



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.813

(06/2012)

SERIES T: TERMINALS FOR TELEMATIC SERVICES
Still-image compression – JPEG 2000

Information technology – JPEG 2000 image coding system – XML representation and reference (JPXML)

CAUTION !

PREPUBLISHED RECOMMENDATION

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU [had/had not] received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2012

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Rec. ITU-T T.813 | ISO/IEC FDIS 15444-14

**Information technology – JPEG 2000 image coding system –
XML representation and reference (JPXML)**

CONTENTS

Page

1	Scope.....	4
2	Normative References.....	4
2.1	Identical Recommendation International Standards	4
2.2	Additional References.....	5
3	Definitions.....	5
4	Conformance Language	6
5	Abbreviations and Symbols	6
5.1	Abbreviations.....	7
6	General Description	7
6.1	Structure of JPXML Document	7
6.2	Creation of JPXML Document	8
6.3	Access with JPXML Document.....	9
7	Document Creation Rules	10
7.1	Common Rule.....	10
7.2	Element Name Rule for Box Format	11
7.3	Element Name Rule for Tagged Image Format.....	11
7.4	Element Name Rule for Marker Segment.....	12
7.5	Element Type Attributes	12
8	Access an Image Data	13
8.1	Rule for Location Conversion.....	13
8.2	Example of location conversion.....	14
Annex A	JPXML Elements for Box File Format.....	16
A.1	Introductions	16
A.2	Box element definitions	16
A.2.1	Still picture format definitions	16
A.2.2	Motion picture format definitions	20
A.2.3	Muliple pages document format definitions	29
A.2.4	Security format definitions	33
A.3	XML Schema Examples	35
A.3.1	Exmaple of XML Schema for common header	35
A.3.2	Example of XML Schema for JPEG 2000 Part 1 image (single image).....	36
A.3.3	Example of XML Schema for JPEG 2000 Part 2 image (single/layered image)	38
A.3.4	Example of XML Schema for JPEG 2000 Part 3 image (motion image)	41
A.3.5	Example of XML Schema for JPEG 2000 Part 6 image (document image).....	42
A.3.6	Example of XML Schema for JPEG 2000 Part 8 image (security)	44
A.3.7	Example of XML Schema for JPEG 2000 Part 12 image (motion image)	48
Annex B	JPXML Elements for Codestream Marker Segments	63
B.1	Introductions	63
B.2	JPEG 2000 Codestream Marker element definitions.....	63
B.2.1	JPEG 2000 Part 1 marker elements.....	63
B.2.2	JPEG 2000 Part 2 marker elements.....	65
B.2.3	JPEG 2000 Part 8 marker elements.....	66
B.2.4	JPEG 2000 Part 10 marker elements.....	68
B.2.5	JPEG 2000 Part 11 marker elements.....	69
B.3	XML Schema Examples	69
B.3.1	Example of XML Schema for common header	69
B.3.2	Example of XML Schema for JPEG 2000 Part 1 codestream (single image)	70
B.3.3	Example of XML Schema for JPEG 2000 Part 2 codestream (single/layered image)..	74
B.3.4	Example of XML Schema for JPEG 2000 Part 8 codestream (security)	77
B.3.5	Example of XML Schema for JPEG 2000 Part 10 codestream (3-D image).....	80

B.3.6	Example of XML Schema for JPEG 2000 Part 11 codestream (wireless).....	82
Annex C	Examples and guidelines.....	83
C.1	Software conventions for the box type	83
C.1.1	Conventions to JPXML element name from a box type	83
C.1.2	Conventions from JPXML element name to a box type	83
C.2	Example of JPXML document conversion	84
C.2.1	JPXML structural document	84
C.2.2	Reference to an internal data in an image	85

.....

INTERNATIONAL STANDARD

ISO/IEC FDIS 15444-14 (E)

Rec. ITU-T T.813 (2012 E)

ITU-T RECOMMENDATION

Information technology – JPEG 2000 image coding system: XML representation and reference (JPXML)

Summary

This Recommendation | International standard specifies an XML document, referred to as JPXML, which is designed primarily for representing JPEG 2000 file format and codestream marker segments, and referring method of internal binary data in a JPEG 2000 image.

This sub-clause contains a list of scope for this XML structural representation and reference.

This Recommendation | International Standard

- specifies XML schema for general box file formats;
- specifies XML schema for JPEG 2000 family file formats and codestream segments;
- specifies a complete referring location path to address to an exact box or codestream data in an image;
- provides guidance on processes for converting a source image data to an XML structural document;
- provides guidance on how to implement these processes in practice.

1 Scope

This Recommendation | International standard specifies an XML document, referred to as JPXML, which is designed primarily for representing JPEG 2000 file format and marker segments in codestream, and referring method of internal data in a JPEG 2000 image.

This Recommendation | International Standard

- specifies JPXML conversion rules for general box file formats.
- specifies JPXML conversion rules for JPEG 2000 family file formats and codestream segments.
- specifies a complete referring location path to address to an exact box or codestream data in an image.
- provides guidance on processes for converting a source image data to an XML structural document.
- provides guidance on how to implement these processes in practice.

2 Normative References

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendation | International Standards

- Recommendation ITU-T T.800 | ISO/IEC 15444-1, *Information technology — JPEG 2000 Image coding system: Core coding system*.
- Recommendation ITU-T T.801 | ISO/IEC 15444-2, *Information technology — JPEG 2000 Image coding system: Extensions*.
- Recommendation ITU-T T.802 | ISO/IEC 15444-3, *Information technology — JPEG 2000 Image coding system: Motion JPEG 2000*.
- Recommendation ITU-T T.805 | ISO/IEC 15444-6, *Information technology — JPEG 2000 Image coding system: Compound image file format*.

- Recommendation ITU-T T.807 | ISO/IEC 15444-8, *Information technology — JPEG 2000 Image coding system: Secure JPEG 2000*.
- Recommendation ITU-T T.808 | ISO/IEC 15444-9, *Information technology — JPEG 2000 Image coding system: Interactivity tools, APIs and protocols*.
- Recommendation ITU-T T.809 | ISO/IEC 15444-10, *Information technology — JPEG 2000 Image coding system: Extensions for three-dimensional data*.
- Recommendation ITU-T T.810 | ISO/IEC 15444-11, *Information technology — JPEG 2000 Image coding system: Wireless*.
- ISO/IEC 15444-12, *Information technology — JPEG 2000 Image coding system: ISO base media file format*.

2.2 Additional References

- Recommendation ITU-T T.832 (03/2009) | ISO/IEC 29199-2:2009, *Information technology — JPEG XR image coding system – Image coding specification*.
- Recommendation ITU-T T.833 (09/2010) | ISO/IEC 29199-3:2010, *Information technology — JPEG XR image coding system – Motion JPEG XR*.
- ISO/IEC 646:1991, *Information technology - ISO 7-bit coded character set for information interchange*.
- ISO 12639:1998, *Graphic technology - Prepress digital data exchange — Tag image file format for image technology (TIFF/IT)*.
- IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One*, November 1996.
- IETF RFC 2279 (1998), *UTF-8, A transformation format of ISO 10646*. January 1998.
- IETF RFC 4648: *The Base16, Base32, and Base64 Data Encodings*, October 2006, Retrieved March 18, 2010.
- W3C Recommendation, *Namespaces in XML*, 8 December 2009.
- W3C Recommendation, *Extensible Markup Language (XML), Version 1.0 (Fifth Edition)*, 26 November 2008.
- W3C Recommendation, *XML Schema Part 0: Primer*, 28 October 2004.
- W3C Recommendation, *XML Schema Part 1: Structures*, 28 October 2004.
- W3C Recommendation, *XML Schema Part 2: Datatypes*, 28 October 2004.
- W3C Recommendation, *XML Path Language (XPath) 2.0*, 14 December 2010.

3 Definitions

For the purposes of this Recommendation | International Standard, the definitions defined in Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2 apply to this Recommendation | International Standard, and also the following definitions are used within this Recommendation | International Standard. In some cases, these definitions differ from those used in other standards and/or Recommendations.

3.1 ...: elision mark. This mark denotes some words or characters are erased or abbreviated.

3.2 4CC: four character codes of the box type generally referred to by an ISO 646 character string translation of the integer value. This value is used for box type which specifies its contents.

3.3 absolute offset: offset to an internal image data from the start of image file. By the JPXML converter, the offset will be made with "length" attributes from the top to the target elements.

