INTERNATIONAL STANDARD

ISO/IEC 15444-1

Second edition
2004-09-15

Information technology — JPEG 2000 image coding system: Core coding system

Technologies de l'information — Système de codage d'image JPEG 2000: Système de codage de noyau

Reference number
ISO/IEC 15444-1:2004(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.

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

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

Page

Page

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents, as indicated in Table L.1.

This part of ISO/IEC 15444 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information* in collaboration with ITU-T. The identical text is published as ITU-T Rec. T.800.

This second edition cancels and replaces the first edition (ISO/IEC 15444-1:2000), of which it constitutes a minor revision. It also incorporates the Amendment ISO/IEC 15444-1:2000/Amd.1:2002 and the Technical Corrigenda ISO/IEC 15444-1:2000/Cor.1:2002 and ISO/IEC 15444-1:2000/Cor.2:2002.

ISO/IEC 15444 consists of the following parts, under the general title *Information technology — JPEG 2000 image coding system*:

— *Part 1: Core coding system*

— *Part 2: Extensions*

— *Part 3: Motion JPEG 2000*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

— *Part 6: Compound image file format*

— *Part 9: Interactivity tools, APIs and protocols*

The following part is under preparation:

— *Part 8: Secure JPEG 2000*

Annex I

JP2 file format syntax

(This annex forms an integral part of this Recommendation | International Standard.
This annex is optional for the minimum decoder.)

In this annex, the flow charts and tables are normative only in the sense that they are defining an output that alternative implementations shall duplicate.

I.1 File format scope

This annex defines an optional file format that applications may choose to use to wrap JPEG 2000 compressed image data. While not all applications will use this format, many applications will find that this format meets their needs. However, those applications that do implement this file format shall implement it as described in this entire annex.

This annex:

- specifies a binary container for both image and metadata;
- specifies a mechanism to indicate image properties, such as the tonescale or colourspace of the image;
- specifies a mechanism by which readers may recognize the existence of intellectual property rights information in the file;
- specifies a mechanism by which metadata (including vendor-specific information) can be included in files specified by this Recommendation | International Standard.

I.2 Introduction to the JP2 file format

The JPEG 2000 file format (JP2 file format) provides a foundation for storing application specific data (metadata) in association with a JPEG 2000 codestream, such as information which is required to display the image. As many applications require a similar set of information to be associated with the compressed image data, it is useful to define the format of that set of data along with the definition of the compression technology and codestream syntax.

Conceptually, the JP2 file format encapsulates the JPEG 2000 codestream along with other core pieces of information about that codestream. The building-block of the JP2 file format is called a box. All information contained within the JP2 file is encapsulated in boxes. This Recommendation | International Standard defines several types of boxes; the definition of each specific box type defines the kinds of information that may be found within a box of that type. Some boxes will be defined to contain other boxes.

I.2.1 File identification

JP2 files can be identified using several mechanisms. When stored in traditional computer file systems, JP2 files should be given the file extension ".jp2" (readers should allow mixed case for the alphabetic characters). On Macintosh file systems, JP2 files should be given the type code 'jp2\040'.

I.2.2 File organization

A JP2 file represents a collection of boxes. Some of those boxes are independent, and some of those boxes contain other boxes. The binary structure of a file is a contiguous sequence of boxes. The start of the first box shall be the first byte of the file, and the last byte of the last box shall be the last byte of the file.

The binary structure of a box is defined in I.4.

Logically, the structure of a JP2 file is as shown in Figure I.1. Boxes with dashed borders are optional in conforming JP2 files. However, an optional box may define mandatory boxes within that optional box. In that case, if the optional box exists, those mandatory boxes within the optional box shall exist. If the optional box does not exist, then the mandatory boxes within those boxes shall also not exist.

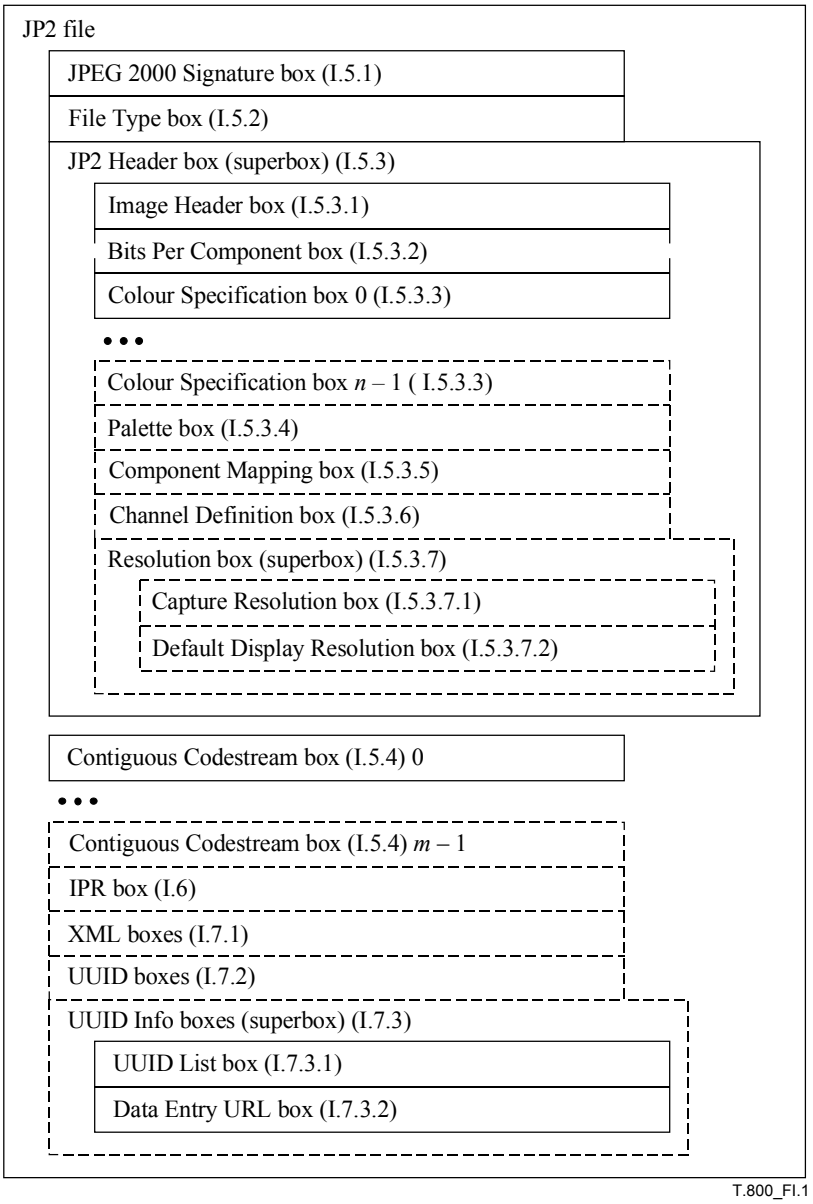

Figure I.1 – Conceptual structure of a JP2 file

Figure I.1 specifies only the containment relationship between the boxes in the file. A particular order of those boxes in the file is not generally implied. However, the JPEG 2000 Signature box shall be the first box in a JP2 file, the File Type box shall immediately follow the JPEG 2000 Signature box and the JP2 Header box shall fall before the Contiguous Codestream box.

The file shown in Figure I.1 is a strict sequence of boxes. Other boxes may be found between the boxes defined in this Recommendation | International Standard. However, all information contained within a JP2 file shall be in the box format; byte-streams not in the box format shall not be found in the file.

As shown in Figure I.1, a JP2 file contains a JPEG 2000 Signature box, JP2 Header box, and one or more Contiguous Codestream boxes. A JP2 file may also contain other boxes as determined by the file writer. For example, a JP2 file may contain several XML boxes (containing metadata) between the JP2 Header box and the first Contiguous Codestream box.

I.2.3 Greyscale, colour, palette, multi-component specification

The JP2 file format provides two methods to specify the colourspace of the image. The enumerated method specifies the colourspace of an image by specifying a numeric value that specifies the colourspace. In this Recommendation | International Standard, images in the sRGB colourspace and greyscale images can be defined using the enumerated method.

The JP2 file format also provides for the specification of the colourspace of an image by embedding a restricted form of an ICC profile in the file. That profile shall be of either the Monochrome or Three-Component Matrix-Based class of input profiles as defined by the ICC Profile Format Specification, ICC.1:1998-09. This allows for the specification of a wide range of greyscale and RGB class colourspaces, as well as a few other spaces that can be represented by those two profile classes. See J.9 for a more detailed description of the legal colourspace transforms, how those transforms are stored in the file, and how to process an image using that transform without using an ICC colour management engine. While restricted, these ICC profiles are fully compliant ICC profiles and the image can thus be processed through any ICC compliant engine that supports profiles as defined in ICC.1:1998-09.

In addition to specifying the colourspace of the image, this Recommendation | International Standard provides a means by which a single component palettized image can be decoded and converted back to multiple-component form by the translation from index space to multiple-component space. Any such depalettization is applied before the colourspace is interpreted. In the case of palettized images, the specification of the colourspace of the image is applied to the multiple-component values stored in the palette.

