when tags in a), b), and c) are not used.

See Table 25.

8.10.3 DeviceLink or Abstract profile types

For DeviceLink or Abstract profile types, the precedence order of the tag usage shall be:

a) Use the DToB0Tag if present, except where this tag is not needed or supported by the CMM (if a particular processing element within the tag is not supported the tag is not supported).

b) Use the AToB0Tag if the DToB0Tag is not used.

See Table 25.

9 Tag definitions

9.1 General

The public tags currently defined by the ICC are listed in 9.2 in alphabetical order. All tags, including private tags, have as their first four bytes a tag signature to identify to profile readers what kind of data is contained within a tag. Each entry in 9.2 contains the tag signatures that shall be used for that tag, the permitted tag types for each tag (see Clause 10), and a brief description of the purpose of each tag. A short form tabular listing of all publicly available tags is given in Table G.13.

These individual tags are used to create all possible profiles. The tag signature indicates only the type of data and does not imply anything about the use or purpose for which the data is intended. Clause 8 specifies the tags that shall be included for each type of profile. Any other tag in 9.2 may be used as an optional tag as long as they are not specifically excluded in the definition of a profile class.

The interpretation of some tags is context dependent. This dependency is described in Table 25 which provides a useful summary of the rendering intent associated with each of the main profile classes and models. The term “undefined” means that the use of the tag in that situation is not specified by the ICC. The ICC recommends that such tags not be included in profiles. If the tag is present, its use is implementation dependent. In general, the BToAxTags represent the inverse operation of the AToBxTags. Note that the AToB1Tag and BToA1Tag are used to provide both of the colorimetric intents.

The AToBxTags and BToAxTags represent a model that can include a multi-dimensional lookup table. These models are described in detail in 10.8, 10.9, 10.10 and 10.11

The DToBxTags and BToDxTags represent colour transforms that operate on a range of values encoded by 32-bit IEEE 754 floating-point numbers. The processing model is described in detail in 10.14. These tags are optional for all profile classes.

The “D” space represents a 32-bit IEEE 754 floating-point-encoded Device Space. In this space, negative as well as positive values (including values greater than 1,0) are allowed when such values are supported by the device.

The “B” space represents the PCS, identical to the “B” space in the AToBxTags and BToAxTags. The encoding range of the “B” space in the DToBxTags and BToDxTags is defined in 6.3.4, as is the case with the “B” space in AToBxTags and BToAxTags. (See 6.3.4.)

The DToB3 and BToD3 tags allow the ability to directly encode the absolute rendering intent in a profile. The PCS for DToB3 and BToD3 represents ICC-absolute colorimetry with the values encoded as float32Number. The mediaWhitePoint tag is NOT used in the calculation of ICC-absolute colorimetry from the data in the DToB3 and BToD3 tags.

9.2 Tag listing

9.2.1 AToB0Tag

Tag signature: ‘A2B0’ (41324230h)

Permitted tag types: lut8Type or lut16Type or lutAToBType

This tag defines a colour transform from Device, Colour Encoding or PCS, to PCS, or a colour transform from Device 1 to Device 2, using lookup table tag element structures. For most profile classes it defines the transform to achieve perceptual rendering (see Table 25). The processing mechanisms are described in lut8Type or lut16Type or lutAToBType (see 10.8, 10.9 and 10.10).

Table 25 — Profile type/profile tag and defined rendering intents

Profile class	AToB0Tag	AToB1Tag	AToB2Tag	TRC/matrix & GrayTRC	BToA0Tag	BToA1Tag	BToA2Tag
Input	Device to PCS: perceptual	Device to PCS: colorimetric	Device to PCS: saturation	Colorimetric	PCS to device: perceptual	PCS to device: colorimetric	PCS to device: saturation
Display	Device to PCS: perceptual	Device to PCS: colorimetric	Device to PCS: saturation	Colorimetric	PCS to device: perceptual	PCS to device: colorimetric	PCS to device: saturation
Output	Device to PCS: perceptual	Device to PCS: colorimetric	Device to PCS: saturation	Undefined	PCS to device: perceptual	PCS to device: colorimetric	PCS to device: saturation
ColorSpace	Colour encoding to PCS: perceptual	Colour encoding to PCS: colorimetric	Colour encoding to PCS: saturation	Undefined	PCS to colour encoding: perceptual	PCS to colour encoding: colorimetric	PCS to colour encoding: saturation
Abstract	PCS to PCS	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
DeviceLink	Device1 to device2 rendering intent defined according to Table 22	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
NamedColor	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

9.2.2 AToB1Tag

Tag signature: ‘A2B1’ (41324231h)

Permitted tag types: lut8Type or lut16Type or lutAToBType

This tag describes the colour transform from Device or Colour Encoding to PCS using lookup table tag element structures. For most profile classes, it defines the transform to achieve colorimetric rendering (see Table 25). The processing mechanisms are described in lut8Type or lut16Type or lutAToBType (see 10.8, 10.9 and 10.10).

9.2.3 AToB2Tag

Tag signature: ‘A2B2’ (41324232h)

Permitted tag types: lut8Type or lut16Type or lutAToBType

This tag describes the colour transform from Device or Colour Encoding to PCS using lookup table tag element structures. For most profile classes it defines the transform to achieve saturation rendering (see Table 25). The processing mechanisms are described in lut8Type or lut16Type or lutAToBType (see 10.8, 10.9 and 10.10).

9.2.4 blueMatrixColumnTag

Tag signature: ‘bXYZ’ (6258595Ah)

Permitted tag type: XYZType

This tag contains the third column in the matrix used in matrix/TRC transforms.

9.2.5 blueTRCTag

Tag signature: ‘bTRC’ (62545243h)

Permitted tag types: curveType or parametricCurveType

This tag contains the blue channel tone reproduction curve. The first element represents no colorant (white) or phosphor (black) and the last element represents 100 % colorant (blue) or 100 % phosphor (blue).

9.2.6 BToA0Tag

Tag signature: ‘B2A0’ (42324130h)

Permitted tag types: lut8Type or lut16Type or lutBToAType

This tag defines a colour transform from PCS to Device or Colour Encoding using the lookup table tag element structures. For most profile classes, it defines the transform to achieve perceptual rendering (see Table 25). The processing mechanisms are described in lut8Type or lut16Type or lutBToAType (see 10.8, 10.9 and 10.11).

9.2.7 BToA1Tag

Tag signature: ‘B2A1’ (42324131h)

Permitted tag types: lut8Type or lut16Type or lutBToAType

This tag defines a colour transform from PCS to Device or Colour Encoding using the lookup table tag element structures. For most profile classes it defines the transform to achieve colorimetric rendering (see Table 25). The processing mechanisms are described in lut8Type or lut16Type or lutBToAType (see 10.8, 10.9 and 10.11).

9.2.8 BToA2Tag

Tag signature: ‘B2A2’ (42324132h)

Permitted tag types: lut8Type or lut16Type or lutBToAType

This tag defines a colour transform from PCS to Device or Colour Encoding using the lookup table tag element structures. For most profile classes it defines the transform to achieve saturation rendering (see Table 25). The processing mechanisms are described in lut8Type or lut16Type or lutBToAType (see 10.8, 10.9 and 10.11).

9.2.9 BToD0Tag

Tag signature ‘B2D0’ (42324430h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from PCS to Device. It supports float32Number-encoded input range, output range and transform, and provides a means to override the BToA0Tag. As with the BToA0Tag, it defines a transform to achieve a perceptual rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.10 BToD1Tag

Tag signature ‘B2D1’ (42324431h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from PCS to Device. It supports float32Number-encoded input range, output range and transform, and provides a means to override the BToA1Tag. As with the BToA1Tag, it defines a transform to achieve a media-relative colorimetric rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.11 BToD2Tag

Tag signature ‘B2D2’ (42324432h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from PCS to Device. It supports float32Number-encoded input range, output range and transform, and provides a means to override the BToA2Tag. As with the BToA2Tag, it defines a transform to achieve a saturation rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.12 BToD3Tag

Tag signature ‘B2D3’ (42324433h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from PCS to Device. It supports float32Number-encoded input range, output range and transform, and provides a means to override the BToA1Tag and associated ICC-absolute colorimetric rendering intent processing. As with the BToA1Tag and associated ICC-absolute colorimetric rendering intent processing, it defines a transform to achieve an ICC-absolute colorimetric rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.13 calibrationDateTimeTag

Tag signature: ‘calt’ (63616C74h)

Permitted tag type: dateTimeType

This tag contains the profile calibration date and time. This allows applications and utilities to verify if this profile matches a vendor’s profile and how recently calibration has been performed.

9.2.14 charTargetTag

Tag signature: ‘targ’ (74617267h)

Permitted tag type: textType

This tag contains the name of the registered characterization data set, or it contains the measurement data for a characterization target. This tag is provided so that distributed utilities can identify the underlying characterization data, create transforms “on the fly” or check the current performance against the original device performance.

The first seven characters of the text shall identify the nature of the characterization data.

If the first seven characters are “ICCHDAT”, then the remainder of the text shall be a single space followed by the Reference Name of a characterization data set in the Characterization Data Registry maintained by ICC, and terminated with a NULL byte (00h). The Reference Name in the text shall match exactly (including case) the Reference Name in the registry which may be found on the ICC web site (www.color.org).

If the first seven characters match one of the identifiers defined in an ANSI or ISO standard, then the tag embeds the exact data file format defined in that standard. Each of these file formats contains an identifying character string as the first seven characters of the format, allowing an external parser to determine which data file format is being used. This provides the facilities to include a wide range of targets using a variety of measurement specifications in a standard manner.

9.2.15 chromaticAdaptationTag

Tag signature: 'chad' (63686164h)

Permitted tag type: s15Fixed16ArrayType

This tag contains a matrix, which shall be invertible, and which converts an nCIEXYZ colour, measured using the actual illumination conditions and relative to the actual adopted white, to an nCIEXYZ colour relative to the PCS adopted white, with complete adaptation from the actual adopted white chromaticity to the PCS adopted white chromaticity.

The tag reflects a survey of the currently used methods of conversion, all of which can be formulated as a matrix transformation. The Bradford transform (see Annex E) is recommended for ICC profiles.

Such a 3 × 3 chromatic adaptation matrix is organized as a 9-element array of s15Fixed16Number numbers (s15Fixed16ArrayType tag). Similarly as in the other occurrences of a 3 × 3 matrix in the ICC tags, the dimension corresponding to the matrix rows varies least rapidly while the one corresponding to the matrix columns varies most rapidly.

$$\text{array} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} X_{\text{PCS}} \\ Y_{\text{PCS}} \\ Z_{\text{PCS}} \end{bmatrix} = \begin{bmatrix} a_o & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 \end{bmatrix} \square \begin{bmatrix} X_{\text{SRC}} \\ Y_{\text{SRC}} \\ Z_{\text{SRC}} \end{bmatrix} \tag{11}$$

where $(XYZ)_{SRC}$ represents the measured nCIEXYZ value in the actual device viewing condition and $(XYZ)_{PCS}$ represents the chromatically adapted nCIEXYZ value.

The chromatic adaptation matrix is a combination of three separate conversions as defined in Annex E.

9.2.16 chromaticityTag

Tag signature: ‘chrm’ (6368726Dh)

Permitted tag type: chromaticityType

This tag contains the type and the data of the phosphor/colorant chromaticity set used.

9.2.17 colorantOrderTag

Tag signature: ’clro’ (636C726Fh)

Permitted tag type: colorantOrderType

This tag specifies the laydown order of colorants.

9.2.18 colorantTableTag

Tag signature: ’clrt’ (636C7274h)

Permitted tag type: colorantTableType

This tag identifies the colorants used in the profile by a unique name and set of PCSXYZ or PCSLAB values. When used in DeviceLink profiles only the PCSLAB values shall be permitted.

9.2.19 colorantTableOutTag

Tag signature: ‘clot’ (636C6F74h)

Permitted tag type: colorantTableType

This tag identifies the colorants used in the profile by a unique name and set of PCSLAB values. This tag is used for DeviceLink profiles only.

9.2.20 colorimetricIntentImageStateTag

Tag signature: ‘ciis’ (63696973h)

Permitted tag type: signatureType

This tag indicates the image state of PCS colorimetry produced using the colorimetric intent transforms. If present, the colorimetricIntentImageStateTag shall specify one of the ICC-defined image states shown in Table 26 and described herein. Other image state specifications are reserved for future ICC use.

NOTE 1 When the state of the image colorimetry represented in the PCS is different from that of the image data in the file, the colorimetric intent image state includes the word “estimates”. This will be the case when transformation of the image file data to colorimetry is not fully deterministic.

EXAMPLE If the spectral sensitivities of a digital camera sensor (or photographic film) are not a linear transform of the CIE XYZ colour matching functions, there will not be a single “correct” transform to focal plane colorimetry.

Table 26 — colorimetricIntentImageStateTag signatures

Colorimetric intent image state	Signature	Hex encoding
Scene colorimetry estimates	'scoe'	73636F65h
Scene appearance estimates	'sape'	73617065h
Focal plane colorimetry estimates	'fpce'	66706365h
Reflection hardcopy original colorimetry	‘rhoc’	72686F63h
Reflection print output colorimetry	‘rpoc’	72706F63h

The tag value 'scoe' (scene colorimetry estimates) shall indicate that colorimetry in the PCS represents estimates of the colorimetry of the scene, as viewed from the capture point, chromatically adapted from the scene adopted white chromaticity to the PCS D50 adopted white chromaticity. With the media-relative colorimetric intent, the colorimetry is relative to the scene encoding maximum white (after chromatic adaptation). With the ICC-absolute colorimetric intent, the colorimetry is relative to the scene adopted white (after chromatic adaptation). The scene colorimetry can result from a real scene, a synthetically generated scene, an edited scene, or some combination of these, but shall be interpreted as actual scene colorimetry for subsequent processing. When this image state is specified, the actual scene viewing conditions, including the adopted white, shall be specified in the viewing conditions tag.

For scene colorimetry estimates, the mediaWhitePointTag is populated with the XYZ tristimulus values of the scene encoding maximum white, normalized to be relative to the scene adopted white (perfect diffuser), and then converted to the corresponding tristimulus values for the D50 PCS white point using the chromaticAdaptationTag matrix (if required). The scene adopted white Y value is normalized to 1,0; the mediaWhitePointTag Y value is relative to the scene adopted white Y value and can be larger than 1,0.

NOTE 2 The un-normalized adopted white values are stored in the illuminant field in the viewing conditions tag.

NOTE 3 Since the media white point Y value can be larger than 1,0, brighter than diffuse white colours (e.g. specular highlights) can be represented.

The tag value 'sape' (scene appearance estimates**)** shall indicate that colorimetry in the PCS represents estimates of the appearance of the scene, as viewed from the capture point, fully adapted to the ISO 3664 P2 viewing conditions. With the media relative colorimetric intent, the corresponding colorimetry is relative to the scene encoding maximum white (after adaptation). With the ICC-absolute colorimetric intent, the corresponding colorimetry is relative to the scene adopted white (after adaptation). The scene appearance estimates may result from a real scene, a synthetically generated scene, an edited scene, or some combination of these, but shall be interpreted as scene appearance estimates for an actual scene in subsequent processing. When this image state is specified, the ISO 3664 P2 viewing conditions shall be specified in the viewing conditions tag.