3.4 box: sequence of byte block contains its length, 4CC data type, and contents. Some box, such as "jp2c" box, contains an image codestream; other box contains image property such as image width and height. This data block is the atom of JPEG 2000 and MPEG 4 image file format.

3.5 box element: JPXML element for a box, and this element name is translated from 4CC of the box type by using conversion rules described in the later subclause.

3.6 box based format: sequence of box contains several image properties expresses an image file format. This image format start with a signature box, and contains at least one codestream.

3.7 codestream: sequence of bits contained in a sequence of *bytes* from which are created by an image coder. This data sequence contains marker segments for holding image coding properties and parsed by decoder or translator. This may be arranged such as the most significant bit of the first *byte* is the first bit of the *codestream*, the next most significant bit of the first *byte* is the second bit of the *codestream*, and so on, to the least significant *bytes*.

3.8 JPXML converter: converter translates data between an image and a JPXML document. The converter to get the JPXML document is called a "forward" converter, and the converter to get an image is called an "inverse" converter. These converters use rules of element name creation, defined container and un-defined chunk container conversions.

3.9 JPXML document: XML document which corresponds to the box file format or codestream, and this is categorized according to included contents; skeleton, fat skeleton and fat representations. For more detail see 6.1.

3.10 JPXML document structure: structure of a JPXML element in a JPXML document, expresses an image data structure. This structure is used for a location path expressed with XPath.

3.11 JPXML element: XML element represents box file format or codestream structure, and it is translated from box, marker, or its content. The 4CC box type, marker type, and the "content" are used as this element name. For more detail see 6.2.

3.12 location path: location to an internal image data expressed with XPath with a JPXML document. This expression represents the absolute offset value and a JPXML document structure.

3.13 fat representation: JPXML document which contains whole image data on text nodes, and this representation can be translated to a genuine image without any additional image information because of having whole image information. However, in this case, this representation needs more data space than having the original image, because of translating whole chunk data into the XML form. For more detail see 6.1.

3.14 fat-skeleton representation: JPXML document which contains image properties except codestream chunk data. This representation may have a location path to a chunk image data by using JPXML form. This represents a box file format image structure. For more detail see 6.1.

3.15 marker segment: binary data block in a codestream which contains marker type and may contains marker properties for coding information.

3.16 marker element: JPXML element for the marker segment, and this element name is translated from marker type with using conversion rule described in 7.1 and 7.3.

3.17 skeleton representation: JPXML document which does not contains text nodes. This only represents a structure of box file format image. For more detail see 6.1.

4 Conformance Language

This Recommendation | International Standard consists of normative and informative text.

Normative text is that text which expresses mandatory requirements. The word "**shall**" is used to express mandatory requirements strictly to be followed in order to conform to this Specification and from which no deviation is permitted. A conforming implementation is one that fulfils all mandatory requirements.

Informative text is text that is potentially helpful to the user, but not indispensable and can be removed, changed or added editorially without affecting interoperability. All text in this Recommendation | International Standard is normative, with the following exceptions: the Introduction, any parts of the text that are explicitly labelled as "**informative**", and statements appearing with the preamble "**NOTE**" and behaviour described using the word "should". The word "**should**" is used to describe behaviour that is encouraged but is not required for conformance to this Specification.

The keywords "**may**" and "**need not**" indicate a course of action that is permissible in a conforming implementation.

The keyword "**reserved**" indicates a provision that is not specified at this time, shall not be used, and may be specified in the future. The keyword "**forbidden**" indicates "reserved" and in addition indicates that the provision will never be specified in the future.

5 Abbreviations and Symbols

The abbreviations and symbols defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, Rec. ITU-T T.801 | ISO/IEC 15444-2, Rec. ITU-T T.805 | ISO/IEC 15444-6, Rec. ITU-T T.808 | ISO/IEC 15444-9, ITU-T T.810 | ISO/IEC 15444-11, and Rec. ITU-T T.812 | ISO/IEC 15444-13 also apply to this Recommendation | International Standard.

5.1 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply.

JPXML: Refers to this Recommendation | International Standard

MIME: Multipurpose Internet Mail Extensions

TIFF: Tag Image File Format

XML: eXtended Metadata Language

XSLT: XML Stylesheet Language Transformations

6 General Description

The structural representation for the box based format and the JPEG 2000 family codestreams is defined in this Recommendation | International Standard, which does not intend to make a new image file format. This image description is described with XML, and this temporary XML document is created as intermediate image description for accessing to an internal data robustly and converting an image type. The following subclauses describe more details of this document.

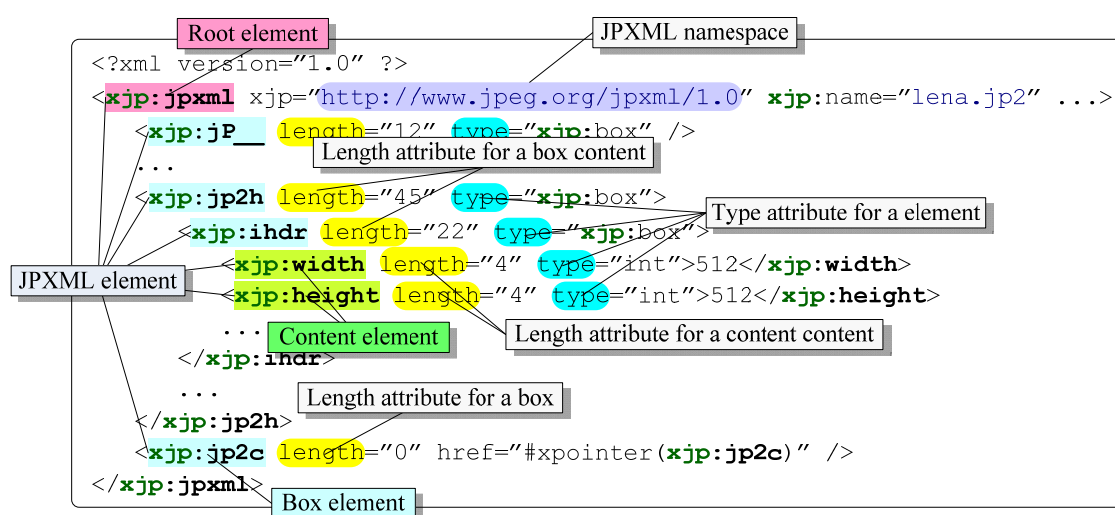


Figure 1 – Example of JPXML document for JPEG 2000 image

6.1 Structure of JPXML Document

The JPXML document is described with three elements; a JPXML element, its attribute, and its content value. The JPXML element structure represents an image structure; box, marker segment, and content structure. This document namespace shall be "http://www.iso.org/jpeg/jpxml/1.0", and this document's root element name shall be 'jpxml'. The JPXML element has two types; the first element is a container element which expresses a box or a marker segment itself, and the second one is a content element which expresses the container's property or a contented box. Some container, such as superbox, contains other containers, and so a JPXML document will have a tree structure. Each JPXML element shall have 'length' and 'type' attributes, and these attributes denote each data chunk's byte length and data type respectively. The content value may be described with text, and its data type denotes with 'type' attribute. Figure 1 is an example of JPXML document for a JPEG 2000 file format.

The container element name, box or marker element name, shall be created with the 4CC box type or the segment marker name respectively, and these converting rules described in Clause 7. The container element may contain some content elements, and these elements are option in the JPXML document. There may be one content element even if a container has several data container, and then this content element type attribute should be "hexbyte" or "unknown" type. This element name shall be "content" or predefined name defined in this Recommendation | International Standard. The attributes in the JPXML elements are used for creating an absolute offset from a location path for pointing a chunk data in the image. The detail of this process is described in 7.5.

The box described in the Recommendation ITU-T T.800 | ISO/IEC 15444-1 and JPXML document for box format are illustrated in Figure 2. The Figure 2 (a), (b), and (c) are illustrations of box format structures; normal box, box with extended length and superbox, and the (d), (e) and (f) documents represent the (a), (b) and (b) box structure respectively.

The superbox element shall have whole children box and superbox elements. All box elements with an extended length have a length element for storing actual box length.

