INTERNATIONAL STANDARD

ISO/IEC 15444-2

First edition
2004-05-15

AMENDMENT 3
2015-07-15

Information technology — JPEG 2000 image coding system: Extensions

AMENDMENT 3: Box-based file format for JPEG XR, extended ROI boxes, XML boxing, compressed channel definition boxes, and representation of floating point

Technologies de l'information — Système de codage d'images JPEG 2000: Extensions

AMENDEMENT 3: Format des fichiers basé sur les boîtes pour JPEG-XR, boîtes de région d'intérêt (ROI) étendues, boîtes XML, boîtes de définition des canaux comprimées et représentation des données à virgule flottante

Reference number
ISO/IEC 15444-2:2004/Amd.3:2015(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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.

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

Amendment 3 to ISO/IEC 15444-2:2004 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.801 (02/2002)/Amd.3.

INTERNATIONAL STANDARD
ISO/IEC 15444-2:2004/FDAM 3(E)
Rec. ITU-T T.801 (02/2002)/Amd.3(200X E)
ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – JPEG 2000 IMAGE CODING SYSTEM: EXTENSIONS

AMENDMENT 3:

Box-based file format for JPEG XR, extended ROI boxes, XML boxing, compressed channel definition boxes, and representation of floating point

1) Clause 2 (References)

Add the following items to the list of normative references:

- ITU-T Recommendation T.805 | ISO/IEC 15444-6, *Information technology - JPEG 2000 image coding system - Part 6: Compound image file format*
- ITU-T Recommendation T.45 *Run-length Colour Encoding*
- ITU-T Recommendation T.832 | ISO/IEC 29199-2, *Information technology – JPEG XR image coding system – Image coding specification*
- IEC 60559 *Binary floating-point arithmetic for microprocessor systems*
- IEC 61966-2-2: *Multimedia systems and equipment – Colour measurement and management – Part 2-2: Colour management – Extended RGB colourspace - scRGB*
- IEEE 754: *IEEE Standard for Floating-Point Arithmetic*

2) Subclause A.3.10 Nonlinearity point transformation (NLT)

Extend the description of the Lnlt syntax element by defining Lnlt to be 6 in case Tnlt equals 0 or 3.

Lnlt: Length of marker segment in bytes (not including the marker). The value of this parameter is determined by the following equation:

$$Lnlt = 6 + \begin{cases} 0 & Tnlt = 0 \quad or \quad Tnlt = 3 \\ 15 & Tnlt = 1 \\ 11 + (N_{points} \cdot \psi_{Tval}) & Tnlt = 2 \end{cases}$$

$$\psi_{Tval} = \begin{cases} 1 & PTval \in [1, 8] \\ 2 & PTval \in [9, 16] \\ 4 & PTval \in [17, 32] \end{cases}$$

(A-7)

Change the description of the Tnlt syntax element to:

Tnlt: Non-linearity type. Table A-44 shows the value for the Tnlt parameter.

Replace Table A-44 by the following:

Value (bits) MSB LSB	Meaning of Tnlt values	STnlt usage
0000 0000	No non-linearity transformation applied	-
0000 0001	Gamma-style non-linearity transformation	Table A-45
0000 0010	LUT-style non-linearity transformation	Table A-46
0000 0011	Binary Complement to Sign Magnitude Conversion	-
	All other values reserved	-

3) Subclause A.3.13 Extended Capabilities

Replace Table A-49 in Subclause A.3.13 by the following (this is defined in AMD.2 of ITU.T 801 | 15444-2):

Parameter	Size (bits)	Value
CAP	16	0xFF50
Lcap	16	6-70
Pcap	32	Table A-50
Ccap[1]	16	Value and meaning specified in ITU.T Rec. 800+(k-1) | ISO/IEC 15444-k, where the i^{th} non-zero bit in Pcap occurs in its k^{th} most-significant bit

4) Subclause K.2 Non-linear transformation specifications

Replace K.2. by the following:

This Recommendation | International Standard allows for the non-linear transformation to be stored in three different forms. The gamma-style non-linearity form specifies the transformation through parameters to an equation, and is specified in subclause K.2.1. The LUT-style non-linearity form specifies the transformation by specifying a set of look up table pairs and is specified in subclause K.2.2. The Binary Complement to Sign Magnitude Conversion transformation takes no parameters, and converts the codestream-internal sample-representation from a binary complement representation to a sign-magnitude representation. This conversion is most suitable for representing floating point data, see subclause O.6 for further informationand its recommended use. The transformation itself is specified in subclause K.2.3.