For scene appearance estimates, the mediaWhitePointTag is populated with the XYZ tristimulus values of the scene encoding maximum white, normalized to be relative to the scene adopted white (perfect diffuser), and then converted to the corresponding tristimulus values for the D50 PCS white point using the chromaticAdaptationTag matrix (if required). The scene adopted white Y value is normalized to 1,0; the mediaWhitePointTag Y value is relative to the scene adopted white Y value and can be larger than 1,0.

NOTE 4 Since the media white point Y value can be larger than 1,0, brighter than diffuse white colours can be represented (e.g. specular highlights).

The tag value 'fpce' (focal plane colorimetry estimates**)** shall indicate that colorimetry in the PCS represents estimates of the colorimetry of the light present at the focal plane of a camera (digital or film), chromatically adapted from the focal plane adopted white chromaticity to the PCS D50 adopted white chromaticity. With the media relative colorimetric intent, the colorimetry is relative to the focal-plane encoding maximum white (after chromatic adaptation). With the ICC-absolute colorimetric intent, the colorimetry is relative to the focal plane adopted white (after chromatic adaptation). The focal plane colorimetry may result from a real scene, a synthetically generated scene, an edited scene, or some combination of these, but shall be interpreted as focal plane colorimetry for subsequent processing. When this colorimetric intent image state is specified, the actual focal plane viewing conditions, including the adopted white, shall be specified in the viewing conditions tag.

For focal plane colorimetry estimates, the mediaWhitePointTag is populated with the XYZ tristimulus values of the focal plane encoding maximum white, normalized to be relative to the focal plane adopted white (perfect diffuser), and then converted to the corresponding tristimulus values for the D50 PCS white point using the chromaticAdaptationTag matrix (if required). The focal plane adopted white Y value is normalized to 1,0; the mediaWhitePointTag Y value is relative to the focal plane adopted white Y value and can be larger than 1,0.

NOTE 5 The effects of any optics in or attached to the camera are included in the focal plane colorimetry estimates; this includes lens flare, filters, etc.

NOTE 6 The un-normalized adopted white values are stored in the illuminant field in the viewing conditions tag.

NOTE 7 Since the media white point Y value can be larger than 1,0, brighter than diffuse white colours can be represented (e.g. specular highlights).

The tag value 'rhoc' (reflection hardcopy original colorimetry**)** shall indicate that colorimetry in the PCS represents the colorimetry of a reflection hardcopy original that has been digitally scanned, and chromatically adapted from the scanner adopted white chromaticity to the PCS D50 adopted white chromaticity. With the media relative colorimetric intent, the colorimetry is normalized relative to the scan condition encoding maximum white after chromatic adaptation to D50. With the ICC-absolute colorimetric intent, the colorimetry is relative to the perfect reflecting diffuser after chromatic adaptation to D50. When this colorimetric intent image state is specified, the scan illumination conditions, including the adopted white, shall be specified in the viewing conditions tag and the chromatic adaptation shall be specified in the chromaticAdaptationTag.

NOTE 8 The un-normalized adopted white values are stored in the illuminant field in the viewing conditions tag.

The tag value 'rpoc' (reflection print output colorimetry**)** shall indicate that colorimetry in the PCS represents the colorimetry of reflection print output, after chromatic adaptation from the print viewing condition adopted white chromaticity to the PCS D50 adopted white chromaticity. With the media relative colorimetric intent, the colorimetry is normalized relative to the print medium white point, measured under the actual print viewing conditions. With the ICC-absolute colorimetric intent, the colorimetry is relative to the perfect reflecting diffuser after chromatic adaptation to D50. When this colorimetric intent image state is specified, the print viewing conditions, including the adopted white, shall be specified in the viewing conditions tag, and the chromatic adaptation shall be specified in the chromaticAdaptationTag.

NOTE 9 The un-normalized adopted white values are stored in the illuminant field in the viewing conditions tag.

9.2.21 copyrightTag

Tag signature: ‘cprt’ (63707274h)

Permitted tag type: multiLocalizedUnicodeType

This tag contains the text copyright information for the profile.

9.2.22 deviceMfgDescTag

Tag signature: ‘dmnd’ (646D6E64h)

Permitted tag type: multiLocalizedUnicodeType

This tag describes the structure containing invariant and localizable versions of the device manufacturer for display. The content of this structure is described in 10.13.

9.2.23 deviceModelDescTag

Tag signature: ‘dmdd’ (646D6464h)

Permitted tag type: multiLocalizedUnicodeType

This tag describes the structure containing invariant and localizable versions of the device model for display. The content of this structure is described in 10.13.

9.2.24 DToB0Tag

Tag signature ‘D2B0’ (44324230h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from Device to PCS. It supports float32Number-encoded input range, output range and transform, and provides a means to override the AToB0Tag. As with the AToB0Tag, it defines a transform to achieve a perceptual rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.25 DToB1Tag

Tag signature ‘D2B1’ (44324231h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from Device to PCS. It supports float32Number-encoded input range, output range and transform, and provides a means to override the AToB1Tag. As with the AToB1Tag, it defines a transform to achieve a media-relative colorimetric rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.26 DToB2Tag

Tag signature ‘D2B2’ (44324232h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from Device to PCS. It supports float32Number-encoded input range, output range and transform, and provides a means to override the AToB2Tag. As with the AToB2Tag, it defines a transform to achieve a saturation rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.27 DToB3Tag

Tag signature ‘D2B3’ (44324233h)

Allowed tag types: multiProcessElementsType

This tag defines a colour transform from Device to PCS. It supports float32Number-encoded input range, output range and transform, and provides a means to override the AToB1Tag with associated ICC-absolute colorimetric rendering intent processing. As with the AToB1Tag and associated ICC-absolute colorimetric rendering intent processing, it defines a transform to achieve an ICC-absolute colorimetric rendering. The processing mechanism is described in multiProcessElementsType (see 10.14).

9.2.28 gamutTag

Tag signature: ‘gamt’ (67616D74h)

Permitted tag types: lut8Type or lut16Type or lutBToAType

Out of gamut tag. The processing mechanisms are described in lut8Type or lut16Type or lutBToAType.

This tag provides a table in which PCS values are the input and a single output value for each input value is the output. If the output value is 0, the PCS colour is in-gamut. If the output is non-zero, the PCS colour is out-of-gamut.

9.2.29 grayTRCTag

Tag signature: 'kTRC' (6B545243h)

Permitted tag types: curveType or parametricCurveType

This tag contains the grey tone reproduction curve. The tone reproduction curve provides the necessary information to convert between a single device channel and the PCSXYZ or PCSLAB encoding. The first element represents black and the last element represents white. The computational model supported by the grayTRC tag is defined in F.2.

9.2.30 greenMatrixColumnTag

Tag signature: 'gXYZ' (6758595Ah)

Permitted tag type: XYZType

This tag contains the second column in the matrix, which is used in matrix/TRC transforms.

9.2.31 greenTRCTag

Tag signature: 'gTRC' (67545243h)

Permitted tag types: curveType or parametricCurveType

This tag contains the green channel tone reproduction curve. The first element represents no colorant (white) or phosphor (black) and the last element represents 100 % colorant (green) or 100 % phosphor (green).

9.2.32 luminanceTag

Tag signature: 'lumi' (6C756D69h)

Permitted tag type: XYZType

This tag contains the absolute luminance of emissive devices in candelas per square metre as described by the Y channel.

NOTE The X and Z values are set to zero.

9.2.33 measurementTag

Tag signature: 'meas' (6D656173h)

Permitted tag type: measurementType

This tag describes the alternative measurement specification, such as a D65 illuminant instead of the default D50.

9.2.34 mediaWhitePointTag

Tag signature: 'wtpt' (77747074h)

Permitted tag type: XYZType

This tag, which is used for generating the ICC-absolute colorimetric intent, specifies the chromatically adapted nCIEXYZ tristimulus values of the media white point. When the measurement data used to create the profile were specified relative to an adopted white with a chromaticity different from that of the PCS adopted white, the media white point nCIEXYZ values shall be adapted to be relative to the PCS adopted white chromaticity using the chromaticAdaptationTag matrix, before recording in the tag. For capture devices, the media white point is the encoding maximum white for the capture encoding. For displays, the values specified shall be those of the PCS illuminant as defined in 7.2.16.

See Clause 6 and Annex A for a more complete description of the use of the media white point.

9.2.35 namedColor2Tag

Tag signature: ‘ncl2’ (6E636C32h)

Permitted tag type: namedColor2Type

This tag contains the named colour information providing a PCS and optional device representation for a list of named colours.

9.2.36 outputResponseTag

Tag signature: ‘resp’ (72657370h)

Permitted tag type: responseCurveSet16Type

This tag describes the structure containing a description of the device response for which the profile is intended. The content of this structure is described in 10.19.

NOTE The user’s attention is called to the possibility that the use of this tag for device calibration can require use of an invention covered by patent rights. By publication of this part of ISO 15076, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and non-discriminatory terms and conditions to applicants desiring to obtain such a license. Details can be obtained from the International Color Consortium (1899 Preston White Drive, Reston, Virginia 20191-4367, USA). See Introduction.

9.2.37 perceptualRenderingIntentGamutTag

Tag signature: ‘rig0’ (72696730h)

Permitted tag type: signatureType

There is only one standard reference medium gamut, as defined in ISO 12640-3. When the signature is present, the specified gamut is defined to be the reference medium gamut for the PCS side of both the A2B0 and B2A0 tags, if they are present. If this tag is not present, the perceptual rendering intent reference gamut is unspecified.

The standard PCS reference medium gamut signatures that shall be used are listed in Table 27:

Table 27 — Perceptual rendering intent gamut

PCS reference medium gamut	Signature	Hex encoding
Perceptual reference medium gamut	‘prmg’	70726D67h

NOTE 1 Because the perceptual intent is the typical default rendering intent, it is most important to use the PRMG for this rendering intent.

NOTE 2 It is possible that the ICC will define other signature values in the future.

9.2.38 preview0Tag

Tag signature: ‘pre0’ (70726530h)

Permitted tag types: lut8Type or lut16Type or lutAToBType or lutBToAType

This tag contains the preview transformation from PCS to device space and back to the PCS. This tag contains the combination of tag B2A0 and tag A2B1, or equivalent transforms. The processing mechanisms are described in lut8Type or lut16Type or lutAToBType or lutBToAType (see 10.8, 10.9, 10.10 and 10.11).

9.2.39 preview1Tag

Tag signature: ‘pre1’ (70726531h)

Permitted tag types: lut8Type or lut16Type or lutBToAType

This tag defines the preview transformation from PCS to device space and back to the PCS. This tag contains the combination of tag B2A1 and tag A2B1, or equivalent transforms. The processing mechanisms are described in lut8Type or lut16Type or lutAToBType or lutBToAType (see 10.8, 10.9, 10.10 and 10.11).

9.2.40 preview2Tag

Tag signature: ‘pre2’ (70726532h)

Permitted tag types: lut8Type or lut16Type or lutBToAType

This tag contains the preview transformation from PCS to device space and back to the PCS. This tag contains the combination of tag B2A2 and tag A2B1, or equivalent transforms. The processing mechanisms are described in lut8Type or lut16Type or lutAToBType or lutBToAType (see 10.8, 10.9, 10.10 and 10.11).

9.2.41 profileDescriptionTag

Tag signature: ‘desc’ (64657363h)

Permitted tag type: multiLocalizedUnicodeType

This tag describes the structure containing invariant and localizable versions of the profile description for display. The content of this structure is described in 10.13. This invariant description has no fixed relationship to the actual profile disk file name.

NOTE It is helpful if an identification of the characterization data that was used in the creation of the profile is included in the profileDescriptionTag (e.g. “based on CGATS TR 001”)[10]. See also 8.2.11.

9.2.42 profileSequenceDescTag

Tag signature: ‘pseq’ (70736571h)

Permitted tag type: profileSequenceDescType

This tag describes the structure containing a description of the profile sequence from source to destination, typically used with the DeviceLink profile. The content of this structure is described in 10.17.

9.2.43 profileSequenceIdentifierTag

Tag signature: ‘psid’ (70736964h)

Permitted tag type: profileSequenceIdentifierType

This tag describes the structure containing information for identification of the profiles used in a sequence. This tag is typically used in DeviceLink profiles to identify the original profiles that were combined to create the final profile.

9.2.44 redMatrixColumnTag

Tag signature: ‘rXYZ’ (7258595Ah)

Permitted tag type: XYZType

This tag contains the first column in the matrix, which is used in matrix/TRC transforms.

9.2.45 redTRCTag

Tag signature: ‘rTRC’ (72545243h)

Permitted tag types: curveType or parametricCurveType

This tag contains the red channel tone reproduction curve. The first element represents no colorant (white) or phosphor (black) and the last element represents 100 % colorant (red) or 100 % phosphor (red).

9.2.46 saturationRenderingIntentGamutTag

Tag signature: ‘rig2’ (72696732h)

Permitted tag type: signatureType

There is only one standard reference medium gamut, as defined in ISO 12640-3. When the signature is present, the specified gamut is defined to be the reference medium gamut for the PCS side of both the A2B2 and B2A2 tags, if they are present. If this tag is not present, the saturation rendering intent reference gamut is unspecified. The standard PCS reference medium gamut signatures that shall be used are listed in Table 28.

Table 28 — Saturation rendering intent gamut

PCS reference medium gamut	Signature	Hex encoding
Perceptual reference medium gamut	‘prmg’	70726D67h

NOTE It is possible that the ICC will define other signature values in the future.

9.2.47 technologyTag

Tag signature: ‘tech’ (74656368h)

Permitted tag type: signatureType

The device technology signatures that shall be used are listed in Table 29.

Table 29 — Technology signatures

Technology	Signature	Hex encoding
Film scanner	‘fscn’	6673636Eh
Digital camera	‘dcam’	6463616Dh
Reflective scanner	‘rscn’	7273636Eh
Ink jet printer	‘ijet’	696A6574h
Thermal wax printer	‘twax’	74776178h
Electrophotographic printer	‘epho’	6570686Fh
Electrostatic printer	‘esta’	65737461h
Dye sublimation printer	‘dsub’	64737562h
Photographic paper printer	‘rpho’	7270686Fh
Film writer	‘fprn’	6670726Eh
Video monitor	‘vidm’	7669646Dh
Video camera	‘vidc’	76696463h
Projection television	‘pjtv’	706A7476h
Cathode ray tube display	‘CRT ’	43525420h
Passive matrix display	‘PMD ’	504D4420h
Active matrix display	‘AMD ’	414D4420h
Photo CD	‘KPCD’	4B504344h
Photographic image setter	‘imgs’	696D6773h
Gravure	‘grav’	67726176h
Offset lithography	‘offs’	6F666673h
Silkscreen	‘silk’	73696C6Bh
Flexography	‘flex’	666C6578h
Motion picture film scanner	‘mpfs’	6D706673h
Motion picture film recorder	‘mpfr’	6D706672h
Digital motion picture camera	‘dmpc’	646D7063h
Digital cinema projector	‘dcpj’	64636A70h

9.2.48 viewingCondDescTag

Tag signature: ‘vued’ (76756564h)

Permitted tag type: multiLocalizedUnicodeType

This tag describes the structure containing invariant and localizable versions of the viewing conditions. The content of this structure is described in 10.13.

9.2.49 viewingConditionsTag

Tag signature: ‘view’ (76696577h)

Permitted tag type: viewingConditionsType

This tag defines the viewing conditions parameters. The content of this structure is described in 10.28.