I.2.4 Inclusion of opacity channels

The JP2 file format provides a means to indicate the presence of auxiliary channels (such as opacity), to define the type of those channels, and to specify the ordering and source of those channels (whether they are directly extracted from the codestream or generated by applying a palette to a codestream component). When a reader opens the JP2 file, it will determine the ordering and type of each component. The application must then match the component definition and ordering from the JP2 file with the component ordering as defined by the colourspace specification. Once the file components have been mapped to the colour channels, the decompressed image can be processed through any needed colourspace transformations.

In many applications, components other than the colour channels are required. For example, many images used on web pages contain opacity information; the browser uses this information to blend the image into the background. It is thus desirable to include both the colour and auxiliary channels within a single codestream.

How applications deal with opacity or other auxiliary channels is outside the scope of this Recommendation | International Standard.

I.2.5 Metadata

One important aspect of the JP2 file format is the ability to add metadata to a JP2 file. Because all information is encapsulated in boxes, and all boxes have types, the format provides a simple mechanism for a reader to extract relevant information, while ignoring any box that contains information that is not understood by that particular reader. In this way, new boxes can be created, either through this or other Recommendations | International Standards or private implementation. Also, any new box added to a JP2 file shall not change the visual appearance of the image.

I.2.6 Conformance with the file format

All conforming files shall contain all boxes required by this Recommendation | International Standard, and those boxes shall be as defined in this Recommendation | International Standard. Also, all conforming readers shall correctly interpret all required boxes defined in this Recommendation | International Standard and thus shall correctly interpret all conforming files.

I.3 Greyscale/Colour/Palettized/multi-component specification architecture

One of the most important aspects of a file format is that it specifies the colourspace of the contained image data. In order to properly display or interpret the image data, it is essential that the colourspace of that image is properly characterized. The JP2 file format provides a multi-level mechanism for characterizing the colourspace of an image.

I.3.1 Enumerated method

The simplest method for characterizing the colourspace of an image is to specify an integer code representing the colourspace in which the image is encoded. This method handles the specification of sRGB, greyscale, and sYCC images. Extensions to this method can be used to specify other colourspaces, including the definition of multi-component images.

For example, the image file may indicate that a particular image is encoded in the sRGB colourspace. To properly interpret and display the image, an application must natively understand the definition of the sRGB colourspace. Because an application must natively understand each specified colourspace, the complexity of this method is dependent on the exact colourspaces specified. Also, complexity of this mechanism is proportional to the number of colourspaces that are specified and required for conformance. While this method provides a high level of interoperability for images encoded using colourspaces for which correct interpretation is required for conformance, this method is very inflexible. This Recommendation | International Standard defines a specific set of colourspaces for which interpretation is required for conformance.

I.3.2 Restricted ICC profile method

An application may also specify the colourspace of an image using two restricted types of ICC profiles. This method handles the specification of the most commonly used RGB and greyscale class colourspaces through a low-complexity method.

An ICC profile is a standard representation of the transformation required to convert one colourspace into another colourspace. With respect to the JP2 file format, an ICC profile defines how decompressed samples from the codestream are converted into a standard colorspace (the Profile Connection Space (PCS)). Depending on the original colourspace of the samples, this transformation may be either very simple or very complex.

The ICC Profile Format Specification defines two specific classes of ICC profiles that are simple to implement, referred to within the profile specification as Monochrome Input and Three-Component Matrix-Based Input Profiles. These profiles limit the transformation from the source colourspace to the PCS_{XYZ} to the application of a non-linearity curve and a 3×3 matrix. It is practical to expect all applications, including simple devices, to be able to process the image through this transformation. Thus all conforming applications are required to correctly interpret the colourspace of any image that specifies the colourspace using this subset of possible ICC profile types.

For the JP2 file format, profiles shall conform to the ICC profile definition as defined by the ICC Profile Format Specification, ICC.1:1998-09, as well as the restrictions specified above. See J.9 for a more detailed description of the legal colourspace transforms, how those transforms are stored in the file, and how to process an image using that transform without using an ICC colour management engine.

I.3.3 Using multiple methods

Architecturally, the format allows for multiple methods to be embedded in a file and allows other standards to define additional enumerated methods and to define extended methods. This provides readers conforming to those extensions a choice as to what image processing path should be used to interpret the colourspace of the image. However, the first method found in the file (in the first Colourspace Specification box in the JP2 Header box) shall be one of the methods as defined and restricted in this Recommendation | International Standard. A conforming reader shall use that first method and ignore all other methods (in additional Colourspace Specification boxes) found in the file.

I.3.4 Palettized images

In addition to specifying the interpretation of the image in terms of colourspace, this Recommendation | International Standard allows for the decoding of a single component where the value of that single component represents an index into a palette of colours. Input of a decompressed sample to the palette converts the single value to a multiple-component tuple. The value of that tuple represents the colour of that sample; that tuple shall then be interpreted according to the other colour specification methods (Enumerated or Restricted ICC) as if that multiple-component sample had been directly extracted from multiple components in the codestream.

I.3.5 Interactions with the decorrelating multiple component transform

The specification of colour within the JP2 file format is independent of the use of a multiple component transformation within the codestream (the CSsiz parameter of the SIZ marker segment as specified in A.5.1 and in Annex G). The colourspace transformations specified through the sequence of Colour Specification boxes shall be applied to the image samples after the reverse multiple component transformation has been applied to the decompressed samples. While the application of these decorrelating component transformations is separate, the application of an encoder-based multiple component transformation will often improve the compression of colour image data.

I.3.6 Key to graphical descriptions (informative)

Each box is described in terms of its function, usage, and length. The function describes the information contained in the box. The usage describes the logical location and frequency of this box in the file. The length describes which parameters determine the length of the box.

These descriptions are followed by a figure that shows the order and relationship of the parameters in the box. Figure I.2 shows an example of this type of figure. A rectangle is used to indicate the parameters in the box. The width of the rectangle is proportional to the number of bytes in the parameter. A shaded rectangle (diagonal stripes) indicates that the parameter is of varying size. Two parameters with superscripts and a grey area between indicate a run of several of these parameters. A sequence of two groups of multiple parameters with superscripts separated by a grey area indicates a run of that group of parameters (one set of each parameter in the group, followed by the next set of each parameter in the group). Optional parameters or boxes will be shown with a dashed rectangle.

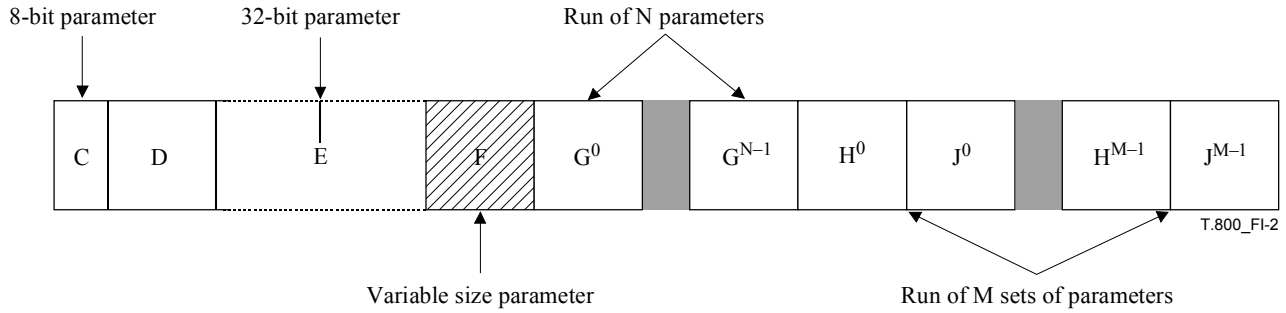

Figure I.2 – Example of the box description figures

The figure is followed by a list that describes the meaning of each parameter in the box. If parameters are repeate, the d length and nature of the run of parameters is defined. As an example, in Figure I.2, parameters C, D, E and F are 8-, 16-, 32-bit and variable length respectively. The notation G^0 and G^{N-1} implies that there are N different parameters, G^i, in a row. The group of parameters H^0 and H^{M-1}, and J^0 and J^{M-1} specify that the box will contain H^0, followed by J^0, followed by H^1 and J^1, continuing to H^{M-1} and J^{M-1} (M instances of each parameter in total). Also, the field E is optional and may not be found in this box.

After the list is a table that either describes the allowed parameter values or provides references to other tables that describe these values.

In addition, in a figure describing the contents of a superbox, an ellipsis (…) will be used to indicate that contents of the file between two boxes is not specifically defined. Any box (or sequence of boxes), unless otherwise specified by the definition of that box, may be found in place of the ellipsis.

For example, the superbox shown in Figure I.3 must contain an AA box and a BB box, and the BB box must follow the AA box. However, there may be other boxes found between boxes AA and BB. Dealing with unknown boxes is discussed in I.8.