5) Subclause K.2.3 Binary complement to sign-magnitude conversion transformation

Add a new subclause K.2.3.:

If **Tnlt** equals 3, the data is processed by a transformation that changes the representation of negative sample values. This type of transformation is most useful for representing IEC 60559 floating point data in JPEG 2000 bitstreams. The integer sample values reconstructed from the codestream are then first converted from a binary complement to a sign-magnitude representation using the transformation described below.

NOTE – This transformation is considered to map integer values to integer values, keeping the bit depth and signed-ness of the samples untouched. While not required by this Recommendation | Standard, the resulting bit-patterns are, however, typically interpreted as IEC 60559 floating point numbers by specifying a floating point sample type in the file format. This operation is not part of the JPEG 2000 decoding process, but is a matter of interpreting the reconstructed samples correctly. It is recommended to use the JPX file format defined in Annex M to annotate the data for proper interpretation. Further information on how to encode floating point samples is found in subclause O.6.

If the binary complement to sign-magnitude conversion transformation is used, the bit-depth of the input samples to this transformation shall be equal to the bit-depth of the samples generated by the transformation, and the signed-ness of the input samples shall be equal to the signed-ness of the output samples. That is, the **BDnlt** value relevant to component i shall be equal to either $\mathbf{BDcbd_i}$ if a multi-component transformation is run (see subclause A.3.6, Table A-31), or shall be equal to $\mathbf{Ssiz_i}$ (see subclause A.5.1. of ITU-T Rec. T.800 | ISO/IEC 15444-1, Table A-11) if no such transformation is used.

Let $\mathbf{Z_i}$ be the input sample value of the component i to which this transformation is to be applied, $\mathbf{b_i}$ its bit depth (i.e. $\mathbf{b_i}$=($\mathbf{Ssiz_i}$ AND 0x7f) +1 or $\mathbf{b_i}$=($\mathbf{BDcbd_i}$ AND 0x7f) + 1, and $\mathbf{Y_i}$ the output of the transformation. Then the transformation is defined as:

$$Y_i = Z_i \qquad \text{if } Z_i \geq 0 \qquad \text{or}$$

$$Y_i = \min(-2^{bi-1} - Z_i - 1, -1) \qquad \text{if } Z_i < 0$$

NOTE – This is intentionally the identity transformation for unsigned components. However, readers should be aware that if the level shifting and/or inverse decorrelation transformation of Annex G of ITU-T Rec. T.800 | ISO/IEC 15444-1 is in effect, an additional DC offset will be added to the reconstructed samples. This offset might be undesirable if the intent is to represent floating point data. To prevent this DC offset, use mechanisms from this Recommendation | International Standard such as the Multiple Component Transformation (MCC and MCO markers) from Annex J, or the Variable DC Offset described in Annex B. For signed components, the transformation maps -1 to -2^{bi-1} and -2^{bi-1} to -1. The motivation for this transformation is given in subclause O.6.

6) Subclause M.2.6 – Storage of a codestream within JPX

Append the following text to the end of M.2.6:

The JPX file format also allows for multiple codestreams to be encapsulated within one or more Multiple Codestream boxes, which contains indexing information to facilitate efficient retrieval of specific codestreams of interest, by rendering and applications.

7) Subclause M.2.8 – Support for various pixel formats

Add a new subclause M.2.8 at the end of Clause M.2:

In ITU.T 800 | ISO/IEC 15444-1, channel values consisting of reconstructed image samples or palette entries were always interpreted as signed or unsigned integers. The JPX Recommendation | International Standard extends this by also allowing the representation of fixed point or floating point data. To this end, it introduces the Pixel Format Box (see M.11.7.8) which defines how the values comprised of reconstructed image data or palette entries are to be interpreted as numerical values. The understanding in ITU-T Rec. T.800 | ISO/IEC 15444-1 is that the codestream data encodes and the palette contains signed or unsigned integer values, but this convention no longer holds in the presence of a Pixel Format Box. If this box is present, the integer channel values are re-interpreted in two stages: In the first stage, the integer values reconstructed from the codestream are represented as bit-patterns encoding integers in binary two's complement notation. In the second stage, these bit patterns are re-interpreted as either integers, IEC 60559 floating point data or fixed point data. Only after this interpretation, the samples are considered to define colour values relative to a colourspace.

NOTE – For most computer architectures, the first conversion stage is transparent and requires no additional operation, and the second stage is usually realized as "casting" operation to the target type.