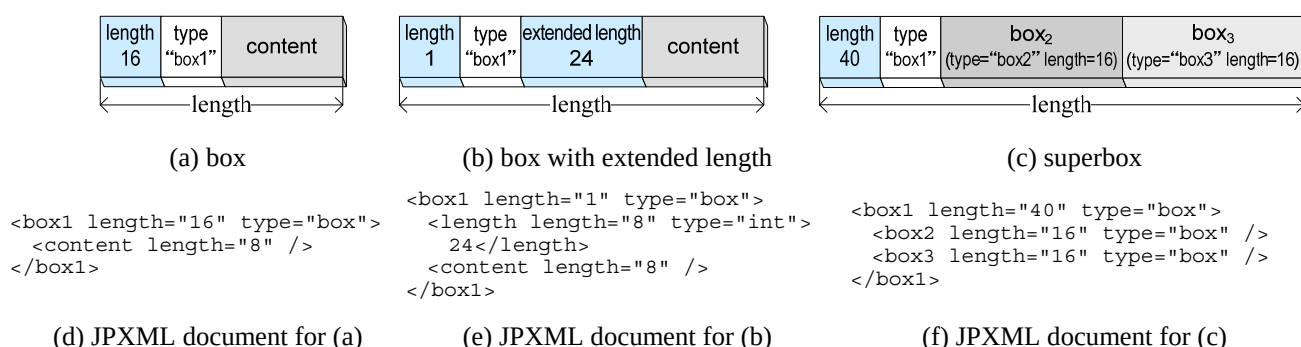


Figure 2 – Examples of box format and JPXML documents

The JPXML document is generated from an image's file format and/or codestreams, and its kind varies from none property to including codestream data representations. With included kinds of image property representation, the JPXML document is categorised as three levels of representations; "skeleton", "fat-skeleton", and "fat" representations.

The first level representation, skeleton, shall express the only structure of image itself, and may contain an attribute for the absolute offset or the location path to the element block. The skeleton shall have no text node in each JPXML elements. This representation is used for a location path which is comparatively robust for changing the box structure of the image and/or marker segment structure of the codestream.

The second level representation, fat-skeleton, expresses the image structure and some variables of box and/or marker contents. The fat-skeleton is an intermediate representation between skeleton and fat representations; consequently, it also has the skeleton's attribute and same text node value of JPXML elements, but no binary data (such as a coded codestream). This representation is used for a location path and also some image transformation with XSLT.

The last one, thirds level representation, fat, expresses the image structure and whole image property values. This whole property may represent as binarized format for use of some application such as secure purpose. The binarized contents are translated with MIME's base64 encoding. This representation requires more data space than an original image data, and so this representation is unsuited for use in a storage file format for image data.

6.2 Creation of JPXML Document

The JPXML document is created from a box based format and/or a JPEG 2000 codestream. This generation process may consist of several steps. Figure 3 is an illustration of a block diagram of an image conversion system using the JPXML, and this illustration includes forward and inverse JPXML generators. This example consists of a forward JPXML generator, an image edit tool, and an inverse JPXML generator. Processes of forward and inverse JPXML generator processes may consist of two modules; a JPXML document generator and a location path generator. The JPXML document generator converts between a binary image and an XML document, and the location path generator converts between an absolute offset number to target data and an XPath location path for a target element.

For creating a JPXML document, the forward JPXML document generator uses several rules; common conversion rule and three element name rules. The common conversion rule is defined in 7.1. The document element name rules are for a box format, a marker segment, and a tagged image format. The inverse JPXML document generator uses inverse rules of the forward converter's rules.

The element name rule for the box based format creates an element name related to the four-character code (4CC) identifying the box container type. All 4CC value is not allowable for an XML element name such as 'xml' and space character, and these 4CC value is modified for an XML element by using conversion rules defined in 7.2. The detail of box based format and four-character code are defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, Rec. ITU-T T.801 | ISO/IEC 15444-2, Rec. ITU-T T.802 | ISO/IEC 15444-3, Rec. ITU-T T.805 | ISO/IEC 15444-6, Rec. ITU-T T.807 | ISO/IEC 15444-8, Rec. ITU-T T.808 | ISO/IEC 15444-9, ISO/IEC 15444-12, and Rec. ITU-T T.833 | ISO/IEC 29199-3. The element name definitions are defined in later Annex A.

The element name rule for the marker segment creates an element name related to a two byte code or a marker segment name identifying the marker segment which is defined in Rec. ITU-T T.800 | ISO/IEC 15444-1, Rec. ITU-T T.801 | ISO/IEC 15444-2, Rec. ITU-T T.807 | ISO/IEC 15444-8, Rec. ITU-T T.809 | ISO/IEC 15444-10, and Rec. ITU-T T.810 | ISO/IEC 15444-11. The two byte code, non character and invisible value, is converted to visible code value for using

as an XML element name by using conversion rules defined in 7.3. The detail of this element names are defined in the Annex B.

The element name rule for the tagged image format creates an element name related to two byte code of tag value identifying the tagged maker property which is defined in the ISO 12639 and Rec. ITU-T T.832 | ISO/IEC 29199-2. The two byte code, non character and invisible value and used in the TIFF image and JPEG XR, is converted to visible code value for using as an XML element name by using conversion rules defined in 7.4.

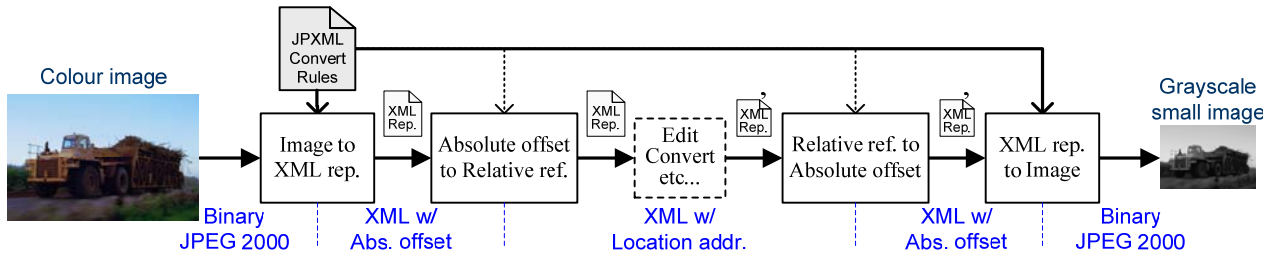


Figure 3 – Block Diagram of Image Converting with JPXML

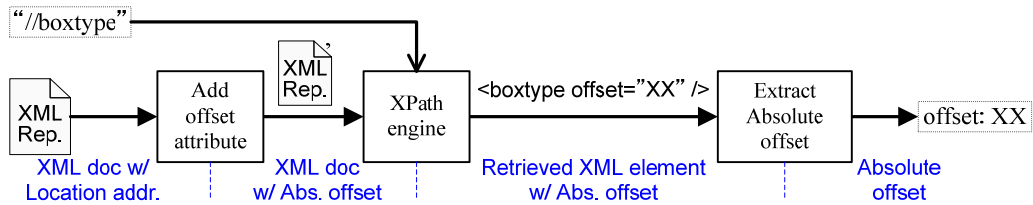


Figure 4 – Example of Converting Process between XML Location Path and Offset

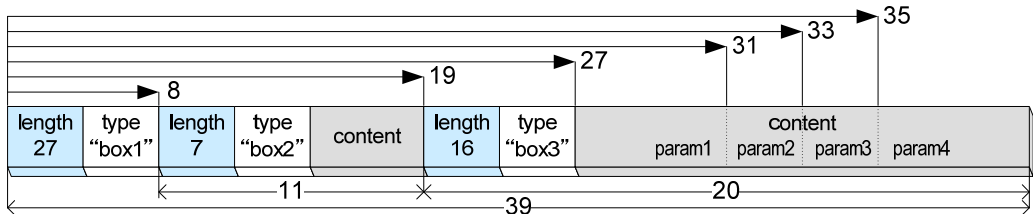
6.3 Access with JPXML Document

The two location representations, a XML location path and an absolute offset, are used for access an internal image data with the JPXML document. The XML location path homologized the target image data identify the target element in the JPXML document. The absolute offset corresponding to the target element identified by the location path is used for locating data chunk of the image, and this value shall be used for a binary data access. For converting between the text location and the offset value, conversion rules defined in 8.1 are used.

An absolute offset can be generated from the target location path with several ways. One example of generating a target offset process consists of three steps; 1) set each element's data chunk offset to its element offset attribute, 2) identify a target element from a target location path by using XML tools, and 3) extract the target element offset attribute value as an absolute offset. Figure 4 is an example of this process. After this process, the generated absolute offset is the location of the target data chunk from beginning of the start of an image.