10 Tag type definitions

10.1 General

All tags, including private tags, have as their first four bytes a tag type signature to identify to profile readers what kind of data is contained within a tag. This encourages tag type reuse and allows profile parsers to reuse code when tags use common tag types. The second four bytes (4 to 7) are reserved for future expansion and shall be set to 0 in this version of the specification. The tag signature for all private tags and any tag type signature not defined in this clause shall be registered with the International Color Consortium (see Clause 5) in order to prevent signature collisions.

One or more tag types are associated with each tag defined in 9.2. The tag type definitions in 10.2 to 10.29 specify the data structure that shall be used in creating the contents of the tag data element for each tag.

All tag data elements, including those of private tags, shall have a tag type signature in bytes 0 to 3. Bytes 4 to 7 are reserved for future expansion and shall be set to 0.

Any private tag types used shall be registered with the International Color Consortium to prevent tag type signature collisions.

NOTE An effort was made to make sure one-byte, 2-byte and 4-byte data lies on 1-byte, 2-byte and 4-byte boundaries respectively. This required occasionally including extra spaces indicated with “reserved for padding” in some tag type definitions.

Where not otherwise specified, value 0 is defined to be of “unknown value” for all enumerated data structures.

Where not specified otherwise, the least-significant 16 bits of all 32-bit flags in the type descriptions below are reserved for use by the International Color Consortium.

When 7-bit ASCII text representation is specified in types below, each individual character is encoded in 8 bits with the most-significant bit set to zero. The details are presented in 10.22.

In many of the tables shown in this clause, the following syntax is used in the encoding column for the various numeric types listed in Clause 4, i.e. numerictype[X] where X represents the number of values in that position. Where [...] is used, the number of values depends on the number of channels in the tag type or number of entries in a table.

10.2 chromaticityType

The chromaticity tag type provides basic chromaticity data and type of phosphors or colorants of a monitor to applications and utilities. When used, the byte assignment shall be as given in Table 30. The CIE xy values shall be as measured, and shall not be chromatically adapted to the PCS adopted white.

Table 30 — chromaticityType encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	’chrm’ (6368726Dh) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of device channels (n)	uInt16Number
10 to 11	2	Encoded value of phosphor or colorant type	see Table 31
12 to 19	8	CIE xy coordinate values of channel 1	u16Fixed16Number[2]
20 to end	$8(n-1)$	CIE xy coordinate values of other channels (if needed)	u16Fixed16Number [...]

When using this type, it is necessary to assign each colour component to a device channel. Table 38 “lut16Type channel encodings” shows these assignments.

The encoding for byte positions 10 and 11 is shown in Table 31. If the value is 0001h to 0004h, the number of channels shall be three and the phosphor chromaticities in byte positions 12 to 35 shall match those listed in the appropriate row of Table 31.

Table 31 — Colorant and phosphor encoding

Phosphor or colorant type as defined in:	**Encoded value**	**Channel 1** (x, y)	**Channel 2** (x, y)	**Channel 3** (x, y)
Unknown	0000h	Any	Any	Any
ITU-R BT.709-2	0001h	(0,640, 0,330)	(0,300, 0,600)	(0,150, 0,060)
SMPTE RP145	0002h	(0,630, 0,340)	(0,310, 0,595)	(0,155, 0,070)
EBU Tech. 3213-E	0003h	(0,640 0,330)	(0,290, 0,600)	(0,150, 0,060)
P22	0004h	(0,625, 0,340)	(0,280, 0,605)	(0,155, 0,070)

When the encoded value in byte position 10 and 11 is 0000h, the actual set of chromaticity values shall be described.

10.3 colorantOrderType

This is an optional tag which specifies the laydown order in which colorants will be printed on an n-colorant device. The laydown order may be the same as the channel generation order listed in the colorantTableTag or the channel order of a colour encoding type such as CMYK, in which case this tag is not needed. When this is not the case (for example, ink-towers sometimes use the order KCMY), this tag may be used to specify the laydown order of the colorants. When used the byte assignments shall be as given in Table 32.

Table 32 — colorantOrderType encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘clro’ (636c726fh) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Count of colorants (n)	uInt32Number
12	1	Number of the colorant to be printed first.	uInt8Number
13 to $(11+n)$	n–1	The remaining n–1 colorants are described in a manner consistent with the first colorant	uInt8Number

The size of the array is the same as the number of colorants. The first position in the array contains the number of the first colorant to be laid down, the second position contains the number of the second colorant to be laid down, and so on, until all colorants are listed.

When this tag is used, the “count of colorants” shall be in agreement with the data colour space signature of 7.2.6.

10.4 colorantTableType

The purpose of this tag is to identify the colorants used in the profile by a unique name and set of PCSXYZ or PCSLAB values to give the colorant an unambiguous value. The first colorant listed is the colorant of the first device channel of a LUT tag. The second colorant listed is the colorant of the second device channel of a LUT tag, and so on. When used the byte assignment shall be as given in Table 33.

Table 33 — colorantTableType encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘clrt’ (636c7274h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Count of colorants (n)	uInt32Number
12 to 43	32	First colorant name (32-byte field, null terminated, unused bytes shall be set to zero)	7-bit ASCII
44 to 49	6	PCS values of the first colorant in the colour space of the profile as described in 7.2.7 (the PCS Signature in the header). PCS values shall be relative colorimetric	16-bit integer uInt16Number [3]
50 to 49+38(n–1)	38(n–1)	The remaining colorants, if $n > 1$, described using the format of bytes 12 to 49 of the first colorant	(7-bit ASCII followed by 16-bit integer uInt16Number [3]) [...]

The PCS values are provided only for convenience and, for many profile classes, should be populated by processing the individual colorants through the AToB1Tag of the profile if this tag exists. Otherwise the user shall supply the values, if this tag is to be used. An individual colorant has the maximum device value in the channel corresponding to that colorant and the minimum device value in all other channels.

EXAMPLE Using a 3CLR profile, the colorant values for the first channel would be (1, 0, 0) where 1 is the maximum device value and 0 is the minimum device value. This would be achieved by dividing all the encoded values by the maximum value for that channel (e.g. dividing the 8-bit values 255, 0, 0 by 255). Processing this colour through the AToB1Tag would produce the PCS values listed in bytes 44 to 49.

When this tag is used, the “count of colorants” shall be in agreement with the data colour space signature of 7.2.6.

NOTE The PCSXYZ or PCSLAB values can also be used to derive the visual density of the colorant, which trapping algorithms can then use to determine overlay values.

10.5 curveType

The curveType contains a 4-byte count value and a one-dimensional table of 2-byte values. When used the byte assignment shall be as given in Table 34.

Table 34 — curveType encoding

Byte position	Field length **bytes**	Content	Encoded as
0 to 3	4	‘curv’ (63757276h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Count value specifying the number of entries (n) that follow	uInt32Number
12 to end	$2n^*$	Actual curve values starting with the zeroth entry and ending with the entry $n - 1$	uInt16Number [...] *
* If $n = 1$, the field length is 1 and the value is encoded as a u8Fixed8Number			

The curveType embodies a one-dimensional function which maps an input value in the domain of the function to an output value in the range of the function. The domain and range values are in the range of 0,0 to 1,0.

- When n is equal to 0, an identity response is assumed.
- When n is equal to 1, then the curve value shall be interpreted as a gamma value, encoded as a u8Fixed8Number. Gamma shall be interpreted as the exponent in the equation $y = x^{\gamma}$ and not as an inverse.
- When n is greater than 1, the curve values (which embody a sampled one-dimensional function) shall be defined as follows:
- The first entry is located at 0,0, the last entry at 1,0, and intermediate entries are uniformly spaced using an increment of 1,0/(n–1). These entries are encoded as uInt16Numbers (i.e. the values represented by the entries, which are in the range 0,0 to 1,0 are encoded in the range 0 to 65 535). Function values between the entries shall be obtained through linear interpolation.

If the input is PCSXYZ, 1+(32 767/32 768) shall be mapped to the value 1,0. If the output is PCSXYZ, the value 1,0 shall be mapped to 1+(32 767/32 768).

10.6 dataType

The dataType is a simple data structure that contains either 7-bit ASCII or binary data, i.e. textType data or transparent 8-bit bytes. The length of the string is obtained by subtracting 12 from the tag data element size portion of the tag itself as defined in 7.3.5. If this type is used for ASCII data, it shall be terminated with a 00h byte. When used, the byte assignment shall be as given in Table 35.

Table 35 — dataType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	'data' (64617461h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to 11	4	Data flag, 00000000h represents ASCII data, 00000001h represents binary data, other values are reserved for future use
12 to end	(tag data element size) – 12	A string of [(tag data element size) – 12] ASCII characters or [(tag data element size) – 12] bytes

10.7 dateTimeType

This dateTimeType is a 12-byte value representation of the time and date. The actual values are encoded as a dateTimeNumber described in 4.2. When used the byte assignment shall be as given in Table 36.

Table 36 — dateTimeType encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	'dtim' (6474696Dh) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 19	12	Date and time	dateTimeNumber

10.8 lut16Type

This structure represents a colour transform using tables with 16-bit precision. This type contains four processing elements: a 3 × 3 matrix (which shall be the identity matrix unless the input colour space is PCSXYZ), a set of one-dimensional input tables, a multi-dimensional lookup table, and a set of one-dimensional output tables. Data is processed using these elements via the following sequence:

(matrix) ⇒ (1D input tables) ⇒ (multi-dimensional lookup table, CLUT) ⇒ (1D output tables).

When used the byte assignment shall be as given in Table 37.

Table 37 — lut16Type encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	'mft2' (6D667432h) [multi-function table with 2-byte precision] type signature	
4 to 7	4	reserved, shall be set to 0	
8	1	Number of Input Channels (i)	uInt8Number
9	1	Number of Output Channels (o)	uInt8Number
10	1	Number of CLUT grid points (identical for each side) (g)	uInt8Number
11	1	Reserved for padding (required to be 00h)	
12 to 15	4	Encoded e1 parameter	s15Fixed16Number
16 to 19	4	Encoded e2 parameter	s15Fixed16Number
20 to 23	4	Encoded e3 parameter	s15Fixed16Number
24 to 27	4	Encoded e4 parameter	s15Fixed16Number
28 to 31	4	Encoded e5 parameter	s15Fixed16Number
32 to 35	4	Encoded e6 parameter	s15Fixed16Number
36 to 39	4	Encoded e7 parameter	s15Fixed16Number
40 to 43	4	Encoded e8 parameter	s15Fixed16Number
44 to 47	4	Encoded e9 parameter	s15Fixed16Number
48 to 49	4	Number of input table entries (n)	uInt16Number
50 to 51	4	Number of output table entries (m)	uInt16Number
52 to 51+($2ni$)	$2ni$	Input tables	uInt16Number [...]
52+($2ni$) to 51+($2ni$) + ($2g^io$)	$2g^io$	CLUT values	uInt16Number [...]
52+($2ni$) + ($2g^io$) to end	$2mo$	Output tables	uInt16Number [...]

Explanation of symbols:

i is the number of input channels

o is the number of output channels

g is the number of grid points

n is the number of input table entries

m is the number of output table entries

The input and output tables, and CLUT, contained in a lut16Type each embodies a one- or multi-dimensional function which maps an input value in the “domain” of the function to an output value in the “range” of the function.

The domain of each of these tables is defined to consist of all real numbers between 0,0 and 1,0, inclusive. The first entry is located at 0,0, the last entry at 1,0, and intermediate entries are uniformly spaced using an increment of 1,0/(K–1). For the input and output tables, K is the number of entries in the table. For the CLUT, K is the number of grid points along each dimension. The range of a function used to generate the contents of a table is likewise defined to be all real numbers between 0,0 and 1,0, inclusive. Since the domain and range of the tables are 0,0 to 1,0 it is necessary to convert all device values and PCSLAB values to this numeric range. It shall be assumed that the maximum value in each case is set to 1,0 and the minimum value to 0,0, and all intermediate values are linearly scaled accordingly.

Because the entries of a table are encoded as uInt16Numbers, it is necessary to round each real value to the nearest 16-bit integer.

Because the entries of lut16Type LUTs represent values in the range 0,0 to 1,0, encoded as uInt16Numbers, these entries should be divided by 65 535,0 for the calculation of the actual output values.

See Annex A for additional guidance on this topic.

The matrix is organized as a 3 × 3 array. The dimension corresponding to the matrix rows varies least rapidly and the dimension corresponding to the matrix columns varies most rapidly and is shown in matrix form below.

$$\begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \qquad (12)$$

When using the matrix of an Output profile, and the input data is PCSXYZ, it gives:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \square \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (13)$$

Each matrix entry is encoded as an s15Fixed16Number. The domain and range of the matrix is 0,0 to 1,0.

The matrix shall be an identity matrix unless the input is in the PCSXYZ colour space.

The input tables are arrays of 16-bit unsigned values. Each input table consists of a minimum of two and a maximum of 4 096 uInt16Number integers. Each input table entry is appropriately normalized to the range 0 to 65 535. The inputTable is of size (InputChannels × inputTableEntries × 2) bytes. When stored in this tag, the one-dimensional lookup tables are packed one after another in the order described in Table 38.

The CLUT is organized as an i-dimensional array with a given number of grid points in each dimension, where i is the number of input channels (input tables) in the transform. The dimension corresponding to the first input channel varies least rapidly and the dimension corresponding to the last input channel varies most rapidly. Each grid point value contains o uInt16Number integers, where o is the number of output channels. The first sequential uInt16Number integer of the entry contains the function value for the first output function, the second sequential uInt16Number integer of the entry contains the function value for the second output function, and so on until all the output functions have been supplied. The CLUT size, in bytes, is equal to 2 times number of OutputChannels times the number of GridPoints raised to the power of the number of InputChannels.

The output tables are arrays of 16-bit unsigned values. Each output table consists of a minimum of two and a maximum of 4 096 uInt16Number integers. Each output table entry is appropriately normalized to the range 0 to 65 535. The outputTable is of size (OutputChannels × outputTableEntries × 2) bytes. When stored in this tag, the one-dimensional lookup tables are packed one after another in the order described in Table 38.

If the number of data points in a one-dimensional table, or in a particular dimension of the CLUT, is two, the data for those points shall be set so that the correct results are obtained when linear interpolation is used to generate intermediate values.

When using this type, it is necessary to assign each data colour space component to an input and output channel. These assignments shall be as shown in Table 38. The channels are numbered according to the order in which their table occurs.

Table 38 — lut16Type channel encodings

Data colour space	Channel 1	Channel 2	Channel 3	Channel 4
'XYZ '	X	Y	Z	—
'Lab '	L	a	b	—
'Luv '	L	u	v	—
'YCbr'	Y	Cb	Cr	—
'Yxy '	Y	x	y	—
'RGB '	R	G	B	—
'GRAY'	K	—	—	—
'HSV '	H	S	V	—
'HLS '	H	L	S	—
'CMYK'	C	M	Y	K
'CMY '	C	M	Y	—
'2CLR'	Channel 1	Channel 2	—	—
'3CLR'	Channel 1	Channel 2	Channel 3	—
'4CLR'	Channel 1	Channel 2	Channel 3	Channel 4

NOTE Additional xCLR colour spaces (up to 15 channels) can be added by specifying the appropriate signature from Table 19, assigning the channels in numerical order and creating the tables.

The colour space used on the PCS side of a lut16Type tag (which may be either the input or output space, or both in the case of an Abstract profile) is identified by the PCS field in the profile header (see 7.2.7). This field does not distinguish between 8-bit and 16-bit PCS encodings. For the lut16Type tag, the 'Lab ' signature is defined to specify a legacy 16-bit PCSLAB encoding and the 'XYZ ' signature is defined to specify the 16-bit PCSXYZ encoding. Note that this definition only applies to the encoding used at the PCS side of the tag. The definition does NOT apply when these signatures are used in the data colour space field in the profile header (see 7.2.6), except in the case of an Abstract profile.