Figure I.3 – Example of the superbox description figures

I.4 Box definition

Physically, each object in the file is encapsulated within a binary structure called a box. That binary structure is as in Figure I.4:

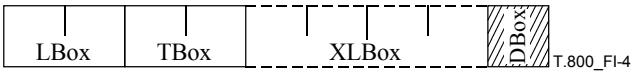

Figure I.4 – Organization of a box

LBox: Box Length. This field specifies the length of the box, stored as a 4-byte big endian unsigned integer. This value includes all of the fields of the box, including the length and type. If the value of this field is 1, then the XLBox field shall exist and the value of that field shall be the actual length of the box. If the value of this field is 0, then the length of the box was not known when the LBox field was written. In this case, this box contains all bytes up to the end of the file. If a box of length 0 is contained within another box (its superbox), then the length of that superbox shall also be 0. This means that this box is the last box in the file. The values 2-7 are reserved for ISO use.

TBox: Box Type. This field specifies the type of information found in the DBox field. The value of this field is encoded as a 4-byte big endian unsigned integer. However, boxes are generally referred to by an ISO/IEC 646 character string translation of the integer value. For all box types defined within this Recommendation | International Standard, box types will be indicated as both character string (normative) and as 4-byte hexadecimal integers (informative). Also, a space character is shown in the character string translation of the box type as "\040". All values of TBox not defined within this Recommendation | International Standard are reserved for ISO use.

XLBox: Box Extended Length. This field specifies the actual length of the box if the value of the LBox field is 1. This field is stored as an 8-byte big endian unsigned integer. The value includes all of the fields of the box, including the LBox, TBox and XLBox fields.

DBox: Box Contents. This field contains the actual information contained within this box. The format of the box contents depends on the box type and will be defined individually for each type.

Table I.1 – Binary structure of a box

Field name	Size (bits)	Value
LBox	32	0, 1, or 8 to $(2^{32}-1)$
TBox	32	Variable
XLBox	64 0	16 to $(2^{64}-1)$; if LBox = 1 Not applicable; if LBox ≠ 1
DBox	Variable	Variable

For example, consider the illustration in Figure I.5 of a sequence of boxes, including one box that contains other boxes:

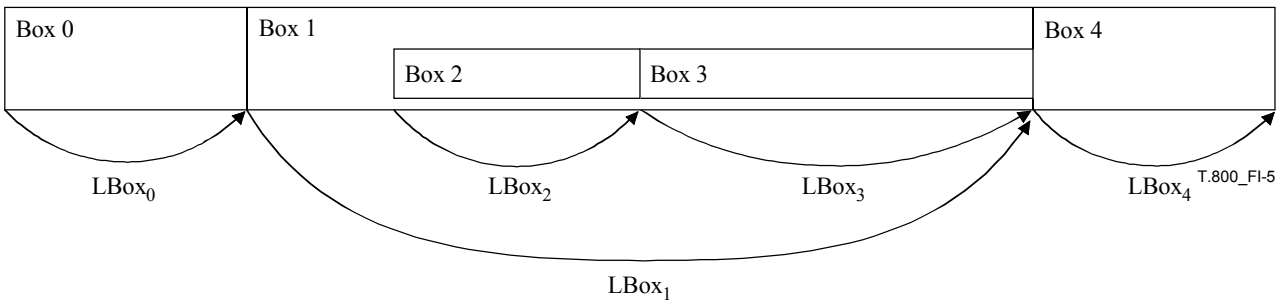

Figure I.5 – Illustration of box lengths

As shown in Figure I.5, the length of each box includes any boxes contained within that box. For example, the length of Box 1 includes the length of Boxes 2 and 3, in addition to the LBox and TBox fields for Box 1 itself. In this case, if the type of Box 1 was not understood by a reader, it would not recognize the existence of Boxes 2 and 3 because they would be completely skipped by jumping the length of Box 1 from the beginning of Box 1.

Table I.2 lists all boxes defined by this Recommendation | International Standard. Indentation within the table indicates the hierarchical containment structure of the boxes within a JP2 file.

Table I.2 – Defined boxes

Box name	Type	Superbox	Required?	Comments
JPEG 2000 Signature box	'jP\040\040' (0x6A50 2020)	No	Required	This box uniquely identifies the file as being part of the JPEG 2000 family of files.
File Type box	'ftyp' (0x6674 7970)	No	Required	This box specifies file type, version and compatibility information, including specifying if this file is a conforming JP2 file or if it can be read by a conforming JP2 reader.
JP2 Header box	'jp2h' (0x6A70 3268)	Yes	Required	This box contains a series of boxes that contain header-type information about the file.
Image Header box	'ihdr' (0x6968 6472)	No	Required	This box specifies the size of the image and other related fields.
Bits Per Component box	'bpcc' (0x6270 6363)	No	Optional	This box specifies the bit depth of the components in the file in cases where the bit depth is not constant across all components.
Colour Specification box	'colr' (0x636F 6C72)	No	Required	This box specifies the colourspace of the image.
Palette box	'pclr' (0x7063 6C72)	No	Optional	This box specifies the palette which maps a single component in index space to a multiple-component image.
Component Mapping box	'cmap' (0x636D 6170)	No	Optional	This box specifies the mapping between a palette and codestream components.
Channel Definition box	'cdef' (0x6364 6566)	No	Optional	This box specifies the type and ordering of the components within the codestream, as well as those created by the application of a palette.
Resolution box	'res\040' (0x7265 7320)	Yes	Optional	This box contains the grid resolution.
Capture Resolution box	'resc' (0x7265 7363)	No	Optional	This box specifies the grid resolution at which the image was captured.
Default Display Resolution box	'resd' (0x7265 7364)	No	Optional	This box specifies the default grid resolution at which the image should be dis played.
Contiguous Codestream box	'jp2c' (0x6A70 3263)	No	Required	This box contains the codestream as defined by Annex A.
Intellectual Property box	'jp2i' (0x6A70 3269)	No	Optional	This box contains intellectual property information about the image.
XML box	'xml\040' (0x786D 6C20)	No	Optional	This box provides a tool by which vendors can add XML formatted information to a JP2 file.
UUID box	'uuid' (0x7575 6964)	No	Optional	This box provides a tool by which vendors can add additional information to a file without risking conflict with other vendors.
UUID Info box	'uinf' (0x7569 6E66)	Yes	Optional	This box provides a tool by which a vendor may provide access to additional information associated with a UUID.
UUID List box	'ulst' (0x7563 7374)	No	Optional	This box specifies a list of UUIDs.
URL box	'url\040' (0x7572 6C20)	No	Optional	This box specifies a URL.

I.5 Defined boxes

The following boxes shall properly be interpreted by all conforming readers. Each of these boxes conforms to the standard box structure as defined in I.4. The following clauses define the value of the DBox field from Table I.1 (the contents of the box). It is assumed that the LBox, TBox and XLBox fields exist for each box in the file as defined in Annex I.4.

I.5.1 JPEG 2000 Signature box

The JPEG 2000 Signature box identifies that the format of this file was defined by the JPEG 2000 Recommendation | International Standard, as well as provides a small amount of information which can help determine the validity of the rest of the file. The JPEG 2000 Signature box shall be the first box in the file, and all files shall contain one and only one JPEG 2000 Signature box.

The type of the JPEG 2000 Signature box shall be 'jP\040\040' (0x6A50 2020). The length of this box shall be 12 bytes. The contents of this box shall be the 4-byte character string '<CR><LF><0x87><LF>' (0x0D0A 870A). For file verification purposes, this box can be considered a fixed-length 12-byte string which shall have the value: 0x0000 000C 6A50 2020 0D0A 870A.

The combination of the particular type and contents for this box enable an application to detect a common set of file transmission errors. The CR-LF sequence in the contents catches bad file transfers that alter newline sequences. The final linefeed checks for the inverse of the CR-LF translation problem. The third character of the box contents has its high-bit set to catch bad file transfers that clear bit 7.

I.5.2 File Type box

The File Type box specifies the Recommendation | International Standard which completely defines all of the contents of this file, as well as a separate list of readers, defined by other Recommendations | International Standards, with which this file is compatible, and thus the file can be properly interpreted within the scope of that other standard. This box shall immediately follow the JPEG 2000 Signature box. This differentiates between the standard which completely describes the file, from other standards that interpret a subset of the file.

All files shall contain one and only one File Type box.

The type of the File Type Box shall be 'ftyp' (0x6674 7970). The contents of this box shall be as in Figure I.6:

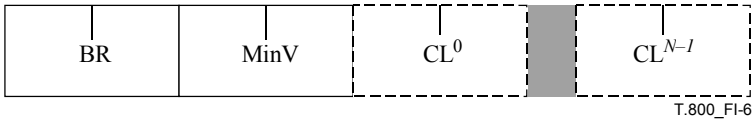

Figure I.6 – Organization of the contents of a File Type box

BR: Brand. This field specifies the Recommendation | International Standard which completely defines this file. This field is specified by a four-byte string of ISO/IEC 646 characters. The value of this field is defined in Table I.3:

Table I.3 – Legal Brand values

Value	Meaning
'jp2\040'	IS 15444-1, Annex I (This Recommendation | International Standard)
other values	Reserved for other ISO uses

In addition, the Brand field shall be considered functionally equivalent to a major version number. A major version change (if there ever is one), representing an incompatible change in the JP2 file format, shall define a different value for the Brand field.