A target location path can be generated from a target absolute offset with following three steps; 1) set each element's data chunk offset to its element offset attribute, 2) identify a target element by comparing the element's offset attribute and the target offset, and 3) create a location path from the selected target element. This process creates the location path which identifies the target absolute offset, and this text representation will not suffer from binary data changes in the image.

These processes have the same offset generator for the JPXML document, and these generator use conversion rules defined in 8.1. Figure 5 is an example of JPXML document with offset calculated by this generator. As shown in this example, the "box2" box offset is eight bytes because each box has eight bytes data space for box's length and type storage, and the "box2" is first child of "box1" box. By using this generated JPXML document, all image data chunk described by the JPXML elements can be access by the XML location path.



(a) Part of pseudo box based format structure example

```

1: <box1 length="39" type="box" offset="0">
2:   <box2 length="11" type="box" offset="8">
3:     <content length="3" type="int" offset="16" />
4:   </box2>
5:   <box3 length="20" type="box" offset="19">
6:     <param1 length="4" type="int" offset="27" />
7:     <param2 length="2" type="short" offset="31" />
8:     <param3 length="2" type="short" offset="33" />
9:     <param4 length="4" type="int" offset="35" />
10:   </box3>
11: </box1>

```

(b) Part of a pseudo JPXML document with offset attribute for the (a) format

```

1: <box3 length="16" type="box" offset="19">
2:   <content length="12" type="hexBinary" offset="27" />
3: </box3>

```

(c) Another representation of the box3 element in the (b) document (single content representation)

Figure 5 – JPXML document accessing example

7 Document Creation Rules

The conversion rules between the JPXML document and the image consist of three parts. The first part is a common conversion rule for file formats and codestreams, the second part is a conversion rule between 4CC and XML element names, and the third part is a conversion rule between marker segment names and XML element names.

7.1 Common Rule

All JPXML forward converters shall use following common forward rules for JPXML conversion. The boxes within a super box shall be represented in JPXML form when these rules are used for a file format forward conversion. An 'offset' attribute value shall be converted to a location path using with a JPXML document structure.

- (1) The namespace of the JPXML document shall be "http://www.iso.org/jpeg/jpxml/1.0".
- (2) The 'jpxml' shall be used for the JPXML root element, represents one file format or codestraem.
- (3) The root element may have a 'name' attribute for identifying an image name.
- (4) All elements shall have 'length' and 'type' attributes, represents data byte length and type respectively.
- (5) The 'type' attributes value for the box or marker element shall be 'box' or 'marker' respectively.
- (6) A box element with length=1 attribute shall have a length element, represents the data length in byte.
- (7) All element may have an 'offset' attribute, represents the absolute offset to the data chunk in byte.
- (8) The box element shall have several box elements when the original box have box inside its contenet.
- (9) The element of box or marker segment may have content elements for representing its properties.
- (10) The predefined element names defined in later Annexes shall be used for element name.
- (11) The box or marker parameters may be represented as XML form, defined in later Annexes, and stored in the JPXML box or marker element.
- (12) The box or marker parameters may be represented as base64 binary, and stored in the box or marker element.

All JPXML inverse converter shall use following common inverse rules for JPXML conversion. A box element having one or more box elements shall be converted to a super box when reconstructing a file format. If an 'offset' attribute value is the location path using a JPXML document structure, this location path shall be converted to an absolute offset value, and stored in its element attribute.

- (1) The JPXML document shall have namespace of "http://www.iso.org/jpeg/jpxml/1.0".
- (2) The 'jpxml' root element shall be converted to one file format or codestream.
- (3) The 'name' attribute value in the root element shall be used for a file name.
- (4) The 'length' and 'type' attribute values shall be used for converted data length and type respectively.

- (5) The element having 'box' or 'marker' type shall be converted to box or marker segment respectively.
- (6) The node value of the length element in the box element with length=1 attribute shall be used for its converted data chunk length.
- (7) The 'offset' attribute value may be used for its converted data chunk location in the image.
- (8) All box elements in the box shall be converted into the box content.
- (9) All child elements in a box or marker element shall be converted to internal contents of the data chunk.
- (10) All content elements' content shall be combined to a binary data of the parent element.
- (11) The JPXML content elements defined in later Annexes shall be converted to binary data and stored its one of parent contents of the box or maker.
- (12) The JPXML content in base64 binary shall be converted to a binary data with base64 converter and stored in its one of parent contents of the box or maker.

Table 1 – Example of 4CC box type conversions

Box name	Box type	Hex decimal	JPXML element name
JPEG 2000 Signature box	jp\040\040	0x6A50 2020	jp__
JP2 Header box	jp2h	0x6A70 3268	jp2h
Resolution box	url\040	0x7572 6C20	url__
URL box	res\040	0x7265 7320	res__
XML box	xml\040	0x786D 6C20	__xml__

7.2 Element Name Rule for Box Format

The JPXML forward converter for file formats, translates a box based format to a JPXML document, shall use following forward conversion rule for 4CC.

- (1) The JPXML element name shall use a 4CC box type.
- (2) The alpha-numeric in 4CC box type shall be directly used for the element name.
- (3) The space, '\040' code, shall be represented with a '_' character for the JPXML element name.
- (4) The code '.HH' (H: hexadecimal character = 0, ..., 9, A, ..., F) shall be used for any other characters.
- (5) The '_' character at the first character of the element name shall be the escape character.

The JPXML inverse converter to file formats, translates a JPXML document to a box based format, shall use following inverse conversion rule for 4CC.

- (1) The JPXML element name shall be converted to a 4CC box type, and creates its type box.
- (2) The '_' character at the first character of the element name may not be removed from the 4CC box type.
- (3) The alpha-numeric name shall be directly used for 4CC box type.
- (4) The '_' character in the element name space shall be represented with a '\040' code for the 4CC.
- (5) The '.HH' string in the name shall be converted to a '0xHH' code character.

7.3 Element Name Rule for Tagged Image Format

The JPXML forward converter for tagged image file formats, translates a tag value in a directory entry to a JPXML document element name, shall use following forward conversion rule.

- (1) The two bytes tag value shall be represented as the four characters hex string notation, 'HHHH' (H: hexadecimal character = 0, ..., 9, A, ..., F).
- (2) The '_' character shall be placed on the front of the four character string, and create five characters string ('_HHHH').

The JPXML inverse converter to tagged image file formats, translates a JPXML document name to a tag value in a directory entry, shall use following inverse conversion rule for 4CC.

- (1) The first '_' character shall be removed from the five characters string, and create a four characters string.

- (2) The four characters string, 'HHHH', shall be converted to a '0xHHHH' code value

7.4 Element Name Rule for Marker Segment

The JPXML forward converter for marker segments, translates JPEG 2000 marker segments to a JPXML document, shall use following forward conversion rule.

- (1) The marker symbol name shall be used for a JPXML codestream element name, or the forward conversion rule for tagged image file format shall be used for converting marker symbol code to an element name.

The JPXML inverse converter to marker segments, translates a JPXML document to JPEG 2000 marker segments, shall use following inverse conversion rule.

- (1) All elements named with its marker symbol name shall be converted to its named marker symbol code.
- (2) All elements named start with '_' character of the five characters string shall be converted to the marker symbol code by using the inverse conversion rule for tagged image file format.

Table 2 – Example of codestream marker conversions

Codestream marker name	Symbol name	Code	JPXML element
Start of codestream	SOC	0xFF4F	SOC
Start of tile-part	SOT	0xFF90	SOT
Start of data	SOD	0xFF93	SOD
End of codestream	EOC	0xFFD9	EOC
Start of packet	SOP	0xFF91	SOP

7.5 Element Type Attributes

All elements have a type attribute which identify the datatype of the element content data. This Recommendation | International Standard defines uses six types datatypes; box, marker, fourcc, location, hexbyte, integer and time. The integer and time datatypes are used for the element content data type as integer and time types respectively and these datatypes are defined in the XML Schema Part 2 recommendation. The box and marker datatypes are used for identify the box element and the marker element respectively. The fourcc datatype indicates the element content data shall be the 4CC datatype, and the 4CC value will not converted with previous described rules. The location indicates the element have any location type value. The hexbyte indicates the element data may be represented as hexadecimal value in big-endian order. The XML definitions for the box, marker, fourcc, location and hexbyte datatypes can be defined by following XML Schema's simple type datatype definitions.

```

<xs:simpleType name="box">
  <xs:restriction base="xs:string" />
</xs:simpleType>

<xs:simpleType name="marker">
  <xs:restriction base="xs:string" />
</xs:simpleType>

<xs:simpleType name="fourcc">
  <xs:restriction base="xs:string">
    <xs:pattern value="[ a-zA-Z0-9_]{4}" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="location">
  <xs:restriction base="xs:anyURL" />
</xs:simpleType>

<xs:simpleType name="hexbyte">
  <xs:restriction base="xs:string">
    <xs:pattern value="([a-f0-9][a-f0-9])+" />
  </xs:restriction>
</xs:simpleType>

```

8 Access an Image Data

The XML location path and the JPXML document can be used to access an internal binary chunk data in the image data. To access the internal binary data will be accomplished with following three steps. First, the target element is identified by using the location path and the JPXML document, and then the target element is converted to an absolute offset, finally the internal image data is accessed with the target absolute offset position. To achieve the second step, the following rules are employed.