For colour values that are in the PCSLAB colour space on the PCS side of the tag, this tag uses the legacy 16-bit PCSLAB encoding defined in Tables 39 and 40, not the 16-bit PCSLAB encoding defined in 6.3.4.2. This encoding is retained for backwards compatibility with profile version 2. The PCSLAB L^* values have a different encoding than the PCSLAB a^* and PCSLAB b^* values.

The legacy PCSLAB L^* encoding is shown in Table 39.

Table 39 — Legacy PCSLAB L^* encoding

Value (PCSLAB L^*)	16-bit
0	0000h
100,0	FF00h
100 + (25 500/65 280)	FFFFh

Although the 16-bit encoding shown in Table 39 can represent values slightly greater than 100,0, these are not valid PCSLAB *L** values and they shall not be used.

The legacy PCSLAB *a** and PCSLAB *b** encoding is shown in Table 40.

Table 40 — Legacy PCSLAB *a or PCSLAB *b** encoding**

Value (PCSLAB *a** or PCSLAB *b**)	16-bit
−128,0	0000h
0	8000h
127,0	FF00h
127 + (255/256)	FFFFh

The 16-bit encoding can represent values slightly greater than 127,0, which are valid PCS values.

To convert colour values from this tag's legacy 16-bit PCSLAB encoding to the 16-bit PCSLAB encoding defined in 6.3.4.2 (Tables 12 and 13), multiply all values with 65 535/65 280 (i.e. FFFFh/FF00h). Any colour values that are in the value range of legacy 16-bit PCSLAB encoding, but not in the more recent 16-bit PCSLAB encoding, shall be clipped on a per-component basis. To convert colour values from the 16-bit PCSLAB encoding defined in 6.3.4.2 to this tag's legacy 16-bit PCSLAB encoding, divide all values by (65 535/65 280).

10.9 lut8Type

This structure represents a colour transform using tables of 8-bit precision. This type contains four processing elements: a 3 × 3 matrix (which shall be the identity matrix unless the input colour space is PCSXYZ), a set of one-dimensional input tables, a multi-dimensional lookup table, and a set of one-dimensional output tables. Data is processed using these elements via the following sequence:

(matrix) ⇒ (1d input tables) ⇒ (multi-dimensional lookup table, CLUT) ⇒ (1d output tables)

When used the byte assignment shall be as given in Table 41.

Table 41 — lut8Type encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	'mft1' (6D667431h) (multi-function table with 1‒byte precision) type signature	
4 to 7	4	Reserved, shall be set to 0	
8	1	Number of Input Channels (*i*)	uInt8Number
9	1	Number of Output Channels (*o*)	uInt8Number
10	1	Number of CLUT grid points (identical for each side) (*g*)	uInt8Number
11	1	Reserved for padding (required to be 00h)	
12 to 15	4	Encoded e1 parameter	s15Fixed16Number
16 to 19	4	Encoded e2 parameter	s15Fixed16Number
20 to 23	4	Encoded e3 parameter	s15Fixed16Number
24 to 27	4	Encoded e4 parameter	s15Fixed16Number

Table 41 (*continued*)

Byte position	**Field length** bytes	**Content**	**Encoded as**
28 to 31	4	Encoded e5 parameter	s15Fixed16Number
32 to 35	4	Encoded e6 parameter	s15Fixed16Number
36 to 39	4	Encoded e7 parameter	s15Fixed16Number
40 to 43	4	Encoded e8 parameter	s15Fixed16Number
44 to 47	4	Encoded e9 parameter	s15Fixed16Number
48 to 47+($256i$)	$256i$	Input tables	uInt8Number [...]
48+($256i$) to 47+($256i$)+($g^i o$)	$g^i o$	CLUT values	uInt8Number [...]
48+($256i$)+($g^i o$) to end	$256o$	Output tables	uInt8Number [...]

Explanation of symbols:

i is the number of input channels

o is the number of output channels

g is the number of grid points

The input and output tables, and CLUT, contained in a lut8Type each embodies a one-dimensional or multi-dimensional function which maps an input value in the “domain” of the function to an output value in the “range” of the function.

The domain of each of these tables is defined to consist of all real numbers between 0,0 and 1,0, inclusive. The first entry is located at 0,0, the last entry at 1,0, and intermediate entries are uniformly spaced using an increment of 1,0/(m^{-1}). For the input and output tables, m is 255. For the CLUT, m is the number of grid points along each dimension. The range of a function used to generate the contents of a table is likewise defined to be all real numbers between 0,0 and 1,0, inclusive. Since the domain and range of the tables are 0 to 1, it is necessary to convert all device values and PCSLAB values to this numeric range. It shall be assumed that the maximum value in each case is set to 1,0 and the minimum value to 0,0, and all intermediate values are linearly scaled accordingly.

Because the entries of a table are encoded as uInt8Numbers, it is necessary to round each real value to the nearest 8-bit integer.

Because the entries of lut8Type LUTs represent values in the range 0,0 to 1,0, encoded as uInt8Numbers, these entries should be divided by 255,0 for the calculation of the actual output values.

See Annex A for additional guidance on this topic.

The colour space used on the PCS side of a lut8Type tag (which may be either the input or output space, or both in the case of an Abstract profile) is identified by the PCS field in the profile header (see 7.2.7). This field does not distinguish between 8-bit and 16-bit PCS encodings. For the lut8Type tag, the 'Lab ' signature is defined to specify the 8-bit PCSLAB encoding. Note that this definition only applies to the encoding used as the PCS side of the tag. It does NOT apply when these signatures are used in the data colour space field in the profile header (see 7.2.6), except in the case of an Abstract profile.

An 8-bit PCSXYZ encoding has not been defined, so the interpretation of a lut8Type in a profile that uses PCSXYZ is implementation specific. Because of the resulting ambiguity and because an 8-bit linear quantization of PCSXYZ results in poor quality, it is recommended that the lut8Type tag not be used in profiles that employ PCSXYZ.

The matrix is organized as a 3 × 3 array. The dimension corresponding to the matrix rows varies least rapidly and the dimension corresponding to the matrix columns varies most rapidly and is shown in matrix form below.

$$\begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \tag{14}$$

When using the matrix of an Output profile, and the input data is PCSXYZ, then:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \square \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{15}$$

Each matrix entry is encoded as an s15Fixed16Number. The domain and range of the matrix is 0,0 to 1,0.

The matrix shall be an identity matrix unless the input is in the PCSXYZ colour space.

The input tables are arrays of uInt8Number values. Each input table consists of 256 uInt8Number integers. Each input table entry is appropriately normalized to the range 0 to 255. The inputTable is of size (InputChannels × 256) bytes. When stored in this tag, the one-dimensional lookup tables are packed one after another in the order described in Table 41.

The CLUT is organized as an i-dimensional array with a given number of grid points in each dimension, where i is the number of input channels (input tables) in the transform. The dimension corresponding to the first input channel varies least rapidly and the dimension corresponding to the last input channel varies most rapidly. Each grid point value is an o-byte array, where o is the number of output channels. The first sequential byte of the entry contains the function value for the first output function, the second sequential byte of the entry contains the function value for the second output function, and so on until all the output functions have been supplied. Each byte in the CLUT is appropriately normalized to the range 0 to 255. The size of the CLUT, in bytes, is $\mathit{GridPoints}^{\mathit{InputChannels}} \times \mathit{OutputChannels}$.

The output tables are arrays of uInt8Number values. Each output table consists of 256 uInt8Number integers. Each output table entry is appropriately normalized to the range 0 to 255. The outputTable is of size (OutputChannels × 256) bytes. When stored in this tag, the one-dimensional lookup tables are packed one after another in the order described in Table 41.

If the number of data points in a particular dimension of the CLUT, is two, the data for those points shall be set so that the correct results are obtained when linear interpolation is used to generate intermediate values.

When using this type, it is necessary to assign each data colour space component to an input and output channel. These assignments shall be as shown in Table 38. The channels are numbered according to the order in which their table occurs.

10.10 lutAToBType

10.10.1 General

This structure represents a colour transform. The type contains up to five processing elements which are stored in the AToBTag tag in the following order: a set of one-dimensional curves, a 3 × 3 matrix with offset terms, a set of one-dimensional curves, a multi-dimensional lookup table, and a set of one-dimensional output curves. Data are processed using these elements via the following sequence:

(“A” curves) ⇒ (multi-dimensional lookup table, CLUT) ⇒ (“M” curves) ⇒ (matrix) ⇒ (“B” curves).

NOTE The processing elements are not in this order in the tag to allow for simplified reading and writing of profiles.

It is possible to use any or all of these processing elements. At least one processing element shall be included. Only the following combinations are permitted:

- B;
- M, Matrix, B;
- A, CLUT, B;
- A, CLUT, M, Matrix, B.

Other combinations may be achieved by setting processing element values to identity transforms. The domain and range of the A and B curves and CLUT are defined to consist of all real numbers between 0,0 and 1,0 inclusive. The first entry is located at 0,0, the last entry at 1,0, and intermediate entries are uniformly spaced using an increment of 1,0/(m–1). For the A and B curves, m is the number of entries in the table. For the CLUT, m is the number of grid points along each dimension. Since the domain and range of the tables are 0,0 to 1,0 it is necessary to convert all device values and PCSLAB values to this numeric range. It shall be assumed that the maximum value in each case is set to 1,0 and the minimum value to 0,0 and all intermediate values are linearly scaled accordingly.

When using this type, it is necessary to assign each data colour space component to an input and output channel. This assignment is specified in Table 38.

When used the byte assignment and encoding shall be as given in Table 42.

Table 42 — lutAToBType encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	'mAB ' (6D414220h) [multi-function A-to-B table] type signature	
4 to 7	4	Reserved, shall be set to 0	
8	1	Number of Input Channels (i)	uInt8Number
9	1	Number of Output Channels (o)	uInt8Number
10 to 11	2	Reserved for padding, shall be set to 0	
12 to 15	4	Offset to first "B" curve	uInt32Number
16 to 19	4	Offset to matrix	uInt32Number
20 to 23	4	Offset to first "M" curve	uInt32Number
24 to 27	4	Offset to CLUT	uInt32Number
28 to 31	4	Offset to first "A" curve	uInt32Number
32 to end	Variable	Data	

Each curve and processing element shall start on a 4-byte boundary. To achieve this, each item shall be followed by up to three 00h pad bytes as needed.

It is permitted to share curve data elements. For example, the offsets for A, B and M curves can be identical.

The offset entries (bytes 12 to 31) point to the various processing elements found in the tag. The offsets indicate the number of bytes from the beginning of the tag to the desired data. If any of the offsets are zero, i.e. an indication that processing element is not present and the operation is not performed.

This tag type may be used independent of the value of the PCS field specified in the header.

10.10.2 “A” curves

There are the same number of “A” curves as there are input channels. The “A” curves may only be used when the CLUT is used. The curves are stored sequentially, with 00h bytes used for padding between them if needed. Each “A” curve is stored as an embedded curveType or a parametricCurveType (see 10.5 or 10.16). The length is as indicated by the convention of the respective curve type. Note that the entire tag type, including the tag type signature and reserved bytes, is included for each curve.

10.10.3 CLUT

The CLUT appears as an n-dimensional array, with each dimension having a number of entries corresponding to the number of grid points.

The CLUT values are arrays of 8-bit or 16-bit unsigned values, normalized to the range of 0 to 255 or 0 to 65 535.

The CLUT is organized as an i-dimensional array with a variable number of grid points in each dimension, where i is the number of input channels in the transform. The dimension corresponding to the first channel varies least rapidly and the dimension corresponding to the last input channel varies most rapidly. Each grid point value is an o-integer array, where o is the number of output channels. The first sequential integer of the entry contains the function value for the first output function, the second sequential integer of the entry contains the function value for the second output function and so on until all of the output functions have been supplied. The size of the CLUT in bytes is (nGrid1 × nGrid2 ×…× nGridN) × number of output channels (o) × size of (channel component).

When used the byte assignment and encoding for the CLUT shall be as given in Table 43.

Table 43 — lutAToBType CLUT encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 15	16	Number of grid points in each dimension. Only the first i entries are used, where i is the number of input channels. Unused entries shall be set to 00h.	uInt8Number[16]
16	1	Precision of data elements in bytes. Shall be either 01h or 02h.	uInt8Number
17 to 19	3	Reserved for padding, shall be set to 0	
20 to end	Variable	CLUT data points (arranged as described in the text).	uInt8Number [...] or uInt16Number [...]

If the number of input channels does not equal the number of output channels, the CLUT shall be present.

If the number of grid points in a one-dimensional curve, or in a particular dimension of the CLUT, is two, the data for those points shall be set so that the correct results are obtained when linear interpolation is used to generate intermediate values.

10.10.4 “M” curves

There are the same number of “M” curves as there are output channels. The curves are stored sequentially, with 00h bytes used for padding between them if needed. Each “M” curve is stored as an embedded curveType or a parametricCurveType (see 10.5 or 10.16). The length is as indicated by the convention of the respective curve type. Note that the entire tag type, including the tag type signature and reserved bytes, is included for each curve. The “M” curves may only be used when the matrix is used.

10.10.5 Matrix

The matrix is organized as a 3 × 4 array. The elements appear in order from e_1–e_{12}. The matrix elements are each s15Fixed16Numbers.

$$\text{array} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} \end{bmatrix} \tag{16}$$

The matrix is used to convert data to a different colour space, according to the following equation:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \square \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} e_{10} \\ e_{11} \\ e_{12} \end{bmatrix} \tag{17}$$

The range of input values X_1, X_2 and X_3 is 0,0 to 1,0. The resultant values Y_1, Y_2 and Y_3 shall be clipped to the range 0,0 to 1,0 and used as inputs to the "B" curves.

10.10.6 "B" curves

There are the same number of "B" curves as there are output channels. The curves are stored sequentially, with 00h bytes used for padding between them if needed. Each "B" curve is stored as an embedded curveType or a parametricCurveType (see 10.5 or 10.16). The length is as indicated by the convention of the respective curve type. Note that the entire tag type, including the tag type signature and reserved bytes, are included for each curve.

10.11 lutBToAType

10.11.1 General

This structure represents a colour transform. The type contains up to five processing elements which are stored in the BToATag in the following order: a set of one-dimensional curves, a 3 × 3 matrix with offset terms, a set of one-dimensional curves, a multi-dimensional lookup table, and a set of one-dimensional curves. Data are processed using these elements via the following sequence:

("B" curves) ⇒ (matrix) ⇒ ("M" curves) ⇒ (multi-dimensional lookup table, CLUT) ⇒ ("A" curves).

It is possible to use any or all of these processing elements. At least one processing element shall be included. Only the following combinations are permitted:

- B;
- B, Matrix, M;
- B, CLUT, A;
- B, Matrix, M, CLUT, A.

Other combinations may be achieved by setting processing element values to identity transforms. The domain and range of the A and B curves and CLUT are defined to consist of all real numbers between 0,0 and 1,0 inclusive. The first entry is located at 0,0, the last entry at 1,0, and intermediate entries are uniformly spaced using an increment of 1,0/(m-1). For the A, M and B curves m is the number of entries in the table. For the CLUT m is the number of grid points along each dimension. Since the domain and range of the tables are 0,0 to 1,0 it is necessary to convert all device values and PCSLAB values to this numeric range. It shall be assumed that the maximum value in each case is set to 1,0 and the minimum value to 0,0 and all intermediate values are linearly scaled accordingly.

