INTERNATIONAL STANDARD

ISO/IEC 15444-14

First edition
2013-07-15

Information technology — JPEG 2000 image coding system —

Part 14: XML representation and reference

Technologies de l'information — Système de codage d'images JPEG 2000 —
Partie 14: Représentation et référence XML

Reference number
ISO/IEC 15444-14:2013(E)

COPYRIGHT PROTECTED DOCUMENT

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

Published in Switzerland

CONTENTS

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.

ISO/IEC 15444-14 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.813 (2012).

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

— *Part 1: Core coding system*

— *Part 2: Extensions*

— *Part 3: Motion JPEG 2000*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

— *Part 6: Compound image file format*

— *Part 8: Secure JPEG 2000*

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

— *Part 10: Extensions for three-dimensional data*

— *Part 11: Wireless*

— *Part 12: ISO base media file format*

— *Part 13: An entry level JPEG 2000 encoder*

— *Part 14: XML representation and reference*

INTERNATIONAL STANDARD

RECOMMENDATION ITU-T

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

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 the codestream, and a referring method for embedding 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 exact box or codestream data in an image;
- provides guidance on processes for converting 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 (2002) | ISO/IEC 15444-1:2004, *Information technology – JPEG 2000 image coding system: Core coding system.*
- Recommendation ITU-T T.801 (2002) | ISO/IEC 15444-2:2004, *Information technology – JPEG 2000 image coding system: Extensions.*
- Recommendation ITU-T T.802 (2005) | ISO/IEC 15444-3:2007, *Information technology – JPEG 2000 image coding system: Motion JPEG 2000.*
- Recommendation ITU-T T.805 (2012) | ISO/IEC 15444-6 (2013), *Information technology – JPEG 2000 image coding system: Compound image file format.*
- Recommendation ITU-T T.807 (2006) | ISO/IEC 15444-8:2007, *Information technology – JPEG 2000 image coding system: Secure JPEG 2000.*
- Recommendation ITU-T T.808 (2005) | ISO/IEC 15444-9:2005, *Information technology – JPEG 2000 image coding system: Interactivity tools, APIs and protocols.*
- Recommendation ITU-T T.809 (2011) | ISO/IEC 15444-10:2011, *Information technology – JPEG 2000 image coding system: Extensions for three-dimensional data.*
- Recommendation ITU-T T.810 (2006) | ISO/IEC 15444-11:2007, *Information technology – JPEG 2000 image coding system: Wireless.*
- Recommendation ITU-T T.812 (2007) | ISO/IEC 15444-13:2008, *Information technology – JPEG 2000 image coding system: Wireless.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- Recommendation ITU-T T.832 (2009), *Information technology – JPEG XR image coding system: An entry level JPEG 2000 encoder.*

  ISO/IEC 29199-2:2009, *Information technology – JPEG XR image coding system – Part 2: Image coding specification.*

- Recommendation ITU-T T.833 (2010), *Information technology – JPEG XR image coding system – Motion JPEG XR*

  ISO/IEC 29199-3:2010, *Information technology – JPEG XR image coding system – Part 3: Motion JPEG XR.*
- ISO 12639:1998, Graphic technology – *Prepress digital data exchange – Tag image file format for image technology (TIFF/IT).*

2.3 Additional references

- ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange.*
- ISO/IEC 15444-12:2012, *Information technology – JPEG 2000 image coding system – Part 12: ISO base media file format.*
- IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One.*
- IETF RFC 2279 (1998), *UTF-8, A transformation format of ISO 10646.*
- IETF RFC 4648 (2006), *The Base16, Base32, and Base64 Data Encodings*.
- W3C Recommendation (2009), *Namespaces in XML 1.0 (Third Edition).*
- W3C Recommendation (2008), *Extensible Markup Language (XML), Version 1.0 (Fifth Edition).*
- W3C Recommendation (2004), *XML Schema Part 0: Primer.*
- W3C Recommendation (2004), *XML Schema Part 1: Structures.*
- W3C Recommendation (2004), *XML Schema Part 2: Datatypes.*
- W3C Recommendation (2010), *XML Path Language (XPath) 2.0.*

3 Definitions

For the purposes of this Recommendation | International Standard, the definitions given in Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2 and those listed below apply. Should there be any difference between the definition given in this clause and the one given in one of the other Recommendation | International Standard cited above, the one given in this clause prevails.

3.1 **…**: elision mark. This mark denotes that 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 a box type that specifies its contents.

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

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

3.5 **box-based format**: A sequence of boxes that contains several image properties and expresses an image file format. This image format starts with a signature box and contains at least one codestream.

3.6 **box element**: A JPXML element for a box, and this element name is translated from the 4CC of the box type by using conversion rules described in clause 7.