8.1 Rule for Location Conversion

The following conversion rules are for calculating an absolute offset of the target chunk from the location path point to the target element which corresponds to the offset. To be accomplished these rules; there may be two calculation approaches to get the target absolute offset value, the target to the root approach and the root to the target approach.

- (1) The absolute offset shall be the sum of length attributes values between the top and the target elements except following elements.
- (2) For the superbox's child elements, the value 8 shall be used for offset counting instead of the superbox's length attribute value.
- (3) The length of the child boxes contained in the parent box ended at the before of the target box will not be used for offset counting, and the length of the parent is used for counting.

Figure 6 and Figure 7 are illustrations of an algorithm example for calculating the absolute offset value from the target element using with these rules. In these algorithms, the root element shall not be used for the target element, because the root element is not a part of image data. The first algorithm calculates with all element nodes from the target to the root elements, and the second algorithm calculates with all element nodes from the first child of the root to the target elements.

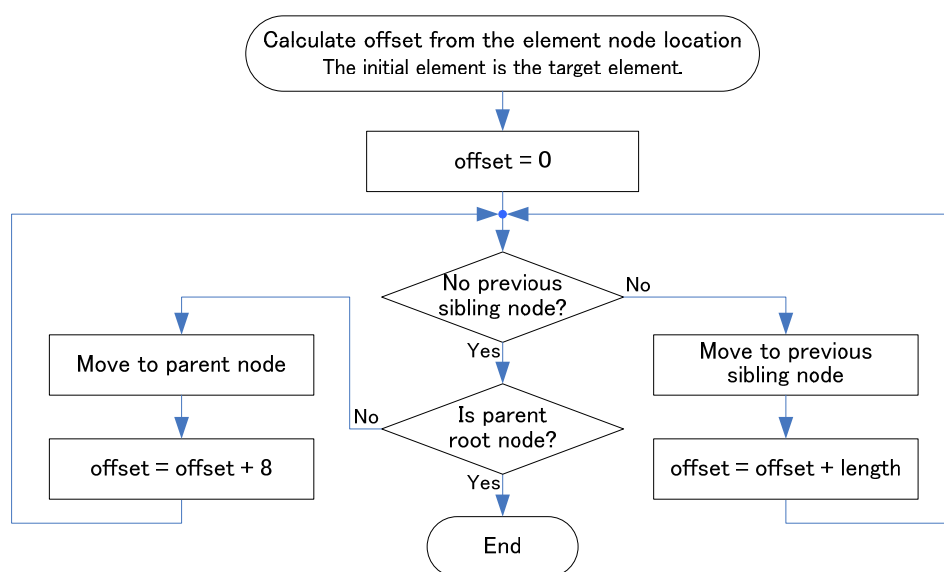


Figure 6 – Algorithm example 1 for Calculating an Offset value from the Target Element

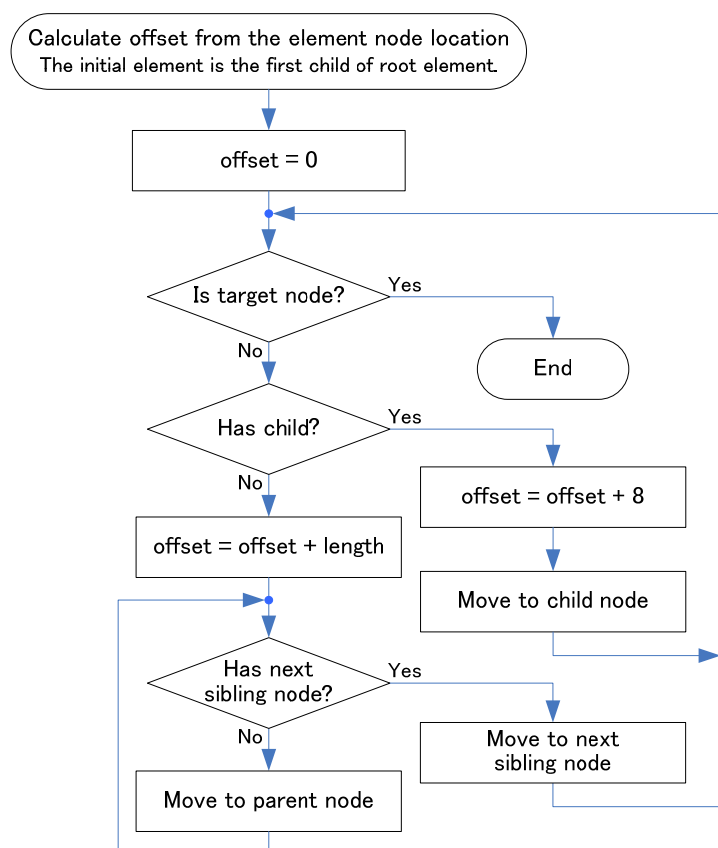


Figure 7 – Algorithm example 2 for Calculating an Offset value from the Target Element

8.2 Example of location conversion

Figure 8 shows an example of offset calculations for a pseudo JPXML document. This figure's second and third columns, labelled "offset 1" and "offset 2", are the result of the target to the root approach and the root to the target approach respectively and there values are calculated by the algorithms in Figure 6 and Figure 7 respectively. In this figure, the offset 1, $a+b+d+8$, for the "BoxH" box is not same as the offset 2, $a+b+f+g+24$. However the "boxD" box length, d , is same as summation of its content and header lengths, $f+g+16$, and so these offsets are same value.

Line	Pseudo JPXM document	offset 1	offset 2
1	<jpxml ... >		
2	<BoxA length="a" type="box" />	0	0
3	<BoxB length="b" type="box" />	a	a
4	<BoxC length="c" type="box">	a+b	a+b
5	<BoxD length="d" type="box">	a+b+8	a+b+8
6	<BoxE length="e" type="box">	a+b+16	a+b+16
7	<BoxF length="f" type="box" />	a+b+24	a+b+24
8	<BoxG length="g" type="box" />	a+b+f+24	a+b+f+24
9	</BoxE>		
10	</BoxD>		
11	<BoxH length="h" type="box" />	a+b+d+8	a+b+f+g+24
12	</BoxC>		
13	<BoxI length="i" type="box" />	a+b+c	a+b+f+g+h+24
14	<BoxJ length="j" type="box">	a+b+c+i	a+b+f+g+h+i+24
15	<BoxK length="k" type="box" />	a+b+c+i+8	a+b+f+g+h+i+32
16	<BoxL length="l" type="box">	a+b+c+i+k+8	a+b+f+g+h+i+k+32
17	<length length="8" type="integer" />	a+b+c+i+k+16	a+b+f+g+h+i+k+40
18	<BoxM length="m" type="box" />	a+b+c+i+k+24	a+b+f+g+h+i+k+48
19	</BoxL>		
20	</BoxJ>		
21	</BoxN length="n" type="box" />	a+b+c+j	a+b+f+g+h+i+k+m+48
22	</jpxml>		

Figure 8 – Generated offsets example for a Pseudo JPXML document

Annex A

JPXML Elements for Box File Format

(This annex forms an integral part of this Recommendation | International Standard)

A.1 Introductions

This Annex provides the content element name definitions, length and type attributes for JPEG 2000 family formats. The box element appearance and structure shall be obedient to the Rec. ITU-T T.800 | ISO/IEC 15444-1, Rec. ITU-T T.801 | ISO/IEC 15444-2, Rec. ITU-T T.802 | ISO/IEC 15444-3, Rec. ITU-T T.805 | ISO/IEC 15444-6, Rec. ITU-T T.807 | ISO/IEC 15444-8, Rec. ITU-T T.808 | ISO/IEC 15444-9, and ISO/IEC 15444-12.