When using this type, it is necessary to assign each data colour space component to an input and output channel. This assignment is specified in Table 38.

When used the byte assignment and encoding shall be as given in Table 44.

Table 44 — lutBToAType encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	‘mBA’ (6D424120h) [multi-function BToA table] type signature	
4 to 7	4	Reserved, shall be set to 0	
8	1	Number of Input Channels (i)	uInt8Number
9	1	Number of Output Channels (o)	uInt8Number
10 to 11	2	Reserved for padding, shall be set to 0	
12 to 15	4	Offset to first “B” curve	uInt32Number
16 to 19	4	Offset to matrix	uInt32Number
20 to 23	4	Offset to first “M” curve	uInt32Number
24 to 27	4	Offset to CLUT	uInt32Number
28 to 31	4	Offset to first “A” curve	uInt32Number
32 to end	Variable	Data	

Each curve and processing element shall start on a 4-byte boundary. To achieve this, each item may be followed by up to three 00h pad bytes as needed.

Curve data elements may be shared. For example, the offsets for A, B and M curves may be identical.

The offset entries (bytes 12 to 31) point to the various processing elements found in the tag. The offsets indicate the number of bytes from the beginning of the tag to the desired data. If any of the offsets are zero, i.e. an indication that processing element is not present and the operation is not performed.

This tag type shall only be used when the PCS field in the header specifies either PCSXYZ or PCSLAB.

10.11.2 “B” curves

There are the same number of “B” curves as there are input channels. The curves are stored sequentially, with 00h bytes used for padding between them if needed. Each “B” curve is stored as an embedded curveType tag or a parametricCurveType (see 10.5 or 10.16). The length is as indicated by the convention of the proper curve type. Note that the entire tag type, including the tag type signature and reserved bytes, is included for each curve.

10.11.3 Matrix

The matrix is organized as a 3 × 4 array. The elements of the matrix appear in the type in order from e_1 to e_{12}. The matrix elements are each s15Fixed16Numbers.

$$\text{array} = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}] \tag{18}$$

The matrix is used to convert data to a different colour space, according to the following equation:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \square \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} e_{10} \\ e_{11} \\ e_{12} \end{bmatrix} \tag{19}$$

The range of input values X_1, X_2 and X_3 is 0,0 to 1,0. The resultant values Y_1, Y_2 and Y_3 shall be clipped to the range 0,0 to 1,0 and used as inputs to the “M” curves.

The matrix is permitted only if the number of output channels, or “M” curves, is 3.

10.11.4 “M” curves

There are the same number of “M” curves as there are input channels. The curves are stored sequentially, with 00h bytes used for padding between them if needed. Each “M” curve is stored as an embedded curveType or a parametricCurveType (see 10.5 or 10.16). The length is as indicated by the convention of the proper curve type. Note that the entire tag type, including the tag type signature and reserved bytes, are included for each curve. The “M” curves may only be used when the matrix is used.

10.11.5 CLUT

The CLUT appears as an n-dimensional array, with each dimension having a number of entries corresponding to the number of grid points.

The CLUT values are arrays of 8-bit or 16-bit unsigned values, normalized to the range of 0 to 255 or 0 to 65 535.The CLUT is organized as an i-dimensional array with a variable number of grid points in each dimension, where i is the number of input channels in the transform. The dimension corresponding to the first channel varies least rapidly and the dimension corresponding to the last input channel varies most rapidly. Each grid point value is an o-integer array, where o is the number of output channels. The first sequential integer of the entry contains the function value for the first output function, the second sequential integer of the entry contains the function value for the second output function and so on until all of the output functions have been supplied. The size of the CLUT in bytes is (nGrid1 × nGrid2 ×…× nGridN) × number of output channels (o) × size of (channel component).

When used the byte assignment and encoding for the CLUT shall be as given in Table 45.

Table 45 — lutBToAType CLUT encoding

Byte position	Field length bytes	Content	Encoded as
0 to 15	16	Number of grid points in each dimension. Only the first i entries are used, where i is the number of input channels. Unused entries shall be set to 00h.	uInt8Number[16]
16	1	Precision of data elements in bytes. Shall be either 01h or 02h.	uInt8Number
17 to 19	3	Reserved for padding.	
20 to end	Variable	CLUT data points (arranged as described in the text).	uInt8Number [...] or uInt16Number [...]

If the number of grid points in a one-dimensional curve, or in a particular dimension of the CLUT, is two, the data for those points shall be set so that the correct results are obtained when linear interpolation is used to generate intermediate values.

If the number of input channels does not equal the number of output channels, the CLUT shall be present.

10.11.6 “A” curves

There are the same number of “A” curves as there are output channels. The “A” curves may only be used when the CLUT is used. The curves are stored sequentially, with 00h bytes used for padding between them if needed. Each “A” curve is stored as an embedded curveType or a parametricCurveType (see 10.5 or 10.16). The length is as indicated by the convention of the proper curve type. Note that the entire tag type, including the tag type signature and reserved bytes, is included for each curve.

10.12 measurementType

The measurementType information refers only to the internal profile data and is meant to provide profile makers an alternative to the default measurement specifications. When used the byte assignment and encoding shall be as given in Table 46.

Table 46 — measurementType structure

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	'meas' (6D656173h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Encoded value for standard observer	see Table 47
12 to 23	12	nCIEXYZ tristimulus values for measurement backing	XYZNumber
24 to 27	4	Encoded value for measurement geometry	see Table 48
28 to 31	4	Encoded value for measurement flare	see Table 49
32 to 35	4	Encoded value for standard illuminant	see Table 50

The encoding for the standard observer field is shown in Table 47.

Table 47 — Standard observer encodings

Standard observer	Hex encoding
Unknown	00000000h
CIE 1931 standard colorimetric observer	00000001h
CIE 1964 standard colorimetric observer	00000002h

The encoding for the measurement geometry field is shown in Table 48.

Table 48 — Measurement geometry encodings

Geometry	Hex encoding
Unknown	00000000h
0°:45° or 45°:0°	00000001h
0°:d or d:0°	00000002h

The encoding for the measurement flare value is shown in Table 49, and is equivalent to the basic numeric type u16Fixed16Number in 4.7.

Table 49 — Measurement flare encodings

Flare	Hex encoding
0 (0 %)	00000000h
1,0 (or 100 %)	00010000h

The encoding for the standard illuminant field is shown in Table 50.

Table 50 — Standard illuminant encodings

Standard illuminant	Hex encoding
Unknown	00000000h
D50	00000001h
D65	00000002h
D93	00000003h
F2	00000004h
D55	00000005h
A	00000006h
Equi-Power (E)	00000007h
F8	00000008h

10.13 multiLocalizedUnicodeType

This tag structure contains a set of records each referencing a multilingual Unicode string associated with a profile. Each string is referenced in a separate record with the information about what language and region the string is for.

The byte assignment and encoding shall be as given in Table 51.

Note that the fourth field of this tag, the record size should, for the time being, contain the value 12, which corresponds to the size in bytes of each record. Any code that needs to access the *n*th record should determine the record's offset by multiplying *n* by the contents of this size field and adding 16. This minor extra effort allows for future expansion of the record encoding, should the need arise, without having to define a new tag type.

Multiple strings within this tag may share storage locations. For example, en/US and en/UK can refer to the same string data.

For the specification of Unicode, see The Unicode Standard published by The Unicode Consortium or visit their website at http://www.unicode.org. For the definition of language codes and country codes, see respectively ISO 639-1 and ISO 3166-1. The Unicode strings in storage should be encoded as 16-bit big-endian, UTF-16BE, and should not be NULL terminated.

NOTE For additional clarification on the encodings used, see the ICC technical note 01-2002 available on www.color.org.

If the specific record for the desired region is not stored in the tag, the record with the same language code should be used. If the specific record for the desired language is not stored in the tag, the first record in the tag is used if no other user preference is available.

Table 51 — multiLocalizedUnicodeType

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	‘mluc’ (0x6D6C7563) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Number of records (n)	uInt32Number
12 to 15	4	Record size: the length in bytes of every record. The value is 12.	0000000Ch
16 to 17	2	First record language code: in accordance with the language code specified in ISO 639-1	uInt16Number
18 to 19	2	First record country code: in accordance with the country code specified in ISO 3166-1	uInt16Number
20 to 23	4	First record string length: the length in bytes of the string	uInt32Number
24 to 27	4	First record string offset: the offset from the start of the tag to the start of the string, in bytes	uInt32Number
28 to 28+[12(n–1)]–1 (or 15+12n)	12(n–1)	Additional records as needed	
28+[12(n–1)] (or 16+12n) to end	Variable	Storage area of strings of Unicode characters	

10.14 multiProcessElementsType

10.14.1 General

This structure represents a colour transform, containing a sequence of processing elements. The processing elements contained in the structure are defined in the structure itself, allowing for a flexible structure. Currently supported processing elements are: a set of one dimensional curves, a matrix with offset terms, and a multidimensional lookup table (CLUT). Other processing element types may be added in the future. Each type of processing element may be contained any number of times in the structure. The processing elements support float32Number-encoded input and output ranges.

If undefined processing element types are present in a multiProcessElementsType tag, the multiProcessElementsType tag shall not be used and fall back behaviour shall be followed.

When using this type, it is necessary to assign each colour space component to an input and output channel. These assignments shall be as shown in Table 38.

When used, the byte assignment and encoding shall be as given in Table 52.

Table 52 — multiProcessElementsType encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	‘mpet’ (6D706574h) [multi-process elements table] type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of input channels	uInt16Number
10 to 11	2	Number of output channels	uInt16Number
12 to 15	2	Number of processing elements (n)	uInt32Number
16 to 15+8n	8n	Process element positions table	positionNumber[...]
16+8n to end		Data	

The number of processing elements (n) shall be greater than or equal to 1. The process element positions table contains information on where and how large the process elements are. Offset locations are relative to the start of the multiProcessElementsType tag. Thus the offset of first stored process element shall be $16+8n$.

Each processing element shall start on a 4-byte boundary. To achieve this, each item shall be followed by up to three 00h pad bytes as needed.

It is permitted to share data between processing elements. For example, the offsets for some processing elements can be identical.

10.14.2 multiProcessElementsType Elements

10.14.2.1 General

Processing elements in the multiProcessElementsType are processed in the order that they are defined in the processing elements position table. The results of a processing element are passed on to the next processing element. The last processing element provides the final result for the containing multiProcessElementsType. Therefore, the input/output channels specified by the processing elements and the containing multiProcessElementsType need to be in agreement.

The first processing element's input channels shall be the same as the input channels of the containing multiProcessElementsType. The input channels of a processing element shall be the same as the previous processing element's output channels. The last processing element's output channels shall be the same as the output channels of the containing multiProcessElementsType.

Clipping of the results of a processing element shall not be performed. Some processing elements may perform clipping as needed on input.

The specification for each processing element will indicate whether that element will perform clipping on input.

The general element encoding for multiProcessElementsType elements is shown in Table 53.

Table 53 — General element encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	Element signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of input channels (p)	uInt16Number
10 to 11	2	Number of output channels (q)	uInt16Number
12 to end		Data	

10.14.2.2 Curve Set element

The Curve Set element encodes multiple one dimensional curves. The encoding is shown in Table 54.

Table 54 — Curve Set element encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	‘cvst’ (6D666C74h) [curve set element table] type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of input channels (p)	uInt16Number
10 to 11	2	Number of output channels (q)	uInt16Number
12 to 11+8p	8p	Curve positions (offset and size)	positionNumber[…]
12+8p to end		Data	

Encoding values for both input and output channels is for consistency with other processing elements. Since each one dimensional curve maps a single input to a single output the number of outputs will be the same as the number of inputs. Thus, the number of output channels (q) shall be set to the same value as the number of input channels (p).

The output value for an input shall be specified by the first segment in the segment list that contains that input. Successive break-points shall not be decreasing.

Each channel shall have a curve position element. Offset locations are relative to the start of the containing curveSetElement. Thus the offset of first stored curve in the curve set shall be 12+8p.

The one-dimensional curves are stored sequentially. Each curve shall start on a 4-byte boundary. To achieve this, each curve shall be followed by up to three 00h pad bytes as needed.

It is permitted to share data between one dimensional curves. For example, the offsets for some one dimensional curves can be identical.

Each curve is stored in one or more curve segments, with break-points specified between curve segments. The first curve segment always starts at $-\infty$, and the last curve segment always ends at $+\infty$. The first and last curve segments shall be specified in terms of a formula, whereas the other segments shall be specified either in terms of a formula, or by a sampled curve.

If a curve has a single curve segment, no break-points shall be specified, and the curve shall be specified in terms of a formula.

If a curve has more than one curve segment, break-points shall be specified between curve segments. If there are n segments, n–1 break-points are specified. The encoding for such a curve is shown in Table 55.

Table 55 — One-dimensional curves encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	‘curf’ (63757266h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of segment(s) (n)	uInt16Number
10 to 11	2	Reserved, shall be set to 0	
12 to 4n+7	4 × (n–1)	n–1 break-points	float32Number[…]
4n+8 to end		Segments 1 to n	

Break-points separate two curve segments. The first curve segment is defined between $-\infty$ and break-point 1 (included). The kth curve segment (k in the range 2 to n–1) is defined between the break-point k–1 (not included) and the break-point k (included). The nth curve-segment is defined between break-point n–1 (not included) and $+\infty$. Curve segments that are specified in terms of a formula shall be encoded as shown in Table 56.

Table 56 — Formula curve segments encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘parf’ (70617266h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Encoded value of the function type	uInt16Number
10 to 11	2	Reserved, shall be set to 0	
12 to end	See Table 57	Parameters (see Table 57)	float32Number[…]

The encoding for the function type field and the parameters is shown in Table 57.

Table 57 — Formula curve segments encoding

Field length bytes	**Function type**	**Encoded value**	**Parameters**
16	$Y = (aX + b)^{\gamma} + c$	0000h	γ, a, b, c
20	$Y = a\log_{10}(bX^{\gamma} + c) + d$	0001h	γ, a, b, c, d
20	$Y = ab^{cX+d} + e$	0002h	a, b, c, d, e

The functional inputs and outputs are defined over the values that can be represented as float32Number. The curve-segment shall be defined to result in float32Number values for the entire curve-segment.

Curve segments that are specified as sampled curves shall be encoded as shown in Table 58.

Table 58 — Sampled curve segment encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘samf’ (73616D66h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Count (n) specifying the number of entries that follow	uInt32Number
12 to end	4n	Curve entries	float32Number[…]

The count (n) shall be greater than or equal to 1.

The curve samples shall be equally-spaced within the segment, and shall include one break-point, as previously described. If the sampled curve represents the curve-segment between break-point k ($B_{\mathrm{P}k}$) and break-point k+1 ($B_{\mathrm{P},k+1}$), the jth sample ($j \in [1, n)$) shall correspond to the input value $B_{\mathrm{P},k} + j\,(B_{\mathrm{P},k+1} - B_{\mathrm{P},k})\,/\,n$. Thus $B_{\mathrm{P},k}$ is excluded.

NOTE The first point used for interpolation of a sampled curve segment is not directly stored in a sampled curve segment.

The value to interpolate from for the sampled curve at point $B_{\mathrm{P},k}$ shall be defined by the output of the immediately preceding curve segment in the curve segment list at the point $B_{\mathrm{P},k}$.