If the value of the Brand field is not 'jp2\040', then a value of 'jp2\040' in the Compatibility list indicates that a JP2 reader can interpret the file in some manner as intended by the creator of the file.

MinV: Minor version. This parameter defines the minor version number of this JP2 specification for which the file complies. The parameter is defined as a 4-byte big endian unsigned integer. The value of this field shall be zero. However, readers shall continue to parse and interpret this file even if the value of this field is not zero.

CL^i: Compatibility list. This field specifies a code representing this Recommendation | International Standard, another standard, or a profile of another standard, to which the file conforms. This field is encoded as a four-byte string of ISO/IEC 646 characters. A file that conforms to this Recommendation | International Standard shall have at least one CL^i field in the File Type box, and shall contain the value 'jp2\040' in one of the CL^i fields in the File Type box, and all conforming readers shall properly interpret all files with 'jp2\040' in one of the CL^i fields

If one of the CL^i fields contains the value "J2P0" then the first codestream contained within this JP2 file is restricted as described for Profile-0 from Table A.45.

If one of the CL^i fields contains the value "J2P1" then the first codestream contained within this JP2 file is restricted as described for Profile-1 from Table A.45.

Other values of the Compatibility list field are reserved for ISO use.

The number of CL^i fields is determined by the length of this box.

Table I.4 – Format of the contents of the File Type box

Field name	Size (bits)	Value
BR	32	0 to $(2^{32}-1)$
MinV	32	0
CL^i	32	0 to $(2^{32}-1)$

I.5.3 JP2 Header box (superbox)

The JP2 Header box contains generic information about the file, such as number of components, colourspace, and grid resolution. This box is a superbox. Within a JP2 file, there shall be one and only one JP2 Header box. The JP2 Header box may be located anywhere within the file after the File Type box but before the Contiguous Codestream box. It also must be at the same level as the JPEG 2000 Signature and File Type boxes (it shall not be inside any other superbox within the file).

The type of the JP2 Header box shall be 'jp2h' (0x6A70 3268).

This box contains several boxes. Other boxes may be defined in other standards and may be ignored by conforming readers. Those boxes contained within the JP2 Header box that are defined within this Recommendation | International Standard are as in Figure I.7:

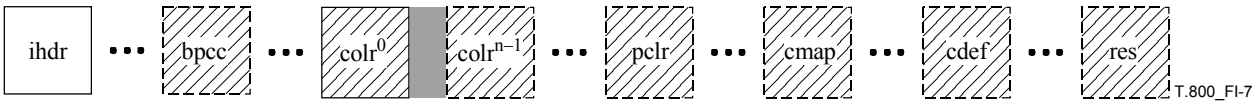

Figure I.7 – Organization of the contents of a JP2 Header box

ihdr: Image Header box. This box specifies information about the image, such as its height and width. Its structure is specified in I.5.3.1. This box shall be the first box in the JP2 Header box.

bpcc: Bits Per Component box. This box specifies the bit depth of each component in the codestream after decompression. Its structure is specified in I.5.3.2. This box may be found anywhere in the JP2 Header box provided that it comes after the Image Header box.

$colr^i$: Colour Specification boxes. These boxes specify the colourspace of the decompressed image. Their structures are specified in I.5.3.3. There shall be at least one Colour Specification box within the JP2 Header box. The use of multiple Colour Specification boxes provides the ability for a decoder to be given multiple optimization or compatibility options for colour processing. These boxes may be found anywhere in the JP2 Header box provided that they come after the Image Header box. All Colour Specification boxes shall be contiguous within the JP2 Header box.

pclr: Palette box. This box defines the palette to use to create multiple components from a single component. Its structure is specified in I.5.3.4. This box may be found anywhere in the JP2 Header box provided that it comes after the Image Header box.

cmap: Component Mapping box. This box defines how image channels are identified from the actual components in the codestream. Its structure is specified in I.5.3.5. This box may be found anywhere in the JP2 Header box provided that it comes after the Image Header box.

cdef: Channel Definition box. This box defines the channels in the image. Its structure is specified in I.5.3.6. This box may be found anywhere in the JP2 Header box provided that it comes after the Image Header box.

res: Resolution box. This box specifies the capture and default display grid resolutions of the image. Its structure is specified in I.5.3.7. This box may be found anywhere in the JP2 Header box provided that it comes after the Image Header box.

I.5.3.1 Image Header box

This box contains fixed length generic information about the image, such as the image size and number of components. The contents of the JP2 Header box shall start with an Image Header box. Instances of this box in other places in the file shall be ignored. The length of the Image Header box shall be 22 bytes, including the box length and type fields. Much of the information within the Image Header box is redundant with information stored in the codestream itself.

All references to "the codestream" in the descriptions of fields in this Image Header box apply to the codestream found in the first Contiguous Codestream box in the file. Files that contain contradictory information between the Image Header box and the first codestream are not conforming files. However, readers may choose to attempt to read these files by using the values found within the codestream.

The type of the Image Header box shall be 'ihdr' (0x6968 6472) and contents of the box shall have the format as in Figure I.8:

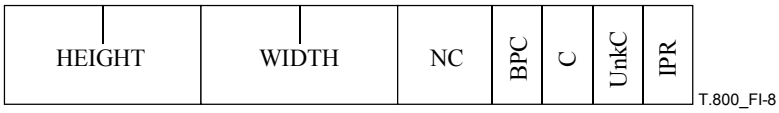

Figure I.8 – Organization of the contents of an Image Header box

HEIGHT: Image area height. The value of this parameter indicates the height of the image area. This field is stored as a 4-byte big endian unsigned integer. The value of this field shall be Ysiz – YOsiz, where Ysiz and YOsiz are the values of the respective fields in the SIZ marker in the codestream. See Figure B.1 for an illustration of the image area. However, reference grid points are not necessarily square; the aspect ratio of a reference grid point is specified by the Resolution box. If the Resolution box is not present, then a reader shall assume that reference grid points are square.

WIDTH: Image area width. The value of this parameter indicates the width of the image area. This field is stored as a 4-byte big endian unsigned integer. The value of this field shall be Xsiz – XOsiz, where Xsiz and XOsiz are the values of the respective fields in the SIZ marker in the codestream. See Figure B.1 for an illustration of the image area. However, reference grid points are not necessarily square; the aspect ratio of a reference grid point is specified by the Resolution box. If the Resolution box is not present, then a reader shall assume that reference grid points are square.

NC: Number of components. This parameter specifies the number of components in the codestream and is stored as a 2-byte big endian unsigned integer. The value of this field shall be equal to the value of the Csiz field in the SIZ marker in the codestream.

BPC: Bits per component. This parameter specifies the bit depth of the components in the codestream, minus 1, and is stored as a 1-byte field.

If the bit depth and the sign are the same for all components, then this parameter specifies that bit depth and shall be equivalent to the values of the $Ssiz^i$ fields in the SIZ marker in the codestream (which shall all be equal). If the components vary in bit depth and/or sign, then the value of this field shall be 255 and the JP2 Header box shall also contain a Bits Per Component box defining the bit depth of each component (as defined in I.5.3.2).

The low 7-bits of the value indicate the bit depth of the components. The high-bit indicates whether the components are signed or unsigned. If the high-bit is 1, then the components contain signed values. If the high-bit is 0, then the components contain unsigned values.

C: Compression type. This parameter specifies the compression algorithm used to compress the image data. The value of this field shall be 7. It is encoded as a 1-byte unsigned integer. Other values are reserved for ISO use.

UnkC: Colourspace Unknown. This field specifies if the actual colourspace of the image data in the codestream is known. This field is encoded as a 1-byte unsigned integer. Legal values for this field are 0, if the colourspace of the image is known and correctly specified in the Colourspace Specification boxes within the file, or 1, if the colourspace of the image is not known. A value of 1 will be used in cases such as the transcoding of legacy images where the actual colourspace of the image data is not known. In those cases, while the colourspace interpretation methods specified in the file may not accurately reproduce the image with respect to some original, the image should be treated as if the methods do accurately reproduce the image. Values other than 0 and 1 are reserved for ISO use.

IPR: Intellectual Property. This parameter indicates whether this JP2 file contains intellectual property rights information. If the value of this field is 0, this file does not contain rights information, and thus the file does not contain an IPR box. If the value is 1, then the file does contain rights information and thus does contain an IPR box as defined in I.6. Other values are reserved for ISO use.