A.2 Box element definitions

This Subclause includes many tables of the box content element definitions, and these tables define the content element name, the content data length in byte, and the content datatype. The JPXML datatypes are described in 7.5. Some tables define the sub-content element names of a defined content element, and some box or content elements have two or more types' content element definitions. For more detail information of these box content meaning and structure, please see each Recommendation | International Standard.

A.2.1 Still picture format definitions

The box file formats defined in Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2 are designed to represent a still picture image. Table A.1 illustrates element structures of these file formats. These content element names and types are defined in following tables.

Table A.1 – Box element structure for a still image

Element name	Box name	Part 1	Part 2	Defined
jp_	JPEG family signature box	required	required	1
ftyp	File type and compatibility box	required	required	1
rreq	Reader requirement box		required	2
jp2h	JP2 header box (superbox)	required	O	1, 2
lbl_	Label box		O	2
ihdr	Image header box	required	required	1, 2, 6
bpcc	Bits per component box	O	O	1, 2
colr	Colour specification box	required	required	1, 2, 6
pclr	Palette box	O	O	1
cmap	Component mapping box	O	O	1
cdef	Channel description box	O	O	1
res_	Resolution box (superbox)	O	O	1
resc	Capture resolution box	O	O	1
resd	Display resolution box	O	O	1
jpch	Codestream header box (superbox)		O	2
lbl_	Label box		O	2
ihdr	Image header box		required	1, 2, 6
bpcc	Bits per component box		O	2
pclr	Palette box		O	1
cmap	Component mapping box		O	1
jp1h	Compositing layer header box (superbox)		O	2
lbl_	Label box		O	2
cgrp	Colour group box (superbox)		O	2
colr	Colour specification box		O	1, 2, 6
opct	Opacity box (superbox)		O	1, 2
cdef	Channel description box		O	1
creg	Codestream registration box		O	2
pxfm	Pixel Format box		O	2
res_	Resolution box (superbox)		O	1
dtbl	Data reference box		O	2, 6, 12

Element name	Box name	Part 1	Part 2	Defined
ftbl	Fragment table box (superbox)		O	2, 6
flst	Fragment list box		O	2, 6
jp2c	Contiguous codestream box	required	O	1, 2, 6
mdat	Metadata box		O	2
comp	Composition box		O	2
copt	Composition options box		O	2
inst	Instruction set box		O	2
drep	Desired reproductions box (superbox)		O	2
gtso	Graphics tech. standard output box		O	2
roid	ROI description box		O	2
cref	Cross reference box (superbox)		O	2, 6
flst	Fragment list box		O	2, 6
asoc	Association box (superbox)		O	2
dxml	Decomposed XML box		O	2
nlst	Number list box		O	2
lbl_	Label box		O	2
bfil	Binary filter box		O	2
chck	Digital signature box		O	2
mp7b	MPEG-7 binary box		O	2
free	Free space box		O	2
xml_	XML container box	O	O	1, 2, 12
jp2i	IPR box	O	O	1
uuid	UUID box	O	O	1
uinf	UUID information box (superbox)	O	O	1
ulst	UUID list box	O	O	1
url_	Data entry box	O	O	1
jclx	Compositing Layer Extensions box		O	2
jlx_i	Compositing Layer Extensions Info box		O	2
j2cx	Multiple Codestream box		O	2
j2ci	Multiple Codestream Info box		O	2
grp_	Grouping box		O	2

A.2.1.1 Box element definitions for JPEG 2000 part 1 format

Table A.2 - Content element name for jp_

content	length	type	loop
signature	12	hexbyte	

Table A.3 - Content element name for ftyp

content	length	type	loop
brand	4	fourcc	
version	4	integer	
compatibility	4	fourcc	1

Table A.4 - Content element name for jp2h

content	length	type	loop
ihdr	-	box	=1
bpcc	optional	box	≤1
colr	-	box	≥1
pclr	optional	box	≥0
cmap	optional	box	≤1
cdef	-	box	≤1
res_	-	box	≤1

Table A.5 - Content element name for ihdr

content	length	type	loop
height	4	integer	
width	4	integer	
num_components	2	integer	
depth	1	integer	
compression	1	integer	
colour_unknown	1	integer	
ipr	1	integer	

Table A.6 - Content element name for colr

Content	length	type	loop
method	1	integer	
precedence	1	integer	
approx	1	integer	
colour	≥4	hexbyte	

Table A.7 - Content element name for bpcc

content	length	type	loop
depth	1	integer	1

Table A.8 - Content element name for pclr

content	length	type	loop
num_entries	2	integer	
num_components	1	integer	
depth	1	integer	]
data	num_entries	hexbyte	]

Table A.9 - Content element name for cdef

content	length	type	loop
num_entries	2	integer	
index	2	integer	]
type	2	integer	
assoc	2	hexbyte	

Table A.10 - Content element name for res_

content	length	type	loop
resc	-	box	≤1
resd	-	box	≤1

Table A.11 - Content element name for resc & resd

content	length	type	loop
vert_num	2	integer	
vert_den	2	integer	
hori_num	2	integer	
hori_den	2	integer	
vert_exp	2	integer	
hori_exp	2	integer	

A.2.1.2 Box element definitions for JPEG 2000 part 2 format**Table A.18 - Content element name for rreq**

content	length	type	loop
length	1	integer	
fua_mask	length	hexbyte	
dc_mask	length	hexbyte	
num_std_flags	2	integer	
std_flag	2	integer	]
std_mask	length	hexbyte	
num_vender_features	2	integer	
vender_feature	16	integer	]
vender_mask	length	hexbyte	

Table A.19 - Content element name for dtbl

content	length	type	loop
num_entries	2	integer	
location	variable	location	

Table A.20 - Content element name for lbl_

content	length	type	loop
label	variable	string	

Table A.21 - Content element name for mp7b

content	length	type	loop
mpeg7	variable	hexbyte	

Table A.22 - Content element name for ftbl

content	length	type	loop
flst	-	box	≥0

Table A.23 - Content element name for free

content	length	type	loop
data	0–65,531	hexbyte	

Table A.12 - Content element name for uuid

content	length	type	loop
id	16	integer	
data	0–65,531	hexbyte	

Table A.13 - Content element name for uinf

content	length	type	loop
ulst	-	box	=1
url_	-	box	=1

Table A.14 - Content element name for ulst

content	length	type	loop
num_entries	2	integer	
data	16	hexbyte	]

Table A.15 - Content element name for url_

content	length	type	loop
location	0–65,531	location	

Table A.16 - Content element name for _xml_

content	length	type	loop
text	0–65,531	string	

Table A.17 - Content element name for jp2c

content	length	type	loop
data	≥0	hexbyte	

Table A.24 - Content element name for flst

content	length	type	loop
num_entries	2	integer	
offset	8	integer	]
length	4	integer	
data_reference	2	integer	

Table A.25 - Content element name for cref

content	length	type	loop
type	4	integer	=1
flst	-	box	=1

Table A.26 - Content element name for jpch

content	length	type	loop
lbl_	-	box	≥0
ihdr	-	box	
bpcc	-	box	
pclr	-	box	
cmap	-	box	
roid	-	box	

Table A.27 - Content element name for jplh

content	length	type	loop
lbl_	-	box	≥0
cgrp	-	box	≥0
opct	-	box	≥0
cdef	-	box	≥0
creq	-	box	≥0
res_	-	box	≤1

Table A.28 - Content element name for cgrp

content	length	type	loop
---------	--------	------	------

colr	0–65,531	box	≥1
------	----------	-----	----

Table A.29 - Content element name for colour

colour	length	type	loop
enum_colour	4	integer	
enum_param	variable	hexbyte	

colour	length	type	loop
vender_colour	16	integer	
vender_param	variable	hexbyte	

Table A.30 - Content element name for enum_param

enum_param	length	type	loop
range_L	4	integer	
offset_L	4	integer	
range_a	4	integer	
offset_a	4	integer	
range_b	4	integer	
offset_b	4	integer	
illuminant	4	integer	

enum_param	length	type	loop
range_J	4	integer	
offset_J	4	integer	
range_a	4	integer	
offset_a	4	integer	
range_b	4	integer	
offset_b	4	integer	