If the number of grid points in a particular segment of a one-dimensional curve is one, the data generated for intermediate values shall be set so that the correct results are obtained when linear interpolation is used.

10.14.2.3 Matrix element

The matrix is organized as an array of $p \times q$ elements, where p is the number of input channels to the matrix, and q is the number of output channels. The matrix elements are each float32Numbers. The array is organized as follows:

$$\text{array} = [e_{11}, e_{12}, \ldots, e_{1p}, e_{21}, e_{22}, \ldots, e_{2p}, \ldots, e_{q1}, e_{q2}, \ldots, e_{qp}, e_1, e_2, \ldots, e_q] \qquad (20)$$

The matrix element encoding is shown in Table 59.

Table 59 — Matrix element encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘matf’ (6D617466h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of input channels (p)	uInt16Number
10 to 11	2	Number of output channels (q)	uInt16Number
12 to end	$4q(p+1)$	Matrix elements	float32Number[…]

The matrix is used to convert data to a different colour space, according to Equation (21):

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \cdots \\ Y_q \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1p} \\ e_{21} & e_{22} & \cdots & e_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ e_{q1} & e_{q2} & & e_{qp} \end{bmatrix} \Box \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_p \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \cdots \\ e_q \end{bmatrix} \qquad (21)$$

The range of the input values X_1, X_2, …, X_p and output values Y_1, Y_2, …, Y_q is the range of values that can be represented as float32Number.

10.14.2.4 CLUT element

The CLUT appears as an n-dimensional array, with each dimension having a number of entries corresponding to the number of grid points.

The CLUT values are arrays of float32Number.

The CLUT is organized as an p-dimensional array with a variable number of grid points in each dimension, where p is the number of input channels in the transform. The dimension corresponding to the first channel varies least rapidly and the dimension corresponding to the last input channel varies most rapidly. Each grid point value is a q-float32Number array, where q is the number of output channels. The first sequential float32Number of the entry contains the function value for the first output function, the second sequential float32Number of the entry contains the function value for the second output function and so on until all of the output functions have been supplied. Equation (22) gives the computation for the byte size of the CLUT.

$$\text{nGrid1} \times \text{nGrid2} \times \ldots \times \text{nGridp} \times \text{number of output channels } (q) \times 4 \qquad (22)$$

When used, the byte assignment and encoding for the CLUT shall be as given in Table 60.

Table 60 — CLUT element encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘clut’ (636C7574h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of input channels (p)	uInt16Number
10 to 11	2	Number of output channels (q)	uInt16Number
12 to 27	16	Number of grid points in each dimension. Only the first p entries are used, where p is the number of input channels. Unused entries shall be set to 00h.	uInt8Number
28 to end	See Equation (22)	CLUT data points (arranged as described in the text)	float32Number[…]

The input range for the CLUT is 0,0 to 1,0. For any input value outside this range, the nearest range limit value shall be the input value. The range of the Output Channels is the range of values that can be represented as float32Number.

If the number of grid points in a particular dimension of the CLUT is two, the data for those points shall be set so that the correct results are obtained when linear interpolation is used to generate intermediate values. CLUT elements require a minimum of two grid points for each dimension.

10.14.2.5 Future Expansion elements

The ‘bACS’ and ‘eACS’ element types are provided as placeholders for future expansion. If present, these elements shall be considered as pass through elements with no modification of channel data. Their encoding shall be as shown in Table 61 and 62.

Table 61 — bACS element encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘bACS‘ (62414353h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of input channels (p)	uInt16Number
10 to 11	2	Number of output channels (q)	uInt16Number
12 to 15	4	Signature	

Table 62 — eACS element encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘eACS‘ (65414353h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of input channels (p)	uInt16Number
10 to 11	2	Number of output channels (q)	uInt16Number
12 to 15	4	Signature	

For both the ‘bACS’ and ‘eACS’ element types the number of input channels (p) shall be the same as the number of output channels (q).

10.15 namedColor2Type

The namedColor2Type is a count value and array of structures that provide colour coordinates for 7-bit ASCII colour names. For each named colour, a PCS and optional device representation of the colour are given. Both representations are 16-bit values and PCS values shall be relative colorimetric. The device representation corresponds to the header’s “data colour space” field. This representation should be consistent with the “number of device coordinates” field in the namedColor2Type. If this field is 0, device coordinates are not provided. The PCS representation corresponds to the header’s PCS field. The PCS representation is always provided. Colour names are fixed-length, 32-byte fields including null termination. In order to maintain maximum portability, it is strongly recommended that special characters of the 7-bit ASCII set not be used.

When used the byte assignment and encoding shall be as given in Table 63.

Table 63 — namedColor2Type encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	‘ncl2’ (6E636C32h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Vendor specific flag (least-significant 16 bits reserved for ICC use)	
12 to 15	4	Count of named colours (n)	uInt32Number
16 to 19	4	Number of device coordinates (m) for each named colour	uInt32Number
20 to 51	32	Prefix for each colour name (32-byte field including null termination)	7-bit ASCII
52 to 83	32	Suffix for each colour name (32-byte field including null termination)	7-bit ASCII
84 to 115	32	First colour root name (32-byte field including null termination)	7-bit ASCII
116 to 121	6	First named colour’s PCS coordinates. The encoding is the same as the encodings for the PCS colour spaces as described in 6.3.4.2 and 10.8. Only PCSXYZ and legacy 16-bit PCSLAB encodings are permitted. PCS values shall be relative colorimetric.	uInt16Number [3]
122 to 121+(2m)	2m	First named colour’s device coordinates. For each coordinate, 0000h represents the minimum value for the device coordinate and FFFFh represents the maximum value for the device coordinate. The number of coordinates is given by the “number of device coordinates” field. If the “number of device coordinates” field is 0, this field is not given.	uInt16Number [...]
122+(2m) to end	(n–1) (38+2m)	If $n > 1$ the remaining n–1 colours are described in a manner consistent with the first named colour, see byte offsets 84 to 121+(2m).	

For colour values that are in PCSLAB, this tag uses the legacy 16-bit PCSLAB encoding defined in 10.8 (Tables 39 and 40), not the 16-bit PCSLAB encoding that is defined in 6.3.4.2 (Tables 12 and 13). This encoding is retained for backwards compatibility with profile version 2. The PCSLAB L^* values have a different encoding than the PCSLAB a^* and PCSLAB b^* values. The 16-bit PCSLAB L^* encoding shall be as shown in Table 39 and the PCSLAB a^* and PCSLAB b^* 16-bit encoding shall be as shown in Table 40. Note that though the 16-bit PCSLAB L^* encoding can represent values slightly greater than 100,0, these are not valid PCSLAB L^* values and they should not be used. The 16-bit PCSLAB a^* and PCSLAB b^* encoding can represent values slightly greater than 127,0 which are valid PCS values.

To convert colour values from this tag's legacy 16-bit PCSLAB encoding to the 16-bit PCSLAB encoding defined in 6.3.4.2 (Tables 12 and 13), multiply all values with (65 535/65 280) (i.e. FFFFh/FF00h). Any colour values that are in the value range of legacy 16-bit PCSLAB, but not in the more recent 16-bit PCSLAB encoding, shall be clipped on a per-component basis. To convert colour values from the 16-bit PCSLAB encoding defined in 6.3.4.2 (Tables 12 and 13) to this tag's legacy 16-bit PCSLAB encoding, divide all values by (65 535/65 280).

NOTE The parameters selected for a parametric curve can result in complex or undefined values for the input range used. This can occur for example if $g \Leftarrow 0$, $a \Leftarrow 0$ or $d < -b/a$ In such cases, the behaviour of the curve is undefined.

10.16 parametricCurveType

The parametricCurveType describes a one-dimensional curve by specifying one of a predefined set of functions using the parameters. When used the byte assignment and encoding shall be as given in Table 64.

Table 64 — parametricCurveType encoding

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	'para' (70617261h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Encoded value of the function type	uInt16Number (see Table 65)
10 to 11	2	Reserved, shall be set to 0	
12 to end	See Table 65	One or more parameters (see Table 65)	s15Fixed16Number [...]

The encoding for the function type field and the parameters are shown in Table 65.

Table 65 — parametricCurveType function type encoding

Field length bytes	Function type	Encoded value	Parameters	Note
4	$Y = X^g$	0000h	g	
12	$Y = (aX + b)^g \quad (X \geqslant -b/a)$ $Y = 0 \quad (X > -b/a)$	0001h	$g\ a\ b$	CIE 122-1996
16	$Y = (aX + b)^g + c \quad (X \geqslant -b/a)$ $Y = c \quad (X > -b/a)$	0002h	$g\ a\ b\ c$	IEC 61966-3
20	$Y = (aX + b)^g \quad (X \geqslant d)$ $Y = cX \quad (X > d)$	0003h	$g\ a\ b\ c\ d$	IEC 61966-2-1 (sRGB)
28	$Y = (aX + b)^g + c \quad (X \geqslant d)$ $Y = (cX + f) \quad (X > d)$	0004h	$g\ a\ b\ c\ d\ e\ f$	
NOTE More functions can be added as necessary.				

The order of the parameters in the tag data, Table 64, follows the left-to-right order of the parameters in Table 65.

The domain and range of each function shall be [0,0 1,0]. Any function value outside the range shall be clipped to the range of the function. When unsigned integer data is supplied as input, it shall be converted to the domain by dividing it by a factor of $(2^{N})-1$, where N is the number of bits used to represent the input data. When unsigned integer data is required as output, it shall be converted from the range by multiplying it by a factor of $(2^{M})-1$, where M is the number of bits used to represent the output data.

If the input is PCSXYZ, the PCSXYZ X, Y, or Z value 1+ (32 767/32 768) shall be mapped to the function input value 1,0. If the output is PCSXYZ, the function output value 1,0 shall be mapped to the PCSXYZ X, Y, or Z value 1+(32 767/32 768).

NOTE The parameters selected for a parametric curve can result in complex or undefined values for the input range used. This can occur, for example, if $d < -b/a$. In such cases the behaviour of the curve is undefined.

10.17 profileSequenceDescType

This type is an array of structures, each of which contains information from the header fields and tags from the original profiles which were combined to create the final profile. The order of the structures is the order in which the profiles were combined and includes a structure for the final profile. This provides a description of the profile sequence from source to destination, typically used with the DeviceLink profile.

When used the byte assignment and structure shall be as given in Table 66.

Table 66 — profileSequenceDescType structure

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘pseq’ (70736571h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to 11	4	Count value specifying number of description structures in the array
12 to end	Variable	Count profile description structures, see Table 67

Each profile description structure has the format shown in Table 67.

Table 67 — Profile description structure

Byte position	**Field length bytes**	**Content**
0 to 3	4	Device manufacturer signature (from corresponding profile’s header)
4 to 7	4	Device model signature (from corresponding profile’s header)
8 to 15	8	Device attributes (from corresponding profile’s header)
16 to 19	4	Device technology information such as CRT, dye sublimation, etc. (corresponding to profile’s technology signature)
20 to m	Variable	Displayable description of device manufacturer (profile’s deviceMfgDescTag)
m+1 to n	Variable	Displayable description of device model (profile’s deviceModelDescTag)

If the deviceMfgDescTag and/or deviceModelDescTag is not present in a component profile, then a “placeholder” tag should be inserted. This tag should have a 0 in the number of names field in the multiLocalizedUnicodeType structure with no name record or strings.

Also note that the entire tag, including the tag type, should be stored.

If the technologyTag is not present, bytes 16 to 19 should be 00000000h.

10.18 profileSequenceIdentifierType

10.18.1 General

This type is an array of structures, each of which contains information for identification of a profile used in a sequence.

When used, the byte assignment and encoding shall be as given in Table 68.

Table 68 — profileSequenceIdentifierType structure

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘psid’ (70736964h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 11	4	Count (n), specifying number of structures in the array	uInt32Number
12 to 11+8n	8n	Positions table for profile identifiers	positionNumber[…]
12+8n to end		Profile identifier structures, see Table 69	

The offsets stored in the positions table shall be relative to the start of the tag.

It is permitted for profile identifier structures to be shared. So it is possible that a positionNumber in the positions table is identical to another positionNumber in the positions table.

Each profile identifier structure shall start on a 4-byte boundary. To achieve this, each structure shall be followed by up to three 00h pad bytes as needed.

Each profile identifier structure has the format shown in Table 69.

Table 69 — Profile identifier structure

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 15	16	Profile ID (see below)	See 7.2.18
16 to end	variable	Profile Description (see below)	multiLocalizedUnicodeType

10.18.2 Profile ID

If a profile contains a Profile ID in the Profile Header, it shall be used in the Profile Identifier structure. If a profile does not contain a Profile ID in the Profile Header, either an all-zero Profile ID or a computed Profile ID shall be used in the Profile Identifier structure.

10.18.3 Profile Description

For profiles conforming to ICC Profile Specification ICC.1:2001-12 (ICC V4.0.0) and later, the entire multiLocalizedUnicodeType contents of the Profile Description Tag shall be included in the Profile Identifier structure. For profiles conforming to ICC Profile Specification ICC.1:2001-04 (ICC V2.4.0) and earlier, the contents of the textDescriptionType Profile Description Tag shall be converted to multiLocalizedUnicodeType and used in the Profile Identifier structure.

NOTE One way of creating a multiLocalizedUnicodeType from a textDescriptionType is by converting the 7-bit ASCII part of the textDescriptionType to a 'enUS' Unicode string by mapping the 7-bit ASCII characters to 16-bit Unicode characters, and storing the 'enUS' Unicode string in the multiLocalizedUnicodeType.

10.19 responseCurveSet16Type

ICC profiles for display and output devices will produce the desired colour only while the device has a particular relationship between normalized device codes and physical colorant amount (the reference response). If the response of the device changes (the current response), the profile will no longer produce the correct result. In many cases it is impractical to produce a new profile for the current response, but the change can be compensated for by modifying the single channel device codes.

The purpose of this tag type is to provide a mechanism to relate physical colorant amounts with the normalized device codes produced by lut8Type, lut16Type, lutAToBType, lutBToAType or multiProcessElementsType tags so that corrections can be made for variation in the device without having to produce a new profile. The mechanism can be used by applications to allow users with relatively inexpensive and readily available instrumentation to apply corrections to individual output colour channels in order to achieve consistent results.

Two pieces of information are necessary for this compensation, the reference response and the current response. This tag type provides a mechanism that allows applications that create profiles to specify the reference response. The way in which applications determine and make use of the current response is not specified at this time.

The measurements are of the standard variety used in the photographic, graphic arts, and television industries for process control. The measurements are intended to represent colorant amounts and so different measurement techniques are appropriate for different device types.

It is the job of the profile creator to provide reference response data in as many measurement units as practical and appropriate so that applications may select the same units that are measured by the user’s instrument. Since it is not possible in general to translate between measurement units, and since most instruments only measure in one unit, providing a wide range of measurement units is vital. The profile originator shall decide which measurement units are appropriate for the device.

Here are some examples of suitable measurement units.

- For process colours, density should be reported.
- Red-filter density should be reported for the cyan channel.
- Green-filter should be reported for the magenta channel.
- Blue-filter should be reported for the yellow channel.
- Visual should be reported for the black channel.
- For other colorants, such as spot colours or Hi-Fi colours, it is the responsibility of the profile creator to select the appropriate units of measure for the system being profiled. Several different density standards are used around the world, so it is important that profile creators report in as many different density units as possible (See Table 72). Examples of suitable density measurements are:
  - Status T,
  - Status E,
  - Status I, and
  - DIN.