Table I.5 – Format of the contents of the Image Header box

Field name	Size (bits)	Value
HEIGHT	32	1 to $(2^{32}-1)$
WIDTH	32	1 to $(2^{32}-1)$
NC	16	1 to 16 384
BPC	8	See Table I.6
C	8	7
Unk	8	0 to 1
IPR	8	0 to 1

Table I.6 – BPC values

Values (bits) MSB LSB	Component sample precision
x000 0000 to x010 0101	Component bit depth = value + 1. From 1 bit deep through 38 bits deep respectively (counting the sign bit, if appropriate)
0xxx xxxx	Components are unsigned values
1xxx xxxx	Components are signed values
1111 1111	Components vary in bit depth
	All other values reserved for ISO use

I.5.3.2 Bits Per Component box

The Bits Per Component box specifies the bit depth of each component. If the bit depth of all components in the codestream is the same (in both sign and precision), then this box shall not be found. Otherwise, this box specifies the bit depth of each individual component. The order of bit depth values in this box is the actual order in which those components are enumerated within the codestream. The exact location of this box within the JP2 Header box may vary provided that it follows the Image Header box.

There shall be one and only one Bits Per Component box inside a JP2 Header box.

The type of the Bits Per Component Box shall be 'bpcc' (0x6270 6363). The contents of this box shall be as in Figure I.9:

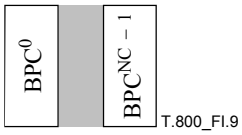

Figure I.9 – Organization of the contents of a Bits Per Component box

BPC^i: Bits per component. This parameter specifies the bit depth of component *i*, minus 1, encoded as a 1-byte value. The ordering of the components within the Bits Per Component Box shall be the same as the ordering of the components within the codestream. The number of BPC^i fields shall be the same as the value of the NC field from the Image Header box. The value of this field shall be equivalent to the respective Ssiz^i field in the SIZ marker in the codestream.

The low 7-bits of the value indicate the bit depth of this component. The high-bit indicates whether the component is signed or unsigned. If the high-bit is 1, then the component contains signed values. If the high-bit is 0, then the component contains unsigned values.

Table I.7 – Format of the contents of the Bits Per Component box

Field name	Size (bits)	Value
BPC^i	8	See Table I.8

Table I.8 – BPC^i values

Values (bits) MSB LSB	Component sample precision
x000 0000 to x010 0101	Component bit depth = value + 1. From 1 bit deep through 38 bits deep respectively (counting the sign bit, if appropriate)
0xxx xxxx	Components are unsigned values
1xxx xxxx	Components are signed values
	All other values reserved for ISO use

I.5.3.3 Colour Specification box

Each Colour Specification box defines one method by which an application can interpret the colourspace of the decompressed image data. This colour specification is to be applied to the image data after it has been decompressed and after any reverse decorrelating component transform has been applied to the image data.

A JP2 file may contain multiple Colour Specification boxes, but must contain at least one, specifying different methods for achieving "equivalent" results. A conforming JP2 reader shall ignore all Colour Specification boxes after the first. However, readers conforming to other standards may use those boxes as defined in those other standards.

The type of a Colour Specification box shall be 'colr' (0x636F 6C72). The contents of a Colour Specification box is as in Figure I.10:

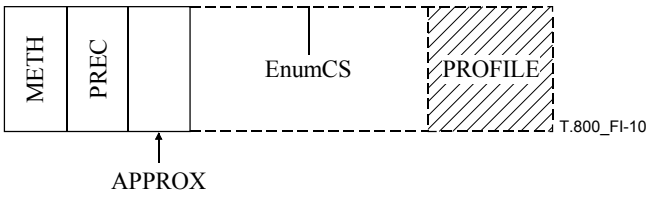

Figure I.10 – Organization of the contents of a Colour Specification box

METH: Specification method. This field specifies the method used by this Colour Specification box to define the colourspace of the decompressed image. This field is encoded as a 1-byte unsigned integer. The value of this field shall be 1 or 2, as defined in Table I.9.

Table I.9 – Legal METH values

Value	Meaning
1	**Enumerated Colourspace**. This colourspace specification box contains the enumerated value of the colourspace of this image. The enumerated value is found in the EnumCS field in this box. If the value of the METH field is 1, then the EnumCS shall exist in this box immediately following the APPROX field, and the EnumCS field shall be the last field in this box
2	**Restricted ICC profile**.This Colour Specification box contains an ICC profile in the PROFILE field. This profile shall specify the transformation needed to convert the decompressed image data into the PCS_{XYZ}, and shall conform to either the Monochrome Input or Three-Component Matrix-Based Input profile class, and contain all the required tags specified therein, as defined in ICC.1:1998-09. As such, the value of the Profile Connection Space field in the profile header in the embedded profile shall be 'XYZ\040' (0x5859 5A20) indicating that the output colourspace of the profile is in the XYZ colourspace. Any private tags in the ICC profile shall not change the visual appearance of an image processed using this ICC profile. The components from the codestream may have a range greater than the input range of the tone reproduction curve (TRC) of the ICC profile. Any decoded values should be clipped to the limits of the TRC before processing the image through the ICC profile. For example, negative sample values of signed components may be clipped to zero before processing the image data through the profile. See J.9 for a more detailed description of the legal colourspace transforms, how those transforms are stored in the file, and how to process an image using that transform without using an ICC colour management engine. If the value of METH is 2, then the PROFILE field shall immediately follow the APPROX field and the PROFILE field shall be the last field in the box.
other values	Reserved for other ISO use. If the value of METH is not 1 or 2, there may be fields in this box following the APPROX field, and a conforming JP2 reader shall ignore the entire Colour Specification box.

PREC: Precedence. This field is reserved for ISO use and the value shall be set to zero; however, conforming readers shall ignore the value of this field. This field is specified as a signed 1-byte integer.

APPROX: Colourspace approximation. This field specifies the extent to which this colour specification method approximates the "correct" definition of the colourspace. The value of this field shall be set to zero; however, conforming readers shall ignore the value of this field. Other values are reserved for other ISO use. This field is specified as 1-byte unsigned integer.

EnumCS: Enumerated colourspace. This field specifies the colourspace of the image using integer codes. To correctly interpret the colour of an image using an enumerated colourspace, the application must know the definition of that colourspace internally. This field contains a 4-byte big endian unsigned integer value indicating the colourspace of the image. If the value of the METH field is 2, then the EnumCS field shall not exist. Valid EnumCS values for the first colourspace specification box in conforming files are limited to 16, 17, and 18 as defined in Table I.10:

Table I.10 – Legal EnumCS values

Value	Meaning
16	sRGB as defined by IEC 61966-2-1
17	greyscale: A greyscale space where image luminance is related to code values using the sRGB non-linearity given in Equations (2) through (4) of IEC 61966-2-1 (sRGB) specification: $Y' = Y_{8\,bit} / 255$ (I-1) $\text{for } (Y' \le 0{,}04045),\ Y_{lin} = Y' / 12{,}92$ (I-2) $\text{for } (Y' > 0{,}04045),\ Y_{lin} = \left(\frac{Y' + 0{,}055}{1{,}055}\right)^{2{,}4}$ where Y_{lin} is the linear image luminance value in the range 0.0 to 1.0. The image luminance values should be interpreted relative to the reference conditions in Section 2 of IEC 61966-2-1.
18	sYCC as defined by IEC 61966-2-1 Amd. 1 NOTE – It is not recommend to use ICT or RCT specified in Annex G with sYCC image data. See J.15 for guidelines on handling YCC codestreams.
other values	Reserved for other ISO uses

PROFILE: ICC profile. This field contains a valid ICC profile, as specified by the ICC Profile Format Specification, which specifies the transformation of the decompressed image data into the PCS. This field shall not exist if the value of the METH field is 1. If the value of the METH field is 2, then the ICC profile shall conform to the Monochrome Input Profile class or the Three-Component Matrix-Based Input Profile class as defined in ICC.1:1998-09.

Table I.11 – Format of the contents of the Colour Specification box

Field name	Size (bits)	Value
METH	8	1 to 2
PREC	8	0
APPROX	8	0
EnumCS	32 if METH = 1 0 if METH = 2	0 to $(2^{32}-1)$ no value
PROFILE	Variable	Variable; see the ICC Profile Format Specification, version ICC.1:1998-09.

I.5.3.4 Palette box

This box specifies a palette that can be used to create channels from components. However, the Palette box does not specify the creation of any particular channel; the creation of channels based on the application of the palette to a component is specified by the Component Mapping box. The colourspace or meaning of the generated channel is specified by the Channel Definition box (or specified through the defaults defined in the specification of the Channel Definition box if the Channel Definition box does not exist). If the JP2 Header box contains a Palette box, then it shall also contain a Component Mapping box. If the JP2 Header box does not contain a Palette box, then it shall not contain a Component Mapping box.

There shall be at most one Palette box inside a JP2 Header box.