Table A.31 - Content element name for opct

content	length	type	loop
type	1	integer	
num_entries	1	integer	
data	1	integer	1

Table A.32 - Content element name for drep

content	length	type	loop
gtso	-	box	

Table A.33 - Content element name for gtso

content	length	type	loop
profile	variable	hexbyte	

Table A.34 - Content element name for creg

content	length	type	loop
hori_size	2	integer	
vert_size	2	integer	
cds_num	2	integer	
hori_resolution	1	integer	
vert_resolution	1	integer	
hori_offset	1	integer	
vert_offset	1	integer	

Table A.35 - Content element name for comp

content	length	type	loop
copt	-	box	=1
inst	-	box	

Table A.36 - Content element name for copt

content	length	type	loop
height	4	integer	
width	4	integer	
loop	1	integer	

Table A.37 - Content element name for inst

content	length	type	loop
type	2	integer	
repetition	2	integer	
duration	1	integer	
instruction	variable	hexbyte	

Table A.38 - Content element name for instruction

instruction	length	type	loop
hori_offset	0, 4	integer	
vert_offset	0, 4	integer	
width	0, 4	integer	
height	0, 4	integer	
life	0, 4	integer	
next_use	0, 4	integer	
hori_crop_offset	0, 4	integer	
vert_crop_offset	0, 4	integer	
crop_width	0, 4	integer	
crop_height	0, 4	integer	

Table A.39- Content element name for roid

content	length	type	loop
num_entries	1	integer	
contained	1	integer	
type	1	integer	
priority	1	integer	
left	4	integer	
top	4	integer	
width	4	integer	
height	4	integer	

Table A.40 - Content element name for asoc

content	length	type	loop
Association box	-	box	=1
Associated box	-	box	1

Table A.41- Content element name for nlst

content	length	type	loop
asoc_num	4	integer	1

Table A.42- Content element name for bfil

content	length	type	loop
type	16	integer	
data	variable	hexbyte	

Table A.43- Content element name for chck

content	length	type	loop
signature_type	1	integer	
source_type	1	integer	
offset	0, 8	integer	
length	0, 8	integer	
data	length	hexbyte	

A.2.2 Motion picture format definitions

The box file formats defined in Rec. ITU-T T.802 | ISO/IEC 15444-3 and ISO/IEC 15444-12 are designed to represent a motion picture image. Table A.44 is the box element and their element structures for these file formats. These content element names and types are defined following tables.

Table A.44 – Box element structure for a motion/video image

Element name	Box name	Part 3	Part 12
jp_	JPEG family signature box	required	
ftyp	File type and compatibility box	required	required
pdin	Progressive download information box		O
moov	Movie box (superbox)	required	required
mvhd	Movie header box	required	required
trak	Track box (superbox)	required	required
tkhd	Tack header box	required	required
tref	Track reference container box	O	O
edts	Edit list container box (superbox)	O	O
elst	Edit list box	O	O
udta	User-data box (superbox)	O	O
cpri	Copyright box	O	O
tsel	Track Selection box	O	O
mdia	Media box (superbox)	required	required
mdhd	Media header box	required	required
hdlr	Handler box	required	required
minf	Media information box (superbox)	required	required
vmhd	Video media header box	O	O
smhd	Sound media header box	O	O
hmhd	Hint media header box	O	O
dinf	Data information box (superbox)	required	required
dref	Data reference box	required	required
url_ urn_	Data Entry URL or Data Entry URN	O	O
url_	Data Entry URL	O	O
urn_	Data Entry URN	O	O
stbl	Sample table box (superbox)	required	required
std	Sample descriptions box	required	required
mjp2	Video Sample Entry box	O	
jp2h	JP2 header box (superbox)	required	
fiel	Field Coding box	O	
jp2p	Motion JPEG 2000 Profile box	O	
jp2x	Motion JPEG 2000 Prefix box	O	
jsub	Motion JPEG 2000 Sub-sampling box	O	
orfo	Motion JPEG 2000 Original Format box	O	O
raw_	Audio Sample Entry box	O	
twos	Audio Sample Entry box	O	
stts	Time-to-sample box (decoding)	required	required
ctts	Time-to-sample box (composition)		O
stsc	Sample-to-chunk box	required	required
stsz	Sample sizes box	required	O
stz2	Compact sample size box		O
stco co64	Chunk offset box (32 bit or 64 bit)	required	required
stss	Sync sample table box		O
stsh	Shadow sync sample table box		O
padb	Sample padding bits box		O
stdp	Sample degradation box		O
sdtg	Independent and disposable sample box		O
sbgp	Sample-to-group box		O

Element name				Box name	Part 3	Part 12
			sgpd	Sample group description box		O
			subs	Sub-sample information box		O
		meta		Metadata box		O
	meta			Metadata box		O
	udta			User-data box (superbox)	O	O
	mvex			Movie extends box (superbox)	O	O
	mehd			Movie extends header box	O	O
	trex			Track extends defaults box	required	required
		ipmc		IPMP control box		O
mdat				Media data container box	O	O
moof				Movie fragment box (superbox)	O	O
	mfhd			Movie fragment header box	required	required
	traf			Track fragment box (superbox)	O	O
	tfhd			Track fragment header box	required	required
	trun			Track fragment run box	O	O
	sdtg			Independent and disposable sample box		O
	sbgp			Sample-to-group box		O
	subs			Sub-sample information box		O
mfra				Movie fragment random access box (superbox)	O	O
	tfra			Track fragment random access box	O	O
	mfro			Movie fragment rand. access offset box	required	required
meta				Metadata box		O
	hdlr			Handler box		required
	dinf			Data information box (superbox)		O
	ipmc			IPMP control box		O
	iloc			Item location box		O
	ipro			Item protection box		O
		sinf		Protection scheme information box (superbox)		O
		frma		Original format box		O
		imif		IPMP information box		O
		schm		Scheme type box		O
		schi		Scheme information box (superbox)		O
	iinf			Item information box		O
	xml			XML container box		O
	bxml			Binary XML container box		O
	pitm			Primary item reference box		O
	fiin			File delivery item information box		O
		paen		Partition entry box (superbox)		O
		fpar		File pattern box		O
		fecr		FEC reservoir box		O
		sgr		File delivery session group box		O
		gitn		Group id name box		O
		tsel		Track selection box		O
	dinf			Data information box (superbox)		O
meco				Additional metadata container box (superbox)		O
	mere			Metabox relation box		O
	meta			Metadata box		O
skip	free			Free space box	O	O

A.2.2.1 Box element definitions for JPEG 2000 part 3 format

Table A.45 - Content element name for mjp2

content	length	type	loop
reserved	6	integer	
data_reference	2	integer	
predefined	2	integer	
reserved	2	integer	
predefined	12	integer	
width	2	integer	
height	2	integer	
hori_resolution	4	integer	
vert_resolution	4	integer	
reserved	4	integer	
predefined	2	integer	
name	32	string	
depth	2	integer	
predefined	2	integer	
jp2h	-	box	
fiel	optional	box	
jp2p	optional	box	
jp2x	optional	box	
jsub	optional	box	
orfo	optional	box	

Table A.46 - Content element name for fiel

content	length	type	loop
field_count	1	integer	
field_order	1	integer	

Table A.47 - Content element name for jp2p

content	length	type	loop
version	1	integer	
flag	3	integer	
compatibility	4	integer	

A.2.2.2 Box element definitions for JPEG 2000 part 12 format

Table A.53 - Content element name for mdat

content	length	type	loop
data	variable	hexbyte	

Table A.54 - Content element name for pdin

content	length	type	loop
version	1	integer	
flag	3	integer	
rate	4	integer	]
inital_delay	4	integer	

Table A.55 - Content element name for moov

content	length	type	loop
mvhd	-	box	=1
trak	-	box	≥1
mvex	optional	box	≤1
udta	optional	box	≤1
meta	optional	box	≤1
meco	optional	box	≤1
ipmc	optional	box	≤1

Table A.48 - Content element name for jp2x

content	length	type	loop
codestream	variable	hexbyte	

Table A.49 - Content element name for fiel

content	length	type	loop
field_count	1	integer	
field_order	1	integer	

Table A.50 - Content element name for orfo

content	length	type	loop
original_field_count	1	integer	
original_field_order	1	integer	

Table A.51 - Content element name for jsub

content	length	type	loop
hori_sub	1	integer	
vert_sub	1	integer	
hori_offset	1	integer	
vert_offset	1	integer	

Table A.52 - Content element name for raw_ & twos

content	length	type	loop
reserved	6	integer	
data_reference	2	integer	
reserved	8	integer	
channel_count	2	integer	
sample_size	2	integer	
predefined	2	integer	
reserved	2	integer	
sample_rate	4	integer	