Normalized device codes resulting from a lut8Type tag should first be multiplied by 257 (101h). Normalized device codes in the 0 to 1 range that are related to lutAToBType and lutBToAType tags should be encoded in a responseCurveSet16Type as 16-bit values converted to the range of 0 to 65,535 (FFFFh). Normalized device codes clipped to the 0 to 1 range that are related to multiProcessElementsType tags should be encoded in a responseCurveSet16Type as 16-bit values converted to the range of 0 to 65,535 (FFFFh).

For those fields that have been structured in arrays of channel data, the channels are ordered as specified for the appropriate data colour space in Table 38.

When used the byte assignment and structure shall be as given in Table 70.

Table 70 — responseCurveSet16Type structure

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	'rcs2' (72637332h) [response curve set with 2-byte precision] type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 9	2	Number of channels (n)	uInt16Number
10 to 11	2	Count of measurement types (m)	uInt16Number
12 to (11 + 4m)	4m	m offsets, each relative to byte 0 of this structure, with one entry for each measurement type. Each offset shall point to the start of the response curve structure for the measurement type.	uInt32Number [m]
(12 + 4 m) to end	Variable	m response curve structures, with one structure for each measurement type	See Table 71

Each response curve structure has the format shown in Table 71.

Table 71 — Curve structure

Byte position	Field length bytes	Content	Encoded as
0 to 3	4	Measurement unit signature	see Table 72
4 to (3 + 4 n)	4n	n counts of measurements in response arrays, one for each channel i. Each count of measurements, q_i, shall be the count of response16Numbers in the response array for channel i p shall be the sum of the counts q_i for all channels. The counts shall be ordered in the channel order specified in Table 38 for the appropriate data colour space.	uInt32Number [n]
(4 + 4 n) to (3 + 4 n + 12 n)	12n	n PCSXYZ values, one for each channel. Each entry shall be a measurement of a patch with the maximum colorant value for the channel. The PCSXYZ values shall be relative colorimetric. The PCSXYZ values shall be ordered in the channel order specified in Table 38 for the appropriate data colour space.	XYZNumber [n]
(4 + 16 n) to (3 + 16 n+8 p)	8 p	n response arrays, one for each channel (i). Each response array shall contain q_i response16Numbers. Each response16Number shall contain a measurement for channel i. The response arrays shall be ordered in the channel order specified in Table 38 for the appropriate data colour space.	response16Number [p]
n is the number of channels as defined in Table 70. i indicates a channel in the range 1 to n.			

The response arrays shall be ordered with normalized device code elements increasing.

The measurement unit shall be encoded as shown in Table 72.

Table 72 — Curve measurement encodings

Measurement unit	Description	Signature	Hex encoding
Status A	ISO 5-3 densitometer response. This is the accepted standard for reflection densitometers for measuring photographic colour prints.	'StaA'	53746141h
Status E	ISO 5-3 densitometer response which is the accepted standard in Europe for colour reflection densitometers.	'StaE'	53746145h
Status I	ISO 5-3 densitometer response commonly referred to as narrow band or interference-type response.	'StaI'	53746149h
Status T	ISO 5-3 wide band colour reflection densitometer response which is the accepted standard in the United States for colour reflection densitometers.	'StaT'	53746154h
Status M	ISO 5-3 densitometer response for measuring colour negatives.	'StaM'	5374614Dh
DIN E	DIN 16536-2 densitometer response, with no polarizing filter.	'DN'	434E2020h
DIN E	DIN 16536-2 densitometer response, with polarizing filter.	'DN P'	434E2050h
DIN I	DIN 16536-2 narrow band densitometer response, with no polarizing filter.	'DNN '	434E4E20h
DIN I	DIN 16536-2 narrow band densitometer response, with polarizing filter.	'DNNP'	434E4E50h

10.20 s15Fixed16ArrayType

This type represents an array of generic 4-byte (32-bit) fixed point quantity. The number of values is determined from the size of the tag.

When used the byte assignment and encoding shall be as given in Table 73.

Table 73 — s15Fixed16ArrayType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘sf32’ (73663332h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to end	Variable	An array of s15Fixed16Number values

10.21 signatureType

The signatureType contains a 4-byte sequence, such as those defined in Table 22. Sequences of less than four characters are padded at the end with spaces, 20h. Typically this type is used for registered tags that can be displayed on many development systems as a sequence of four characters.

When used the byte assignment and encoding shall be as given in Table 74.

Table 74 — signatureType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘sig ’ (73696720h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to 11	4	4-byte signature

10.22 textType

The textType is a simple text structure that contains a 7-bit ASCII text string. The length of the string is obtained by subtracting 8 from the element size portion of the tag itself. This string shall be terminated with a 00h byte.

When used the byte assignment and encoding shall be as given in Table 75.

Table 75 — textType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘text’ (74657874h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to end	Variable	A string of (element size 8) 7-bit ASCII characters

10.23 u16Fixed16ArrayType

This type represents an array of generic 4-byte (32-bit) quantity. The number of values is determined from the size of the tag.

When used the byte assignment and encoding shall be as given in Table 76.

Table 76 — u16Fixed16ArrayType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘uf32’ (75663332h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to end	Variable	An array of u16Fixed16Number values

10.24 uInt16ArrayType

This type represents an array of generic 2-byte (16-bit) quantity. The number of values is determined from the size of the tag.

When used the byte assignment and encoding shall be as given in Table 77.

Table 77 — uInt16ArrayType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘ui16’ (75693136h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to end	Variable	An array of unsigned 16bit integers

10.25 uInt32ArrayType

This type represents an array of generic 4–byte (32-bit) quantity. The number of values is determined from the size of the tag.

When used the byte assignment and encoding shall be as given in Table 78.

Table 78 — uInt32ArrayType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘ui32’ (75693332h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to end	Variable	An array of unsigned 32-bit integers

10.26 uInt64ArrayType

This type represents an array of generic 8–byte (64-bit) quantity. The number of values is determined from the size of the tag.

When used the byte assignment and encoding shall be as given in Table 79.

Table 79 — uInt64ArrayType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘ui64’ (75693634h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to end	Variable	An array of unsigned 64-bit integers

10.27 uInt8ArrayType

This type represents an array of generic 1–byte (8-bit) quantity. The number of values is determined from the size of the tag.

When used the byte assignment and encoding shall be as given in Table 80.

Table 80 — uInt8ArrayType encoding

Byte position	**Field length** bytes	**Content**
0 to 3	4	‘ui08’ (75693038h) type signature
4 to 7	4	Reserved, shall be set to 0
8 to end	Variable	An array of unsigned 8-bit integers

10.28 viewingConditionsType

This type represents a set of viewing condition parameters. When used the byte assignment and encoding shall be as given in Table 81.

Table 81 — viewingConditionsType encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘view’ (76696577h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to 19	12	Un-normalized CIEXYZ values for illuminant (in which Y is in cd/m^2)	XYZNumber
20 to 31	12	Un-normalized CIEXYZ values for surround (in which Y is in cd/m^2)	XYZNumber
32 to 35	4	Illuminant type	As described in measurementType

The viewing condition described in this tag is the actual viewing condition assumed for the media for which the profile is defined, specified in un-normalized CIEXYZ values. Note that the luminanceTag shall be the same as the Y value given in this tag.

10.29 XYZType

The XYZType contains an array of three encoded values for PCSXYZ, CIEXYZ, or nCIEXYZ values. The number of sets of values is determined from the size of the tag. When used the byte assignment and encoding shall be as given in Table 82. Tristimulus values shall be non-negative. The signed encoding allows for implementation optimizations by minimizing the number of fixed formats.

Table 82 — XYZType encoding

Byte position	**Field length** bytes	**Content**	**Encoded as**
0 to 3	4	‘XYZ ’ (58595A20h) type signature	
4 to 7	4	Reserved, shall be set to 0	
8 to end	Variable	An array of PCSXYZ, CIEXYZ, or nCIEXYZ values	XYZNumber

Annex A
(informative)

Data colour encodings and rendering intents

A.1 General

The colour profile format defined in this part of ISO 15076 supports a wide variety of colour encodings divided into three basic families which are nCIEXYZ based, RGB based and CMYK based. An achromatic (grey) colour space is also specified. The basic spaces, together with spaces which may be derived from them, are given in Table A.1.

Table A.1 — Colour encoding types supported

Base space	Description	Derivative spaces
nCIEXYZ	Base CIE device-independent colour space	CIELAB, PCSXYZ, PCSLAB, CIELUV, CIEYxy
GRAY	Monochrome device-dependent colour space	
RGB	Base device-dependent colour space	HLS, HSV, YCbCr
CMY	Base device-dependent colour space	
CMYK	Base device-dependent colour space	

The CIE colour spaces are defined in CIE 15. Derivatives of the nCIEXYZ space are defined as connection spaces (PCSXYZ and PCSLAB) in order to provide the unambiguous colour specification required (see Annex D for further information). The device dependent spaces above are only representative and other device dependent colour spaces may be used without needing to update the profile format specification or the software that uses it. Such spaces are specified in this part of ISO 15076 as xCLR (where x is 2 to 15, see Table 19).

A.2 Colour measurement parameters

The default measurement parameters for the PCS, and all other colour spaces defined in this part of ISO 15076, are based on ISO 13655.

NOTE ISO 13655 requires that reflectance measurements be made using a 0°:45° or 45°:0° measurement geometry and that tristimulus values be calculated for a standard illuminant of D50 using the 1931 CIE standard colorimetric observer.

The only deviation from that specification is that all tristimulus values are divided by 100 so that $Y = 1$ for the perfect diffuser (and for the media white point following calculation of media-relative values). It should be noted that ISO 13655 currently makes provision for either a black or white backing for making measurements for reflecting media. Many users of colour management systems prefer to use a white backing for this purpose.

One of the first steps in profile building involves determining the colorimetry of a set of colours from some imaging test object or reproduction medium. The colorimetric values need to be corrected for flare, if the measurement conditions are such that they produce a level of flare different from that normally associated with high quality reflection measurements. Furthermore, if the illumination on the test object or reproduction medium differs from the reference illuminant (D50), it is necessary to apply a chromatic adaptation transform to the measured values. For the media-relative colorimetric intent, scaling to the media white point is then performed to produce values appropriate for the PCS. For the perceptual intent, other factors such as the viewing conditions, differences in gamut between the actual and reference media, and user preferences also need to be considered by the profile builder.

A PCS illuminant field is provided in the profile header, as defined in 7.2.16. The values contained in this field are the nCIEXYZ values of the PCS adopted white, which are equal to the nCIEXYZ values of CIE Illuminant D50 [X=0,964 2, Y=1,000 0, Z=0,824 9]. (Note that the PCS illuminant field should not be confused with the viewing conditions tag defined in viewingCondDescTag and viewingConditionsTag, see 9.2.48 and 9.2.49.)

As described in 6.3, the PCS is based on media-relative colorimetry. This is in comparison to ICC-absolute colorimetry. In ICC-absolute colorimetry colours are represented in nCIEXYZ with respect to the PCS adopted white. In media-relative colorimetry, colours are represented in PCSXYZ with respect to the media white, e.g. unprinted paper for a printer profile.

The actual media and viewing conditions used in practice will typically differ from the reference conditions. The profile specification defines tags which provide information about the actual white point of a given media or display and the viewing environment. These tags may be used by a CMM to provide functionality beyond that of the default. For example, an advanced CMM could use the tags to provide the option of no or partial chromatic adaptation, or to calculate colour appearance correlates for improved dynamic gamut mapping. Tag information can also be useful in choosing a profile appropriate for a specific use.

A.3 PCS encodings

There are many ways of encoding CIE colorimetry. The nCIEXYZ space represents a linear transformation of the average colour matching data, obtained by mixing red, green and blue lights to match all spectral colours, derived experimentally in the 1920s. The CIELAB space represents a transformation of the nCIEXYZ space into one that is more perceptually uniform. This uniformity allows colour errors to be approximately equally weighted throughout its domain.

This part of ISO 15076 provides two methods in order to satisfy conflicting requirements for accuracy and storage space. These encodings, a PCSLAB encoding and a 16-bit per component PCSXYZ encoding, are described in 6.3.4. While supporting multiple encodings increases the complexity of colour management, it provides immense flexibility in addressing different user requirements such as colour accuracy and memory footprint.

The relationship between PCSXYZ and PCSLAB is given by the set of equations defined in ISO 13655, but where the media white point (rather than the illuminant) is used as the relevant white point. Thus

$$\frac{X}{X_\mathrm{n}} \text{ is replaced by } \frac{X_\mathrm{r}}{X_\mathrm{i}} \text{ (or } \frac{X_\mathrm{a}}{X_\mathrm{mw}}\text{)} \tag{A.1}$$

$$\frac{Y}{Y_\mathrm{n}} \text{ is replaced by } \frac{Y_\mathrm{r}}{Y_\mathrm{i}} \text{ (or } \frac{Y_\mathrm{a}}{Y_\mathrm{mw}}\text{)} \tag{A.2}$$

$$\frac{Z}{Z_\mathrm{n}} \text{ is replaced by } \frac{Z_\mathrm{r}}{Z_\mathrm{i}} \text{ (or } \frac{Z_\mathrm{a}}{Z_\mathrm{mw}}\text{)} \tag{A.3}$$

where $X_\mathrm{a}Y_\mathrm{a}Z_\mathrm{a}$, $X_\mathrm{i}Y_\mathrm{i}Z_\mathrm{i}$, $X_\mathrm{mw}Y_\mathrm{mw}Z_\mathrm{mw}$ and $X_\mathrm{r}Y_\mathrm{r}Z_\mathrm{r}$ are as defined in 6.3.2.

The equations are as follows:

$$L^* = 116\left[f\left(\frac{Y}{Y_\mathrm{n}}\right)\right] - 16 \tag{A.4}$$

$$a^* = 500\left[f\left(\frac{X}{X_\text{n}}\right) - f\left(\frac{Y}{Y_\text{n}}\right)\right] \tag{A.5}$$

$$b^* = 200\left[f\left(\frac{Y}{Y_\text{n}}\right) - f\left(\frac{Z}{Z_\text{n}}\right)\right] \tag{A.6}$$

$$\text{for} \quad \frac{X}{X_\text{n}} > \left(\frac{6}{29}\right)^3, \; f\left(\frac{X}{X_\text{n}}\right) = \left(\frac{X}{X_\text{n}}\right)^{\frac{1}{3}} \tag{A.7}$$

$$\text{for} \quad \frac{Y}{Y_\text{n}} > \left(\frac{6}{29}\right)^3, \; f\left(\frac{Y}{Y_\text{n}}\right) = \left(\frac{Y}{Y_\text{n}}\right)^{\frac{1}{3}} \tag{A.8}$$

$$\text{for} \quad \frac{Z}{Z_\text{n}} > \left(\frac{6}{29}\right)^3, \; f\left(\frac{Z}{Z_\text{n}}\right) = \left(\frac{Z}{Z_\text{n}}\right)^{\frac{1}{3}} \tag{A.9}$$

$$\text{for} \quad \frac{X}{X_\text{n}} \leqslant \left(\frac{6}{29}\right)^3, \; f\left(\frac{X}{X_\text{n}}\right) = \left(\frac{841}{108}\right) \times \left(\frac{X}{X_\text{n}}\right) + \frac{4}{29} \tag{A.10}$$

$$\text{for} \quad \frac{Y}{Y_\text{n}} \leqslant \left(\frac{6}{29}\right)^3, \; f\left(\frac{Y}{Y_\text{n}}\right) = \left(\frac{841}{108}\right) \times \left(\frac{Y}{Y_\text{n}}\right) + \frac{4}{29} \tag{A.11}$$

$$\text{for} \quad \frac{Z}{Z_\text{n}} \leqslant \left(\frac{6}{29}\right)^3, \; f\left(\frac{Z}{Z_\text{n}}\right) = \left(\frac{841}{108}\right) \times \left(\frac{Z}{Z_\text{n}}\right) + \frac{4}{29} \tag{A.12}$$

where X, X_n, Y, Y_n, Z, Z_n are as defined in ISO 13655.