The type of the Palette box shall be 'pclr' (0x7063 6C72). The contents of this box shall be as in Figure I.11:

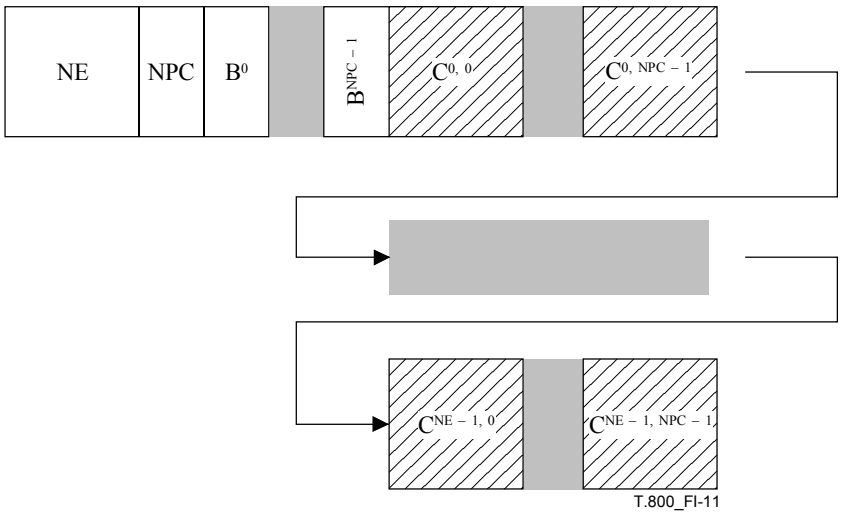

Figure I.11 – Organization of the contents of the Palette box

NE: Number of entries in the table. This value shall be in the range 1 to 1024 and is encoded as a 2-byte big endian unsigned integer.

NPC: Number of palette columns specified in the Palette box. For example, if the palette is to be used to map a single index component into a three-component RGB image, then the value of this field shall be 3. This field is encoded as a 1-byte unsigned integer.

B^i: This parameter specifies the bit depth of values created by palette column *i*, encoded as a 1-byte big endian integer. The low 7-bits of the value indicate the bit depth of this palette column. The high-bit indicates whether the palette column is signed or unsigned. If the high-bit is 1, then the palette column contains signed values. If the high-bit is 0, then the palette column contains unsigned values. The number of B^i values shall be the same as the value of the NPC field.

C^{ji}: The value for entry *j* for palette column *i*. C^{ji} values are organized in component major order; all of the values for entry *j* are grouped together, followed by all of the values for entry *j* + 1. In the example given above, this table would therefore read R_1, G_1 ,B_1 ,R_2 ,G_2 ,B_2, etc. The size of C^{ji} is the value specified by field B^i. The number of palette columns shall be the same as the NPC field. The number of C^{ji} values shall be the number of palette columns (the NPC field) times the number of entries in the palette (NE). If the value of B^i is not a multiple of 8, then each C^{ji} value is padded with zeros to a multiple of 8 bits and the actual value shall be stored in the low-order bits of the padded value. For example, if the value of B^i is 10 bits, then the individual C^{ji} values shall be stored in the low 10 bits of a 16-bit field.

Table I.12 – Format of the contents of the Palette box

Field name	Size (bits)	Value
NE	16	1 to 1024
NPC	8	1 to 255
B^i	8	See Table I.13
C^{ji}	Variable	Variable

Table I.13 – B^i values

Values (bits) MSB LSB	Palette column sample precision
x000 0000 to x010 0101	Palette column bit depth = value + 1. From 1 bit deep through 38 bits deep respectively (counting the sign bit, if appropriate)
0xxx xxxx	Palette column values are unsigned values
1xxx xxxx	Palette column values are signed values
	All other values reserved for ISO use.

I.5.3.5 Component Mapping box

The Component Mapping box defines how image channels are identified from the actual components decoded from the codestream. This abstraction allows a single structure (the Channel Definition box) to specify the colour or type of both palettized images and non-palettized images. This box contains an array of CMP^i, $MTYP^i$ and $PCOL^i$ fields. Each group of these fields represents the definition of one channel in the image. The channels are numbered in order starting with zero, and the number of channels specified in the Component Mapping box is determined by the length of the box.

There shall be at most one Component Mapping box inside a JP2 Header box.

If the JP2 Header box contains a Palette box, then the JP2 Header box shall also contain a Component Mapping box. If the JP2 Header box does not contain a Component Mapping box, the components shall be mapped directly to channels, such that component *i* is mapped to channel *i*.

The type of the Component Mapping box shall be ‘cmap’ (0x636D 6170). The contents of this box shall be as in Figure I.12:

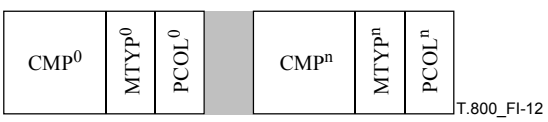

Figure I.12 – Organization of the contents of a Component Mapping box

CMP^i: This field specifies the index of component from the codestream that is mapped to this channel (either directly or through a palette). This field is encoded as a 2-byte big endian unsigned integer.

MTYP^i: This field specifies how this channel is generated from the actual components in the file. This field is encoded as a 1-byte unsigned integer. Legal values of the MTYP^i field are as in Table I.14:

Table I.14 – MTYP^i field values

Value	Meaning
0	Direct use. This channel is created directly from an actual component in the codestream. The index of the component mapped to this channel is specified in the CMP^i field for this channel.
1	Palette mapping. This channel is created by applying the palette to an actual component in the codestream. The index of the component mapped into the palette is specified in the CMP^i field for this channel. The column from the palette to use is specified in the PCOL^i field for this channel.
2 to 255	Reserved for ISO use

PCOL^i: This field specifies the index component from the palette that is used to map the actual component from the codestream. This field is encoded as a 1-byte unsigned integer. If the value of the MTYP^i field for this channel is 0, then the value of this field shall be 0.

Table I.15 – Format of the contents of the Component Mapping box

Field name	Size (bits)	Value
CMP^i	16	0 to 16 384
MTYP^i	8	0 to 1
PCOL^i	8	0 to 255

I.5.3.6 Channel Definition box

The Channel Definition box specifies the meaning of the samples in each channel in the image. The exact location of this box within the JP2 Header box may vary provided that it follows the Image Header box.The mapping between actual components from the codestream to channels is specified in the Component Mapping box. If the JP2 Header box does not contain a Component Mapping box, then a reader shall map component *i* to channel *i*, for all components in the codestream.

There shall be at most one Channel Definition box inside a JP2 Header box.

This box contains an array of channel descriptions. For each description, three values are specified: the index of the channel described by that association, the type of that channel, and the association of that channel with particular colours. This box may specify multiple descriptions for a single channel.

If a multiple component transform is specified within the codestream, the image must be in an RGB colourspace and the red, green and blue colours as channels 0, 1 and 2 in the codestream, respectively.

The type of the Channel Definition box shall be 'cdef' (0x6364 6566). The contents of this box shall be as in Figure I.13:

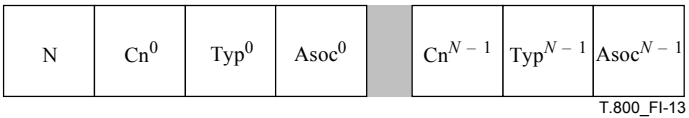

Figure I.13 – Organization of the contents of a Channel Definition box

N: Number of channel descriptions. This field specifies the number of channel descriptions in this box. This field is encoded as a 2-byte big endian unsigned integer.

Cn^i: Channel index. This field specifies the index of the channel for this description. The value of this field represents the index of the channel as defined within the Component Mapping box (or the actual component from the codestream if the file does not contain a Component Mapping box). This field is encoded as a 2-byte big endian unsigned integer.

Typ^i: Channel type. This field specifies the type of the channel for this description. The value of this field specifies the meaning of the decompressed samples in this channel. This field is encoded as a 2-byte big endian unsigned integer. Legal values of this field are shown in Table I.16:

Table I.16 – Typ^i field values

Value	Meaning
0	This channel is the colour image data for the associated colour.
1	Opacity. A sample value of 0 indicates that the sample is 100% transparent, and the maximum value of the channel (related to the bit depth of the codestream component or the related palette component mapped to this channel) indicates a 100% opaque sample. All opacity channels shall be mapped from unsigned components.
2	Premultiplied opacity. An opacity channel as specified above, except that the value of the opacity channel has been multiplied into the colour channels for which this channel is associated. Premultiplication is defined as follows: $$S_p = S \times \frac{\alpha}{\alpha_{max}} \quad \text{(I-3)}$$ where S is the original sample, S_p is the pre multiplied sample (the sample stored in the image, α is the value of the opacity channel, and α_{max} is the maximum value of the opacity channel as defined by the bit depth of the opacity channel.
3 to (2^{16}– 2)	Reserved for ISO use
2^{16}– 1	The type of this channel is not specified.

Asoc^i: Channel association. This field specifies the index of the colour for which this channel is directly associated (or a special value to indicate the whole image or the lack of an association). For example, if this channel is an opacity channel for the red channel in an RGB colourspace, this field would specify the index of the colour red. Table I.17 specifies legal association values. Table I.18 specifies legal colour indices. This field is encoded as a 2-byte big endian unsigned integer.