3.7 **codestream**: A sequence of bits contained in a sequence of *bytes*, created by an image coder. This data sequence contains marker segments for holding image coding properties that are parsed by a decoder or translator. This may be arranged so that 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 **fat representation**: A JPXML document that contains whole image data on text nodes. This representation can be translated to a genuine image without any additional image information because it contains whole image information. However, because of translating whole chunk data into the XML format, this representation needs more data space than was required for the original image. For more details, see 6.1.

3.9 **fat-skeleton representation**: A JPXML document that contains image properties excluding codestream chunk data. This representation may have a location path to chunk image data by using the JPXML format. This represents a box file format image structure. For more details, see 6.1.

3.10 JPXML converter: A converter that translates data between an image and a JPXML document. The JPXML "forward" converter translates an image to a JPXML document, and the JPXML "inverse" converter translates the edited document and codestream data to an image. These converters use rules of element name creation, defined container and undefined chunk container conversions.

3.11 JPXML document: An XML document that corresponds to the box file format or codestream, categorized according to included contents; skeleton, fat-skeleton and fat representations. For more details, see 6.1.

3.12 JPXML document structure: The structure of a JPXML element in a JPXML document, which expresses an image data structure. This structure is used for a location path using XPath expression.

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

3.14 location path: The location of an internal image data using XPath expression with a JPXML document. This expression represents the absolute offset value and the JPXML document structure.

3.15 marker element: A JPXML element for the marker segment. This element name is translated from the marker type using the conversion rule described in 7.1 and 7.3.

3.16 marker segment: A binary data block in a codestream which contains the marker type and may contain marker properties for coding information.

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

4 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, Rec. 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.

4.1 Abbreviations

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

JPXML	Refers to this Recommendation	International Standard
MIME	Multipurpose Internet Mail Extension
TIFF	Tag Image File Format
XML	eXtended Metadata Language
XSLT	XML Stylesheet Language Transformation

5 Conventions

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.

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 an intermediate image description for accessing internal data robustly and converting an image type. The following subclauses describe more details of this document.

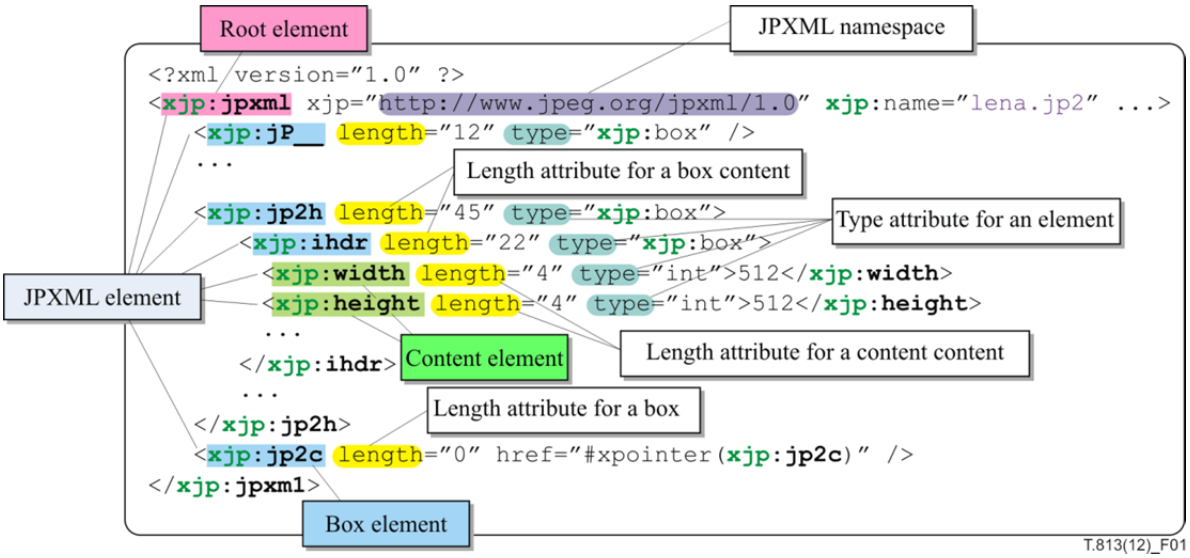


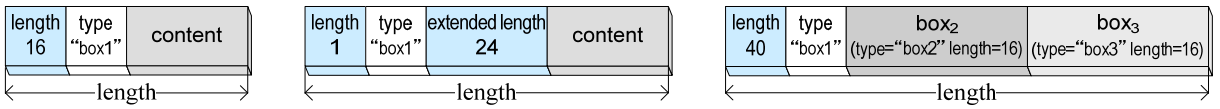
Figure 1 – Example of a JPXML document for a JPEG 2000 image

6.1 Structure of the 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 a container's property or a box content. Some containers, such as a superbox, contain 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 the byte length and data type of each data chunk, respectively. The content value may be described with text, and its data type denoted with 'type' attribute. Figure 1 is an example of a JPXML document for a JPEG 2000 file format.