Table A.56 - Content element name for mvhd

content	length	type	loop
version	1	integer	
flag	3	integer	
creation_time	4, 8	integer	
modification_time	4, 8	integer	
time_scale	4	integer	
duration	4, 8	integer	
rate	4	integer	
volume	2	integer	
reserved	10	integer	
matrix	36	hexbyte	
predefined	24	integer	
next_track_id	4	integer	

Table A.57 - Content element name for free & skip

content	length	type	loop
data	variable	hexbyte	

Table A.58 - Content element name for trak

content	length	type	loop
tkhd	-	box	=1
mdia	-	box	=1
tref	optional	box	≤1
edts	optional	box	≤1
udta	optional	box	≤1
meta	optional	box	≤1
meco	optional	box	≤1

Table A.59 - Content element name for tkhd

content	length	type	loop
version	1	integer	
flag	3	integer	
creation_time	4, 8	integer	
modification_time	4, 8	integer	
track_id	4	integer	
reserved	4	integer	
duration	4, 8	time	
reserved	8	integer	
layer	2	integer	
alternate_group	2	integer	
volume	2	integer	
reserved	2	integer	
matrix	36	hexbyte	
width	4	integer	
height	4	integer	

Table A.60 - Content element name for matrix

content	length	type	loop
matrix			
cell	4	integer	

Table A.61 - Content element name for tref

content	length	type	loop
hint cdsc hind	-	hexbyte	

Table A.62 - Content element name for hint, cdsc, & hind

content	length	type	loop
hint cdsc hind			
track_id	4	integer	1

Table A.63 - Content element name for mdia

content	length	type	loop
mdhd	-	box	=1
hdlr	-	box	=1
minf	-	box	=1

Table A.64 - Content element name for mdhd

content	length	type	loop
version	1	integer	
flag	3	integer	
creation_time	4, 8	integer	
modification_time	4, 8	integer	
time_scale	4	integer	
duration	4, 8	integer	
language	2	integer	
predefined	2	integer	

Table A.65 - Content element name for hdlr

content	length	type	loop
version	1	integer	
flag	3	integer	
predefined	4	integer	
type	4	integer 'vide' 'soun' 'hint' 'meta'	
reserved	12	integer	
name	variable	string	

Table A.66 - Content element name for minf

content	length	type	loop
vmhd smhd hmhd nmhd	-	box	=1
stbl	-	box	=1
dinf	-	box	=1

Table A.67 - Content element name for vmhd

content	length	type	loop
version	1	integer	
flag	3	integer	
graphic_mode	2	integer	
op_code	2	integer	3

Table A.68 - Content element name for hmhd

content	length	type	loop
version	1	integer	
flag	3	integer	
max_pdu_size	2	integer	
avg_pdu_size	2	integer	
max_bitrate	4	integer	
avg_bitrate	4	integer	
reserved	4	integer	

Table A.69 - Content element name for smhd

content	length	type	loop
version	1	integer	
flag	3	integer	
balance	2	integer	
reserved	2	integer	

Table A.70 - Content element name for stbl

content	length	type	loop
stsd	-	box	=1
stdp	optional	box	≤1
stts	-	box	=1
ctts	optional	box	≤1
stss	optional	box	≤1
stsh	optional	box	≤1
sdtp	optional	box	≤1
stsz stz2	-	box	=1
stsc	-	box	=1
stco co64	-	box	=1
padb	optional	box	≤1
subs	optional	box	≤1
sbgp	optional	box	≥0
sgpd	optional	box	≥0

Table A.71 - Content element name for btrt

content	length	type	loop
buffer_size	4	integer	
max_bitrate	4	integer	
avg_bitrate	4	integer	

Table A.72 - Content element name for metx

content	length	type	loop
reserved	6	integer	
data_reference	4	integer	
content_encoding	optional	string	
namespace	variable	string	
location	optional	string	
btrt	optional	box	≥0

Table A.73 - Content element name for mett

content	length	type	loop
reserved	6	integer	
data_reference	4	integer	
content_encoding	optional	string	
mine_format	variable	string	
btrt	optional	box	≥0

Table A.74 - Content element name for pasp

content	length	type	loop
hori_spacing	4	integer	
vert_spacing	4	integer	

Table A.75 - Content element name for clap

content	length	type	loop
width_num	4	integer	
width_den	4	integer	
height_num	4	integer	
height_den	4	integer	
hori_offset_num	4	integer	
hori_offset_den	4	integer	
vert_offset_num	4	integer	
vert_offset_den	4	integer	

Table A.76 - Content element name for Video coding

content	length	type	loop
reserved	6	integer	
data_reference	2	integer	
predefined	2	integer	
reserved	2	integer	
predefined	12	integer	
width	2	integer	
height	2	integer	
hori_resolution	4	integer	
vert_resolution	4	integer	
reserved	4	integer	
predefined	2	integer	
name	32	string	
depth	2	integer	
predefined	2	integer	
clap	optional	box	≤1
pasp	optional	box	≤1

Table A.77 - Content element name for Audio coding

content	length	type	loop
reserved	6	integer	
data_reference	2	integer	
reserved	8	integer	
channel_count	2	integer	
sample_size	2	integer	
predefined	2	integer	
reserved	2	integer	
sample_rate	4	integer	

Table A.78 - Content element name for stsd

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
mjp2 raw_ twos	-	box	⌋

Table A.79 - Content element name for stdp

content	length	type	loop
version	1	integer	
flag	3	integer	
priority	2	integer	⌋

Table A.80 - Content element name for stsl

content	length	type	loop
version	1	integer	
flag	3	integer	
constant_flag	1	integer	
scale_method	1	integer	
display_center_x	2	integer	
display_center_y	2	integer	

Table A.81 - Content element name for stts

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
sample_count	4	integer	⌋
sample_delta	4	integer	

Table A.82 - Content element name for ctts

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
sample_count	4	integer	⌋
sample_offset	4	integer	

Table A.83 - Content element name for stss

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
sample_number	4	box	⌋

Table A.84 - Content element name for stsh

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
shadowed_sample_number	4	integer	]
sync_sample_number	4	integer	

Table A.85 - Content element name for sdtp

content	length	type	loop
version	1	integer	
flag	3	integer	
reserved	2	integer	]
sample_depends_on	2	integer	
sample_is_depends_on	2	integer	
sample_has_redundancy	2	integer	

Table A.86 - Content element name for edts

content	length	type	loop
elst	optional	box	≤1

Table A.87 - Content element name for elst

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
segment_duration	4, 8	integer	]
media_time	4, 8	integer	
media_rate_integer	2	integer	
media_rate_fraction	2	integer	

Table A.88 - Content element name for dinf

content	length	type	loop
dref url_ urn_	-	box	=1

Table A.89 - Content element name for url_

content	length	type	loop
version	1	integer	
flag	3	integer	
location	variable	string	

Table A.90 - Content element name for urn_

content	length	type	loop
version	1	integer	
flag	3	integer	
name	variable	string	
location	variable	string	

Table A.91 - Content element name for dref

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
url_ urn_	-	box	]

Table A.92 - Content element name for stsz

content	length	type	loop
version	1	integer	
flag	3	integer	
sample_size	4	integer	
num_entries	4	integer	
entry_size	0, 4	integer	]

Table A.93 - Content element name for stz2

content	length	type	loop
version	1	integer	
flag	3	integer	
reserved	3	integer	
field_size	1	integer	
num_entries	4	integer	
entry_size	filed_size	integer	]

Table A.94 - Content element name for stsc

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
first_chunk	4	integer	]
sample_per_chunk	4	integer	
description_index	4	integer	

Table A.95 - Content element name for stco

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
chunk_offset	4	integer	]

Table A.96 - Content element name for co64

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
chunk_offset	8	integer	]

Table A.97 - Content element name for padb

content	length	type	loop
version	1	integer	
flag	3	integer	
pad1	4	integer	]
pad2	8	integer	

Table A.98 - Content element name for subs

content	length	type	loop
version	1	integer	
flag	3	integer	
num_entries	4	integer	
sample_delta	4	integer	]
num_subsample	2	integer	
subsample_size	4, 8	integer	
subsample_priority	1	integer	
discardable	1	integer	
reserved	4	integer	

Table A.99 - Content element name for mvex

content	length	type	loop
mehd	optional	box	≤1
trax	-	box	=1

Table A.100 - Content element name for mehd

content	length	type	loop
version	1	integer	
flag	3	integer	
fragment_duration	4, 8	integer	