Table I.17 – Asoc^i field values

Value	Meaning
0	This channel is associated as the image as a whole (for example, an independent opacity channel that should be applied to all color channels).
1 to (2^{16}– 2)	This channel is associated with a particular colour as indicated by this value. This value is used to associate a particular channel with a particular aspect of the specification of the colourspace of this image. For example, indicating that a channel is associated with the red channel of an RGB image allows the reader to associate that decoded channel with the Red input to an ICC profile contained within a Colour Specification box. Colour indicators are specified in Table I.18.
2^{16}– 1	This channel is not associated with any particular colour.

Table I.18 – Colours indicated by the $Asoc^i$ field

Class of colourspace	Colour indicated by the following value of the $Asoc^i$ field			
	1	**2**	**3**	**4**
RGB	R	G	B	
Greyscale	Y			
YC_bC_r	Y	C_b	C_r	
The following colourspace classes are listed for future reference, as well as to aid in under standing of the use of the $Asoc^i$ field:				
XYZ	X	Y	Z	
Lab	L	a	b	
Luv	L	u	v	
YC_bC_r	Y	C_b	C_r	
Yxy	Y	x	y	
HSV	H	S	V	
HLS	H	L	S	
CMYK	C	M	Y	K
CMY	C	M	Y	
Jab	J	a	b	
n colour colourspaces	1	2	3	4

The values in Table I.18 specify indices that have been assigned to represent specific "colours" and do not refer to specific channels (or components within the codestream or palette). Readers must use the information contained within the Channel Definition box to determine which channels contain which colours.

In this box, channel indices are mapped from particular components within the codestream or palette. Colour indices specify how a particular channel shall be interpreted based on the specification of the colourspace of the image. There shall be one channel definition in this box for every colour required by the colourspace specification of this file as specified by the Colourspace Specification box.

For example, the green colour in an RGB image is specified by a {Cn, Typ, Asoc} value of {*i*, 0, 2}, where *i* is the index of that channel (either directly or as generated by applying the reverse multiple component transform to the actual components in the codestream). Applications that are only concerned with extracting the colour channels can treat the Typ/Asoc field pair as a four-byte value where the combined value maps directly to the colour indices (as the Typ field for a colour channel shall be 0).

In another example, the codestream may contain a channel *i* that specifies opacity blending samples for the red and green channels, and a channel *j* that specifies opacity blending samples for the blue channel. In that file, the following {Cn, Typ, Asoc} tuples would be found in the Channel Definition box for the two opacity channels: {*i*, 1, 1}, {*i*, 1, 2} and {*j*, 1, 3}.

There shall not be more than one channel in a JP2 file with a the same Typ^i and $Asoc^i$ value pair, with the exception of Typ^i and $Asoc^i$ values of $2^{16}– 1$ (not specified). For example a JP2 file in an RGB colourspace shall only contain one green channel, and a greyscale image shall contain only one grey channel. There shall be either exactly one opacity channel, exactly one pre-multiplied opacity channel, or neither associated with a single colour channel in an image.

If the codestream contains only colour channels and those channels are ordered in the same order as the associated colours (for example, an RGB image with three channels in the order R, G, then B), then this box shall not exist. If there are any auxiliary channels or the channels are not in the same order as the colour indices, then the Channel Definition box (see Table I.19) shall be found within the JP2 Header box with a complete list of channel definitions.

Table I.19 – Format of the Channel Definition box

Parameter	Size (bits)	Value
N	16	1 to (2^{16}– 1)
Cn^i	16	0 to (2^{16}– 1)
Typ^i	16	0 to (2^{16}– 1)
$Asoc^i$	16	0 to (2^{16}– 1)

I.5.3.7 Resolution box (superbox)

This box specifies the capture and default display grid resolutions of this image. If this box exists, it shall contain either a Capture Resolution box, or a Default Display Resolution box, or both.

There shall be at most one Resolution box inside a JP2 Header box.

The type of a Resolution box shall be 'res\040' (0x7265 7320). The contents of the Resolution box are as in Figure I.14:

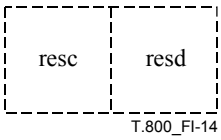

Figure I.14 – Organization of the contents of the Resolution box

resc: Capture Resolution box. This box specifies the grid resolution at which this image was captured. The format of this box is specified in I.5.3.7.1.

resd: Default Display Resolution box. This box specifies the default grid resolution at which this image should be displayed. The format of this box is specified in I.5.3.7.2

I.5.3.7.1 Capture Resolution box

This box specifies the grid resolution at which the source was digitized to create the image samples specified by the codestream. For example, this may specify the resolution of the flatbed scanner that captured a page from a book. The capture grid resolution could also specify the resolution of an aerial digital camera or satellite camera.

The vertical and horizontal capture grid resolutions are calculated using the six parameters (Table I.20) stored in this box in the following two equations, respectively:

$$VRc = \frac{VRcN}{VRcD} \times 10^{VRcE} \tag{I-4}$$

$$HRc = \frac{HRcN}{HRcD} \times 10^{HRcE} \tag{I-5}$$

The values *VRc* and *HRc* are always in reference grid points per meter. If an application requires the grid resolution in another unit, then that application must apply the appropriate conversion.

The type of a Capture Resolution box shall be 'resc' (0x7265 7363). The contents of the Capture Resolution box are as in Figure I.15:

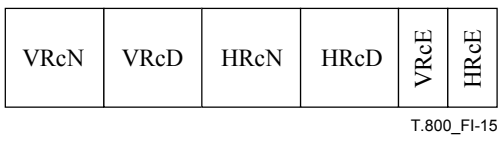

Figure I.15 – Organization of the contents of the Capture Resolution box

VRcN: Vertical Capture grid resolution numerator. This parameter specifies the *VRcN* value in Equation (I-4), which is used to calculate the vertical capture grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

VRcD: Vertical Capture grid resolution denominator. This parameter specifies the *VRcD* value in Equation (I-4), which is used to calculate the vertical capture grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

HRcN: Horizontal Capture grid resolution numerator. This parameter specifies the *HRcN* value in Equation (I-5), which is used to calculate the horizontal capture grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

HRcD: Horizontal Capture grid resolution denominator. This parameter specifies the *HRcD* value in Equation (I-5), which is used to calculate the horizontal capture grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

VRcE: Vertical Capture grid resolution exponent. This parameter specifies the *VRcE* value in Equation (I-4), which is used to calculate the vertical capture grid resolution. This parameter is encoded as a twos-complement 1-byte signed integer.

HRcE: Horizontal Capture grid resolution exponent. This parameter specifies the *HRcE* value in Equation (I-5), which is used to calculate the horizontal capture grid resolution. This parameter is encoded as a twos-complement 1-byte signed integer.

Table I.20 – Format of the contents of the Capture Resolution box

Field name	Size (bits)	Value
VRcN	16	1 to (2^{16}– 1)
VRcD	16	1 to (2^{16}– 1)
HRcN	16	1 to (2^{16}– 1)
HRcD	16	1 to (2^{16}– 1)
VRcE	8	–128 to 127
HRcE	8	–128 to 127

I.5.3.7.2 Default Display Resolution box

This box specifies a desired display grid resolution. For example, this may be used to determine the size of the image on a page when the image is placed in a page-layout program. However, this value is only a default. Each application must determine an appropriate display size for that application.

The vertical and horizontal display grid resolutions are calculated using the six parameters (Table I.21) stored in this box in the following two equations, respectively:

$$VRd = \frac{VRdN}{VRdD} \times 10^{VRdE} \tag{I-6}$$

$$HRd = \frac{HRdN}{HRdD} \times 10^{HRdE} \tag{I-7}$$

The values *VRd* and *HRd* are always in reference grid points per meter. If an application requires the grid resolution in another unit, then that application must apply the appropriate conversion.

The type of a Default Display Resolution box shall be 'resd' (0x7265 7364). The contents of the Default Display Resolution box are as in Figure I.16:

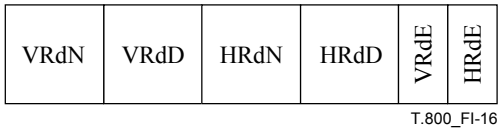

Figure I.16 – Organization of the contents of the Default Display Resolution box

VRdN: Vertical Display grid resolution numerator. This parameter specifies the *VRdN* value in Equation (I-6), which is used to calculate the vertical display grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

VRdD: Vertical Display grid resolution denominator. This parameter specifies the *VRdD* value in Equation (I-6), which is used to calculate the vertical display grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

HRdN: Horizontal Display grid resolution numerator. This parameter specifies the *HRdN* value in Equation (I-7), which is used to calculate the horizontal display grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

HRdD: Horizontal Display grid resolution denominator. This parameter specifies the *HRdD* value in Equation (I-7), which is used to calculate the horizontal display grid resolution. This parameter is encoded as a 2-byte big endian unsigned integer.

VRdE: Vertical Display grid resolution exponent. This parameter specifies the *VRdE* value in Equation (I-6), which is used to calculate the vertical display grid resolution. This parameter is encoded as a twos-complement 1-byte signed integer.