The PCS encodings do not represent a quantization of the connection space. The purpose of the encodings is to allow points within the space to be specified. Since the processing models benefit from interpolation between table entries, the interpolated AToB results should be used as the inputs to the BToA transforms. The AToB results should not be rounded to the nearest encoding value.

A.4 External and internal conversions

CMMs or other applications that use ICC tags to perform colour transformations typically need to perform two types of data processing in addition to table interpolation. First, because the colour values being processed (such as image pixels) may not match the native precision of an ICC tag (such as a lut16Type or lut8Type), it may be necessary to alter the precision of the input to (or results from) these transforms. Second, because there is more than one PCS encoding, it may be necessary to convert the output from a first transform before applying it to the input of a second transform. These two types of additional processing may be thought of as primarily affecting the **external** and **internal** interfaces of ICC processing, respectively.

In the first (external) case, the appropriate conversion method is to multiply each colour value by $(2^M–1)/(2^N–1)$, where N is the starting number of bits and M is the required number of bits. This converts a number with values from 0 to $(2^N–1)$ to a number with values from 0 to $(2^M–1)$. For example, to prepare an 8-bit image value for input to a lut16Type tag the scale factor is $(2^{16}–1)/(2^8–1) = (65\,535{,}0)/(255{,}0) = 257{,}0$. Note that the colours represented by the scaled numbers (be they device coordinates or values in some other colour space) are not intentionally altered by the change in precision. For example, if a particular image value represents a PCSLAB L^* of 31,0, then the scaled value is also intended to represent a PCSLAB L^* of 31,0.

However, a reduction in precision may force a small error. Additionally, if an integer value is required from the scaling operation, it should be obtained via rounding rather than truncation.

In the second (internal) case, the appropriate conversion uses the equations specified in A.3 to convert between PCSXYZ and PCSLAB.

Annex B
(informative)

Embedding profiles

B.1 General

This annex gives guidelines for embedding device profiles within EPS, TIFF, JFIF, and GIF image files. All profiles except Abstract and DeviceLink profiles can be embedded. The complete profile should be embedded with all tags intact and unchanged.

NOTE Other file formats, such as ISO 15444-2 [4] and proprietary file formats such as PSD, specify the embedding of ICC profiles. The embedding guidelines given in this annex are for file formats that do not specifically define how they are to be embedded. File formats that support embedding of ICC profiles are given on www.color.org.

B.2 Embedding ICC profiles in EPS files

The two places within EPS files that embedding ICC profiles are appropriate are when associated with a screen preview and when associated with the page description. Embedding ICC profiles within a screen preview is necessary so that applications using this screen preview to display a representation of the EPS page description can do so with accurate colours. Embedding ICC profiles within a page description is necessary so that sophisticated applications, such as OPI server software, can perform colour conversions along with image replacement. For general information concerning PostScript's Document Structuring Conventions (DSC), the EPS file format, or specific PostScript operators, see the PostScript Language Reference Manual[11].

There are a variety of different methods of storing a screen preview within an EPS file depending on the intended environment. For cross platform applications with embedded ICC profiles, TIFF screen previews are recommended. The TIFF format has been extended to support the embedding of ICC profiles. ICC profiles can also be embedded in a platform specific manner.

A given page description may use multiple distinct colour spaces. In such cases, colour conversions should be performed to a single colour space to associate with the screen preview.

ICC profiles can also be embedded in the page description portion of an EPS file using the %%BeginICCProfile: / %%EndICCProfile comments. This convention is defined as follows.

```
%%BeginICCProfile: <profileid> <numberof> [<type> [<bytesorlines>]]
<profileid> ::= <text> (Profile ID)
<numberof> ::= <int> (Lines or physical bytes)
<type> ::= Hex | ASCII (Type of data)
<bytesorlines> ::= Bytes | Lines (Read in bytes or lines)
%%EndICCProfile (no keywords)
```

These comments are designed to provide information about embedded ICC profiles. If the type argument is missing, ASCII data is assumed. ASCII refers to an ASCII base-85 representation of the data. If the bytesorlines argument is missing, <numberof> should be considered to indicate bytes of data. If <numberof> = –1, the number of bytes of data are unknown. In this case, to skip over the profile it is necessary to read data until the encountering the %%EndICCProfile comment.

<profileID> provides the profile's ID in order to synchronize it with PostScript's setcolorspace and findcolorrendering operators and associated operands (see below). Note that <numberof> indicates the bytes of physical data, which vary from the bytes of virtual data in some cases. With hex, each byte of virtual

data is represented by two ASCII characters (two bytes of physical data). Although the PostScript interpreter ignores white space and percent signs in hex and ASCII data, these count toward the byte count.

Each line of profile data should begin with a single percent sign (%) followed by a space. This makes the entire profile section a PostScript language comment so the file can be sent directly to a printer without modification. The space avoids confusion with the open extension mechanism associated with DSC comments.

ICC profiles can be embedded within EPS files to allow sophisticated applications, such as OPI server software, to extract the profiles, and to perform colour processing based on these profiles. In such situations it is desirable to locate the page description's colour space and rendering intent, since this colour space and rendering intent may need to be modified based on any colour processing. The %%BeginSetColorSpace: / %%EndSetColorSpace and %%BeginRenderingIntent: / %%EndRenderingIntent comments are used to delimit the colour space and rendering intent respectively.

```
%%BeginSetColorSpace: <profileid>
<profileid> ::= <text> (ICC Profile ID)
%%EndSetColorSpace (no keywords)
```

<profileid> provides the ICC profile's ID corresponding to this colour space. The ICC profile with this profile ID should have occurred in the PostScript job using the %%BeginICCProfile: / %%EndICCProfile comment convention prior to this particular %%BeginSetColorSpace: comment.

NOTE 1 An example of usage is shown here for CIE 1931 XYZ with D65 white point that refers to the ICC profile with <profileid> = XYZProfile.

```
%%BeginSetColorSpace: XYZProfile
[/CIEBasedABC <<
/WhitePoint [0.9505 1 1.0890]
/RangeABC [0 0.9505 0 1 0 1.0890]
/RangeLMN [0 0.9505 0 1 0 1.0890]
>>] setcolorspace
%%EndSetColorSpace
```

The setcolorspace command is included within the comments. The PostScript enclosed in these comments should not perform any other operations other than setting the colour space and should have no side effects.

```
%%BeginRenderingIntent: <profileid>
<profileid> ::= <text> (ICC Profile ID)
%%EndRenderingIntent (no keywords)
```

<profileid> provides the ICC profile's ID corresponding to this rendering intent. The ICC profile with this profile ID sholuld have occurred in the PostScript job using the %%BeginICCProfile: / %%EndICCProfile comment convention prior to invocation of this particular %%BeginRenderingIntent: comment.

NOTE 2 An example of usage is shown here for the Perceptual rendering intent that refers to the ICC profile with <profileid> = RGBProfile.

```
%%BeginRenderingIntent: RGBProfile
/Perceptual findcolorrendering pop
/ColorRendering findresource setcolorrendering
%%EndRenderingIntent
```

The setcolorrendering command is included within the comments. The PostScript enclosed in these comments should not perform any other operations other than setting the rendering intent and should have no side effects.

Annex C describes the method to be used to identify ICC profiles used to generate PostScript CSAs and CRDs.

B.3 Embedding ICC profiles in TIFF files

The discussion below assumes some familiarity with TIFF internal structure. It is beyond the scope of this part of ISO 15076 to detail the TIFF format, and readers are referred to the "TIFF™ Revision 6.0" specification, which is available from Adobe Systems Incorporated.

The International Color Consortium has been assigned a private TIFF tag for purposes of embedding ICC device profiles within TIFF image files. This is not a required TIFF tag, and Baseline TIFF readers are not currently required to read it. It is, however, strongly recommended that this tag be honoured.

An ICC device profile is embedded, in its entirety, as a single TIFF field or Image File Directory (IFD) entry in the IFD containing the corresponding image data. An IFD should contain no more than one embedded profile. A TIFF file may contain more than one image, and so, more than one IFD. Each IFD may have its own embedded profile. Note, however, that Baseline TIFF readers are not required to read any IFDs beyond the first one.

The structure of the ICC Profile IFD Entry is given in Table B.1.

Table B.1 — ICC profile IFD entry structure

Byte offset	Field length bytes	Content
0 to 1	2	The TIFFTag that identifies the field = 34675(8773h)
2 to 3	2	The field Type is 7 = UNDEFINED (treated as 8-bit bytes).
4 to 7	4	The Count of values = the size of the embedded ICC profile in bytes.
8 to 11	4	The Value Offset = the file offset, in bytes, to the beginning of the ICC profile.

Like all IFD entry values, the embedded profile should begin on a 2-byte boundary, so the Value Offset will always be an even number.

A TIFF reader should have no knowledge of the internal structure of an embedded ICC profile and should extract the profile intact.

B.4 Embedding ICC profiles in JPEG files

The JPEG standard (ISO/IEC 10918-1[2]) supports application specific data segments. These segments may be used for tagging images with ICC profiles. The APP2 marker is used to introduce the ICC profile tag. Given that there are only 15 supported APP markers, there is a chance of many applications using the same marker. ICC tags are thus identified by beginning the data with a special null terminated byte sequence, "ICC_PROFILE".

The length field of a JPEG marker is only two bytes long; the length of the length field is included in the total. Hence, the values 0 and 1 are not legal lengths. This would limit the maximum data length to 65 533. The identification sequence would lower this even further. As it is quite possible for an ICC profile to be longer than this, a mechanism is required to break the profile into chunks and place each chunk in a separate marker. A mechanism to identify each chunk in sequence order is therefore necessary.

The identifier sequence is followed by one byte indicating the sequence number of the chunk (counting starts at 1) and one byte indicating the total number of chunks. All chunks in the sequence should indicate the same total number of chunks. The 1-byte chunk count limits the size of embeddable profiles to 16 707 345 bytes.

B.5 Embedding ICC profiles in GIF files

The GIF89a image file format supports Application Extension blocks, which are used for “application specific” information. These blocks may be used for tagging images with ICC profiles.

The Application Identifier for an embedded profile should be the following 8 bytes: “ICCRGBG1”. The Authentication Code should be “012”. The entire profile should be embedded as application data, using the conventional technique of breaking the data into chunks of at most 255 bytes of data.

Annex C
(informative)

Relationship between ICC profiles and PostScript CSAs and CRDs

C.1 General

When ICC profiles are used to generate PostScript “color space arrays” (CSAs) or “color rendering dictionaries” (CRDs) it is useful to be able to identify the profile used to define the CSA or CRD. This can be achieved by adding the following keys to the CSA or CRD. This mechanism does not rely on comments, and enables a parser to obtain the original profile from outside the PostScript file.

C.2 Profile identification keys for a PostScript CSA

The following keys are recommended by Adobe Systems for inclusion in PostScript (and EPS) colorspace arrays:

a) **/CreationDate (string)**: Identifies the date and time at which the colorspace array was created or most recently modified. The value of this entry should be coordinated with the calibrationDateTimeTag attribute of any associated ICC profile, and its syntax should conform to this part of ISO 15076 and ASN.1 as defined in ISO/IEC 8824[1].

b) **/RenderingIntent (name or string)**: Identifies the rendering intent that this colorspace array is designed to achieve. The options are: AbsoluteColorimetric, RelativeColorimetric, Saturation or Perceptual.

c) **/Description (string)**: 7-bit ASCII description string from the ICC profile ‘desc’ tag.

d) **/Copyright (string)**: 7-bit ASCII copyright string from the ICC profile ‘cprt’ tag.

NOTE In profiles conforming to this part of ISO 15076 (ICC v4.0), the copyright and description strings are multi-lingual. Only the U.S. English string from the ICC Profile is present in the CSA/CRD. If the ICC Profile does not contain a U.S. English string, one can be computed from the first multilingual string.

e) **/ColorSpace (string)**: Data colour space field of the profile data from the ICC profile header. This should be the 4-character ASCII string representing the colour space signature (see 7.2.6).

f) **/ProfileID (hexadecimal string)**: This is the Profile ID of the ICC Profile. This should be encoded as hexadecimal data, enclosed in < and >. For profiles conforming to ICC.1:2004-10, Profile ID is generally present in the profile header. For those ICC profiles not containing a Profile ID, a Profile ID should be computed using the method described in 7.2.18.

EXAMPLE Colorspace Array (CSA) from Photoshop:

```
[ /CIEBasedABC
<<
/CreationDate (19990603000000)
/RenderingIntent (Perceptual)
/Description (not Adobe RGB (1998))
/ColorSpace (RGB )
/Copyright (Copyright 1999 Adobe Systems Incorporated)
/ProfileID <33BC7F1C156FA0D72F8F717AE5886BD4>
/DecodeLMN [{2.1992 exp}bind {2.1992 exp}bind {2.1992 exp}bind]
                              /MatrixLMN [0.3805 0.7083 0.9959
0.1282 0.0593 0.7144
0.4554 0.2324 0.0145]
/WhitePoint [0.9642 1.0000 0.8249]
>>                            ]
```

C.3 Profile identification keys for a PostScript CRD

C.3.1 The following keys are recommended for inclusion in PostScript CRDs by Adobe Systems Inc. in the PostScript Language Reference Manual.

a) **/CreationDate (string)**: Identifies the date and time at which the CRD was created or most recently modified. The value of this entry should be coordinated with the calibrationDateTimeTag attribute of any associated ICC profile, and its syntax should conform ASN.1, defined in ISO/IEC 8824-1[1].

b) **/RenderingIntent (name or string)**: Identifies the rendering intent that this CRD is designed to achieve. The options are: AbsoluteColorimetric, RelativeColorimetric, Saturation or Perceptual.

C.3.2 The use of the following additional keys is also recommended in cases where it is important to establish a clear relationship between the CRD and the ICC profile from which it was derived.

a) **/Description (string)**: 7-bit ASCII description string from the ICC profile ‘desc’ tag.

b) **/Copyright (string)**: 7-bit ASCII copyright string from the ICC profile ‘cprt’ tag.

NOTE In profiles conforming to this part of ISO 15076 (ICC v4.0), the copyright and description strings are multi-lingual. Only the U.S. English string from the ICC Profile is present in the CSA/CRD. If the ICC Profile does not contain a U.S. English string, one can be computed from the first multi-lingual string.

c) **/ColorSpace (string)**: Data colour space field of the profile data from the ICC profile header. This should be the 4-character ASCII string representing the colour space signature (see 7.2.6).

d) **/ProfileID (hexadecimal string)**: ASCII string representation of the hex-encoded Profile ID of the ICC Profile. For profiles conforming to this part of ISO 15076 (ICC v4.0), Profile ID is generally present in the profile header. For those ICC profiles not containing a Profile ID, a Profile ID should be computed using the method described in 7.2.18.