A non-integer pixel format is specified as follows: The desired sample format is encoded in a Pixel Format box (see subclause M.11.7.8) which is a sub-box of the Compositing Layer Header box or the JP2 Header box. The total number of bits required to represent samples in the requested format replaces the channel depth information of integer formats. For example, IEC 60559 single precision floating point numbers require 32 bits for their representation; hence the channel depth will be 32. This information is either recorded in the Image Header box (see subclause M.11.5.1) if the channel depth is identical for all channels, or in the Bits Per Component box (see subclause M.11.5.2) if the channel depth differs across channels. Furthermore, if the channel data is created indirectly through a palette, the channel depth generated by the palette lookup process is indicated in the $\mathbf{B^i}$ field of the Palette box (see subclause I.5.3.4 of ITU-T Rec. T.800 | ISO/IEC 15444-1) which then carries the relevant information for further sample interpretation.

The information on the total number of bits is augmented by information from the Pixel Format box if it is present. In addition to the numerical representation of the samples in the channel it also refines the format by splitting the total number of bits of a sample into integer and fractional, or exponent and mantissa bits. For fixed point data, the Pixel Format box indicates the number of fractional bits, i.e. the number of bits right of the (binary) point, for floating point data it indicates the mantissa bits of the floating point format, not including any implicit (hidden) bits. The number of integer (non-fractional) or exponent bits can then be derived from the total number of bits per sample and the fractional or mantissa bits. Further information on floating point number encodings are found in IEC 60559.

Fixed point and floating point samples are mapped to device colours by means of the Colour Specification box (see subclause M.11.7.2) in the same way as integer samples are mapped to device colours. For that, the Channel Definition box (see subclause M.11.7.5) defines which channels are used to generate which device colours. This process differs for non-integer samples from integer samples only in so far as the maximum intensity of the device colour is no longer represented by the maximum possible integer channel value, but as the value 1.0 expressed in the corresponding fixed point or floating point format. The handling of sample values outside of this range is implementation specific. For further information on the mapping from channel values to device colours, read subclause M.11.7.2.

8) Subclause M.2.9 – Support for JPEG XR codestreams

Add subclause M.2.9 at the end of Clause M.2:

This Recommendation | International Standard also allows the encapsulation of codestreams of other image compression Recommendations | Standards, such as JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2); the JPX file format can represent all metadata encoded in the TIFF-based file format of JPEG XR in boxes defined in this Recommendation | International Standard. A recommendation of how the JPEG XR TIFF-based container should be mapped into the JPX file format is found in subclause O.5.

The File Type Box of an ITU.T Rec. 802 | ISO/IEC 15444-2 file containing an ITU-T Rec. T.832 | ISO/IEC 29199-2 compliant codestream shall have the following values in the compatibility list **CL**^{i} (see ITU-T Rec. T.800 | ISO/IEC 15444-1, subclause I.5.2 and subclause M.11 in this Recommendation | International Standard) if JPEG XR codestreams are present in the file:

Table M-1: Brand Values for JPEG XR Codestreams

Value	Meaning
'jxrc'	JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) compliant bitstream is present
'jxr0'	JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) sub-baseline profile bitstream present and the file conforms to the JPEG XR sub-baseline profile defined in M.9.2.
'jxr1'	JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) baseline profile bitstream present and the file conforms to the JPEG XR baseline profile defined in M.9.2.
'jxr2'	JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) main profile bitstream present and the file conforms to the JPEG XR main profile defined in M.9.2.
'jxr3'	JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-3) advanced profile bitstream and the file conforms to the JPEG XR advanced profile defined in M.9.2.

NOTE - Brand values ‘jxr0’ to ‘jxr3’ indicate JPEG XR profiles of this Recommendation | International Standard. The corresponding profiles are defined in Annex M.9.2.

Relabel tables M-1 through M-24 and all references thereto to M-2 through M-25.

9) Subclause M.5.2 – Sharing header and metadata information between codestreams and compositing layers

Replace first paragraph:

To minimize file overhead, it is useful to allow header and metadata information to be shared between codestreams and compositing layers where that information is identical. The JPX file format provides three mechanisms to share information: default headers, cross-references and label associations.

with the following:

To minimize file overhead, it is useful to allow header and metadata information to be shared between codestreams and compositing layers where that information is identical. The JPX file format provides four mechanisms to share information: default headers, compositing layer extensions, cross-references and label associations.