HRdE: Horizontal Display grid resolution exponent. This parameter specifies the *HRdE* value in Equation (I-7), which is used to calculate the horizontal display grid resolution. This parameter is encoded as a twos-complement 1-byte signed integer.

Table I.21 – Format of the contents of the Default Display Resolution box

Field name	Size (bits)	Value
VRdN	16	1 to $(2^{16} – 1)$
VRdD	16	1 to $(2^{16} – 1)$
HRdN	16	1 to $(2^{16} – 1)$
HRdD	16	1 to $(2^{16} – 1)$
VRdE	8	–128 to 127
HRdE	8	–128 to 127

I.5.4 Contiguous Codestream box

The Contiguous Codestream box contains a valid and complete JPEG 2000 codestream, as defined in Annex A. When displaying the image, a conforming reader shall ignore all codestreams after the first codestream found in the file. Contiguous Codestream boxes may be found anywhere in the file except before the JP2 Header box.

The type of a Contiguous Codestream box shall be 'jp2c' (0x6A70 3263). The contents of the box shall be as in Figure I.17:

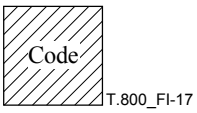

Figure I.17 – Organization of the contents of the Contiguous Codestream box

Code: This field contains a valid and complete JPEG 2000 codestream as specified by Annex A.

Table I.22 – Format of the contents of the Contiguous Codestream box

Field name	Size (bits)	Value
Code	Variable	Variable

I.6 Adding intellectual property rights information in JP2

This Recommendation | International Standard specifies a box type for a box which is devoted to carrying intellectual property rights information within a JP2 file. Inclusion of this information in a JP2 file is optional for conforming files. The definition of the format of the contents of this box is reserved for ISO. However, the type of this box is defined in this Recommendation | International Standard as a means to allow applications to recognize the existence of IPR information. Use and interpretation of this information is beyond the scope of this Recommendation | International Standard.

In general, an IPR box found at the top level of the file specifies IPR for the file as a whole. IPR boxes may be found at other locations, including inside superboxes defined by other Recommendations | International Standards. For those IPR boxes, the rights specified refer to the entity defined by the containing superbox.

The type of the Intellectual Property Box shall be 'jp2i' (0x6A70 3269).

I.7 Adding vendor-specific information to the JP2 file format

The following boxes provide a set of tools by which applications can add vendor-specific information to the JP2 file format. All of the following boxes are optional in conforming files and may be ignored by conforming readers.

I.7.1 XML boxes

An XML box contains vendor-specific information (in XML format) other than the information contained within boxes defined by this Recommendation | International Standard. There may be multiple XML boxes within the file, and those boxes may be found anywhere in the file except before the File Type box.

The type of an XML box is 'xml\040' (0x786D 6C20). The contents of the box shall be as in Figure I.18:

Figure I.18 – Organization of the contents of the XML box

DATA: This field shall contain a well-formed XML document as defined by REC-xml-19980210.

The existence of any XML boxes is optional for conforming files. Also, any XML box shall not contain any information necessary for decoding the image to the extent that is defined within this Recommendation | International Standard, and the correct interpretation of the contents of any XML box shall not change the visual appearance of the image. All readers may ignore any XML box in the file.

I.7.2 UUID boxes

A UUID box contains vendor-specific information other than the information contained within boxes defined within this Recommendation | International Standard. There may be multiple UUID boxes within the file, and those boxes may be found anywhere in the file except before the File Type box.

The type of a UUID box shall be 'uuid' (0x7575 6964). The contents of the box shall be as in Figure I.19:

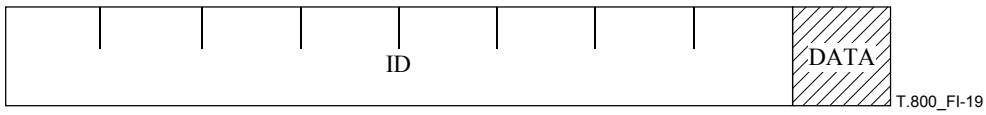

Figure I.19 – Organization of the contents of the UUID box

ID: This field contains a 16-byte UUID as specified by ISO/IEC 11578. The value of this UUID specifies the format of the vendor-specific information stored in the DATA field and the interpretation of that information.

DATA: This field contains the vendor-specific information. The format of this information is defined outside of the scope of this Recommendation | International Standard, but is indicated by the value of the UUID field.

Table I.23 – Format of the contents of a UUID box

Field name	Size (bits)	Value
UUID	128	Variable
DATA	Variable	Variable

The existence of any UUID boxes is optional for conforming files. Also, any UUID box shall not contain any information necessary for decoding the image to the extent that is defined within this part of this Recommendation | International Standard, and the interpretation of the information in any UUID box shall not change the visual appearance of the image. All readers may ignore any UUID box.

I.7.3 UUID Info boxes (superbox)

While it is useful to allow vendors to extend JP2 files by adding information using UUID boxes, it is also useful to provide information in a standard form which can be used by non-extended applications to get more information about the extensions in the file. This information is contained in UUID Info boxes. A JP2 file may contain zero or more UUID Info boxes. These boxes may be found anywhere in the top level of the file (the superbox of a UUID Info box shall be the JP2 file itself) except before the File Type box.

These boxes, if present, may not provide a complete index for the UUIDs in the file, may reference UUIDs not used in the file, and possibly may provide multiple references for the same UUID.

The type of a UUID Info box shall be 'uinf' (0x7569 6E66). The contents of a UUID Info box are as in Figure I.20:

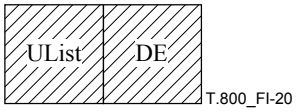

Figure I.20 – Organization of the contents of a UUID Info box

UList: UUID List box. This box contains a list of UUIDs for which this UUID Info box specifies a link to more information. The format of the UUID List box is specified in I.7.3.1.

DE: Data Entry URL box. This box contains a URL. An application can acquire more information about the UUIDs contained in the UUID List box. The format of a Data Entry URL box is specified in I.7.3.2

I.7.3.1 UUID List box

This box contains a list of UUIDs. The type of a UUID List box shall be ‘ulst’ (0x756C 7374). The contents of a UUID List box shall be as in Figure I.21:

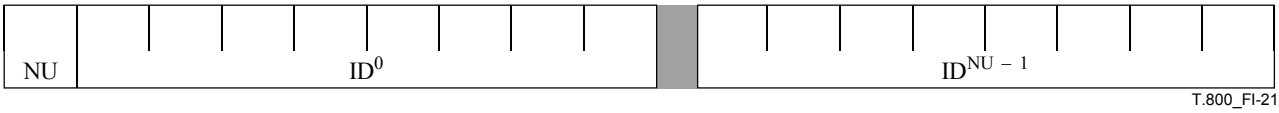

Figure I.21 – Organization of the contents of a UUID List box

NU: Number of UUIDs. This field specifies the number of UUIDs found in this UUID List box. This field is encoded as a 2-byte big endian unsigned integer.

ID^{i}: ID. This field specifies one UUID, as specified in ISO/IEC 11578, which shall be associated with the URL contained in the URL box within the same UUID Info box. The number of UUID^{i} fields shall be the same as the value of the NU field. The value of this field shall be a 16-byte UUID.

Table I.24 – UUID List box contents data structure values

Parameter	Size (bits)	Value
NU	16	0 to (2^{16}– 1)
UUID^{i}	128	0 to (2^{128}– 1)

I.7.3.2 Data Entry URL box

This box contains a URL which can use used by an application to acquire more information about the associated vendor-specific extensions. The format of the information acquired through the use of this URL is not defined in this Recommendation | International Standard. The URL type should be of a service which delivers a file (e.g., URLs of type file, http, ftp, etc.), which ideally also permits random access. Relative URLs are permissible and are relative to the file containing this Data Entry URL box.

The type of a Data Entry URL box shall be 'url\040' (0x7572 6C20). The contents of a Data Entry URL box shall be as in Figure I.22:

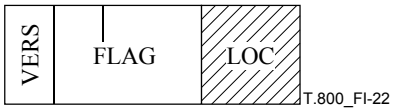

Figure I.22 – Organization of the contents of a Data Entry URL box

VERS: Version number. This field specifies the version number of the format of this box and is encoded as a 1-byte unsigned integer. The value of this field shall be 0.

FLAG: Flags. This field is reserved for other use to flag particular attributes of this box and is encoded as a 3-byte unsigned integer. The value of this field shall be 0.

LOC: Location. This field specifies the URL of the additional information associated with the UUIDs contained in the UUID List box within the same UUID Info superbox. The URL is encoded as a null terminated string of UTF-8 characters.

Table I.25 – Data Entry URL box contents data structure values

Parameter	Size (bits)	Value
VERS	8	0
FLAG	24	0
LOC	varies	varies

I.8 Dealing with unknown boxes

A conforming JP2 file may contain boxes not known to applications based solely on this Recommendation | International Standard. If a conforming reader finds a box that it does not understand, it shall skip and ignore that box.