Relabel subclause M.5.2.3 and all references thereto as M.5.2.4

Relabel subclause M.5.2.2 and all references thereto as M.5.2.3

Insert new subclause M.2.2.2 as follows:

M.5.2.2 Compositing layer extensions

The Compositing Layer Extensions box can be used to specify a repeating pattern of compositing layer headers and (optionally) codestream headers.

A file which contains a Compositing Layer Extensions boxes can be meaningfully interpreted by readers which do not understand the Compositing Layer Extensions box, because all top-level Codestream Header boxes and Compositing Layer Header boxes shall precede any Compositing Layer Extensions box and all compositing instructions found within a Compositions box shall refer only to top-level compositing layers. The Compositing Layer Extensions box may be used only when there are a well-defined number of top-level codestream headers and top-level compositing layers, which means that at least one Codestream Header box and at least one Compositing Layer Header box must appear at the top-level of the file.

A Compositing Layer Extensions box defines one or more additional compositing layers, zero or more additional codestream headers and zero or more additional compositing instructions, to augment the information provided by top-level Codestream Header, Compositing Layer Header and Composition boxes.

Each Compositing Layer Extensions box has an associated repetition factor **Mjclx**. The Compositing Layer Extensions box implicitly defines **Mjclx** × **Cjclx** additional codestream headers and **Mjclx** × **Ljclx** additional compositing layers, where **Cjclx** and **Ljclx** are the number of Codestream Header boxes and the number of Compositing Layer Header boxes that are found within the Compositing Layer Extensions box. The additional codestream headers and compositing layers are assigned consecutive indices, starting from the number of codestream headers and compositing layers defined by all top-level Codestream Header boxes and Compositing Layer Header boxes, together with all preceding Compositing Layer Extensions boxes. The codestreams that are associated with the additional compositing layers are determined from the information found in each embedded Compositing Layer Header box, using a codestream index remapping procedure that accounts for the repetition index. A similar remapping procedure is applied to the Number List boxes which may be found within a Compositing Layer Extensions box, so that embedded metadata is correctly associated.

The principle purpose of Compositing Layer Extensions is to facilitate the efficient description of a large number of compositing layers that follow a simple repeating pattern. This can be particularly beneficial if the JPX file is communicated incrementally via the tools and methods described in IS15444-9.

Compositing Layer Extensions also provide a mechanism for extending the single animation described by a Compositions box into multiple alternate presentation threads – see subclause M.5.3.

10) Subclause M.5.3 – Composition

Replace first paragraph:

Composition data is divided into fixed options, contained in the Composition Options box (subclause M.11.10.1), and a sequence of instructions contained in one or more Instruction Set boxes (subclause M.11.10.2) boxes. Each instruction comprises a set of render parameters. Each instruction set has an associated repeat count which allows for the efficient representation of long sequences of repeating instructions such as occur in full motion sequences or in slide shows which

use a repeated frame transition animation. A JPX file reader shall display a JPX file by reading and executing the instructions in sequence order, from each instruction set in sequence order and repeated according to its repeat value. The file is considered fully rendered either when there are no more instructions to execute, or no compositing layer is present for the current instruction.

with:

Composition data is divided into fixed options, contained in the Composition Options box (subclause M.11.10.1), and a sequence of instructions contained in one or more Instruction Set boxes (subclause M.11.10.2) boxes. Instructions may be further divided into those which appear within a top-level Composition box and those which appear within Compositing Layer Extensions box. The former constitute the primary presentation, while the latter constitute supplementary presentation threads.

Each instruction comprises a set of render parameters. Each instruction set has an associated repeat count which allows for the efficient representation of long sequences of repeating instructions such as occur in full motion sequences or in slide shows which use a repeated frame transition animation. A JPX file reader shall display a JPX file by reading and executing the instructions in sequence order, from each instruction set in sequence order and repeated according to its repeat value. The file is considered fully rendered either when there are no more instructions to execute, or no compositing layer is present for the current instruction.

Instructions found within the top-level Composition box are applied only to top-level compositing layers (i.e., compositing layers other than those defined by Compositing Layer Extensions boxes). Instructions found within a Compositing Layer Extensions box apply only to the compositing layers defined by that box.

11) Subclause M.9.2 – Support for JPX feature set boxes

Replace the entire subclause M.9.2 by the following:

In general, a JPX reader is not required to support the entire set of features defined within this Recommendation | International Standard. However, to promote interoperability, five profiles are defined, of which the first defines a set of baseline features required to decode images using codestream representations conforming to ITU.T 800 | ISO/IEC 15444-1 and ITU.T 801 | ISO/IEC 15444-2 only, and four additional profiles describing images containing only codestreams conforming to ITU.T 832 | ISO/IEC 29199-2.

The ITU.T 80x | ISO/IEC 15444-x based profile is denoted **JPX Baseline** in the following; Files that are written in such a way as to allow a reader that supports only this JPX baseline set of features to properly open the file shall contain a CL$_i$ field in the File Type box with the value 'jpxb' (0x6a70 7862); all JPX baseline readers are required to properly support all files with this code in the compatibility list in the File Type box. The definition of a JPX baseline file given in Annex M.9.2.1 through M.9.2.9, the JPEG XR profiles based on 29199-2 codestreams are defined in Annex M.9.2.10 and following:

12) Subclause M.9.2.10 – JPEG XR Profiles

Insert the following subclauses as Annex M.9.2.10 and following:

In addition to codestreams conforming to the ITU.T 80x | 15444-x series of Recommendations | Standards, a JPX file may also include codestreams conforming to 29199-2 (JPEG XR), and four profiles are defined in the following closely mirroring the profiles of 29199-2. The four profiles are denoted **JPEG XR Sub-baseline Profile**, **JPEG XR Baseline Profile**, **JPEG XR Main Profile**, and **JPEG XR Advanced Profile.** All profiles have in common that files conforming to these profiles shall only contain codestreams conforming to 29199-2.

Files conforming to the JPEG XR profiles shall contain a CL^i field in the File Type box with the values 'jxr0' through 'jxr3', according to the profiles the corresponding 29199-2 codestreams conform to; these compatibilities are defined in Table M-1.

13) Subclause M.9.2.11 – Compression Type

Readers conforming to one of the four JPEG XR profiles only need to support the compression type C = 11 (JPEG XR) indicated in the Image Header Box; see Table M-20 for all compression types defined in this Recommendation | International Standard. Support for other compression types shall not be required to display a file conforming to one of the four JPEG XR profiles.

14) Subclause M.9.2.12 – Compositing Layers

Support for multiple compositing layers is not required to properly display the file; however, the main file may contain multiple compositing layers, but if so, only the first one need to be rendered by an implementation conforming to the JPEG XR profiles. Compositing layers may consist of one or two codestreams that both shall conform to 29199-2. If a compositing layer consists of two codestreams, the two codestreams shall describe images of the same size that are aligned pixel by pixel, and the second codestream shall consist of a single component representing the opacity of the samples encoded in the first codestream. If a second codestream is present in a compositing layer, the first codestream shall not include any opacity information.

15) Subclause M.9.2.13 – Colour Specification

The first composting layer shall contain at least one Color Specification Box from the following list:

- The enumerated method EnumCS value indicating either sRGB, scRGB, sRGB-grey, scRGB-grey, bi-level black on white or bi-level white on black for the JPEG XR baseline and JPEG XR sub-baseline profiles.
- In addition to the above, the enumerated method EnumCS value indicating CMYK or the Any ICC method for the JPEG XR main profile.
- In addition to the above, the enumerated method EnumCS value indicating YCbCr(1) through YCbCr(3) for the JPEG XR advanced profile.

16) Subclause M.9.2.14 – Codestream Fragmentation

The codestreams representing the data of the first compositing layer of files conforming to the JPEG XR profiles shall not be fragmented.

17) Subclause M.9.2.15 – Cross Reference Boxes

Files conforming to the JPEG XR profiles shall not use Cross Reference Boxes for replacing boxes necessary to decode the first compositing layer.

18) Subclause M.9.2.16 – JP2 Header Box Location

The JP2 Header box shall be found in the file before the first Contiguous Codestream box, and Compositing Layer Header box. Any information contained within the JP2 Header box shall be applied to the codestreams encoding the first compositing layer, and as well being used as default information for all other compositing layers; the boxes within the JP2 Header box shall not be found within the Compositing Layer Header box or the Codestream Header box associated with the first compositing layer.

19) Subclause M.9.2.17 – Opacity

A JPEG XR profile conforming reader shall properly interpret opacity channels, through either direct mapping to a codestream component using the Channel Definition box. Other means of indicating opacity, e.g. by the Opacity box, need not to be supported. The use of opacity outside of compositing layers within the JPX file indicates that the decoded image data shall be composited onto an application defined background.

20) Subclause M.9.2.18 – Rotation

A JPEG XR conforming reader shall properly interpret the ROT field of the Instruction Set box defined in Annex M.11.10.2 and Annex M.11.10.2.1. Other instructions defined in the Instruction Set box need not to be honored for compliance to the JPEG XR profiles.

21) Subclause M.9.2.19 – Other Data in the File

A JPEG XR profile file may contain other features or metadata, provided they do not modify the visual appearance of the still image as viewed using a reader that supports only the JPEG XR feature set. All JPX readers should be aware of the existence of this data, as parsing or processing this data may be required in some extended applications. Applications that understand other data or features in the file are encouraged to support the behaviors and functions associated with that extended data.

22) Subclause M.9.2.20 – Conformance Testing

A conformance testing procedure for the JPEG XR profiles as well as test files suitable for conformance testing are defined in ITU.T 834 | ISO/IEC 29199-4.

23) Subclause M.11 – Defined boxes

Edit Table M-6 (now Table M-7) as follows:

- *Add the Pixel Format Box as sub-box to the JP2 Header Box and the Compositing Layer Header Box and add a reference to M.11.7.8.*
- *Add the Compositing Layer Extensions box as a top-level box and add a reference to subclause M.11.21.*
- *Add the Compositing Layer Extensions Info box as a sub-box to the Compositing Layer Extensions box and add a reference to subclause M.11.22.*
- *Add the Multiple Codestream as a top-level box and add a reference to subclause M.11.23. Add a codestream box with reference to M.11.8 and a Fragment Table box with reference to M.11.3.*
- *Add the Multiple Codestream Info box as sub-box to the Multiple Codestream box and add a reference to subclause M.11.24.*
- *Add the Grouping box as a top-level box and add a reference to subclause M.11.25. Add a sub-box labeled "…" to this grouping box.*
- *Add a top-level asoc box to Table M-6 (now Table M-7) with a reference to M.11.11, add a Decomposed XML box as its first sub-box and add a reference to subclause M.11.2.26. Add a second asoc box with reference to M.11.11, and add a XML header box and a reference to subclause M.11.27 as its first sub-box, and a XML box with reference to M.11.18 as its second sub-box.*

24) Table M-13 – Boxes defined within this Recommendation | International Standard

Add the following rows to the table M-13 (now M-14):

Pixel Format box (M.11.7.8)	'pxfm' (0x7078 666d)	NO	This box specifies the interpretation of reconstructed sample values as integer, fixed point or floating point numbers.
XML box (M.11.18)	'xml\040' (0x786D 6C20)	NO	This box contains XML formatted information.
Compositing Layer Extensions box (M.11.21)	'jclx' (0x6A63 6C78)	NO	This box defines an extended set of compositing layers, codestream headers

			and compositing instructions.
Compositing Layer Extensions Info box (M.11.22)	'jlxi' (0x6A6C7869)	NO	This box provides information concerning the repetition factor, compositing layer indices and other attributes of the compositing layers and compositing instructions found within the Compositing Layer Extensions box.
Multiple Codestream box (M.11.23)	'j2cx' (0x6A32 6378)	NO	This box represents a concatenated collection of one or more contiguous codestream boxes or fragment table boxes.
Multiple Codestream Info box (M.11.24)	'j2ci' (0x6A32 6369)	NO	This box contains information describing the Multiple Codestream box in which it is found.
Grouping box (M.11.25)	'grp\040' (0x6772 7020)	NO	This superbox is a container (or wrapper) for any number of boxes which might otherwise be found as the non-initial sub-box of an Association box.
Decomposed XML box (M.11.26)	'dxml' (0x786D 6C64)	NO	This box provides provides front-matter from an XML document as part of a mechanism for decomposing a single XML document into a hierarchical collection of Association boxes.
XML Header box (M.11.27)	'hxml' (0x786D 6C68)	NO	This box provides an element header (the opening element tag with attributes) as part of a mechanism for decomposing a single XML document into a hierarchical collection of Association boxes.

25) Subclause M.11.1 – Reader Requirements box

Add the following rows to Table M-14 (now M-15):

Value	Meaning
75	Codestream contains a JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) compliant bitstream.
76	Codestream contains a Sub-baseline profile JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) compliant bitstream.
77	Codestream contains a Baseline profile JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) compliant bitstream.
78	Codestream contains a Main profile JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) compliant bitstream.
79	Codestream contains an Advanced profile JPEG XR (ITU-T Rec. T.832 | ISO/IEC 29199-2) compliant bitstream.
80	Pixel format "Fixed Point" is used.
81	Pixel format "Floating Point" is used.