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

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



(a) box (b) box with extended length (c) superbox

```
<box1 length="16" type="box">
  <content length="8" />
</box1>
```

```
<box1 length="1" type="box">
  <length length="8" type="int">
    24</length>
  <content length="8" />
</box1>
```

```
<box1 length="40" type="box">
  <box2 length="16" type="box" />
  <box3 length="16" type="box" />
</box1>
```

(d) JPXML document for (a) (e) JPXML document for (b) (f) JPXML document for (c)

Figure 2 – Examples of box format and JPXML documents

The JPXML document is generated from an image file format and/or codestreams, and its kind varies from none property to including codestream data representations. When kinds of image property representation are included, the JPXML document is categorized with three levels of representation: "skeleton", "fat-skeleton", and "fat" representations.

The first-level representation, the skeleton representation, shall express only the structure of the 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 the JPXML elements. This representation is used for a location path that is comparatively robust for changing the box structure of the image and/or marker segment structure of the codestream.

The second-level representation, the fat-skeleton representation, 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 the 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 third-and final level representation, the fat representation, expresses the image structure and whole image property values. This whole property may represent a binarized format for use of some applications, such as secure purpose. The binarized contents are translated with MIME's base64 encoding. As this representation requires more data space than the original image data, it is unsuited for use in a storage file format for image data.

6.2 Creation of a JPXML document

A 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 includes forward and inverse JPXML generators. This example consists of a forward JPXML generator, an image edit tool, and an inverse JPXML generator. The processes of forward and inverse JPXML generators 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: the common conversion rule and the 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 the 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. Not all 4CC values are allowable for an XML element name, such as an 'xml' and space character, and these 4CC values are modified for an XML element by using the conversion rules defined in 7.2. The details of the 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, Rec. ITU-T T.833 | ISO/IEC 29199-3 and ISO/IEC 15444-12. The element names are defined in 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 a visible code value for use as an XML element name by using conversion rules defined in 7.3. The details of these element names are defined in Annex B.

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

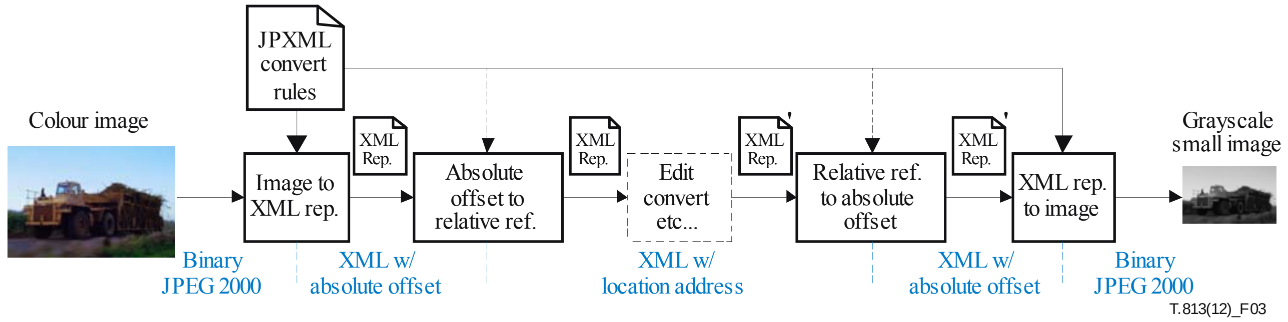


Figure 3 – Block diagram of image converting with JPXML

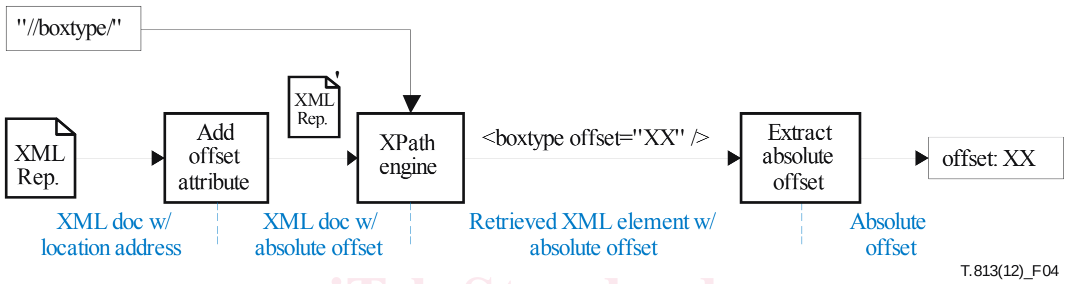


Figure 4 – Example of the process for converting between the XML location path and offset

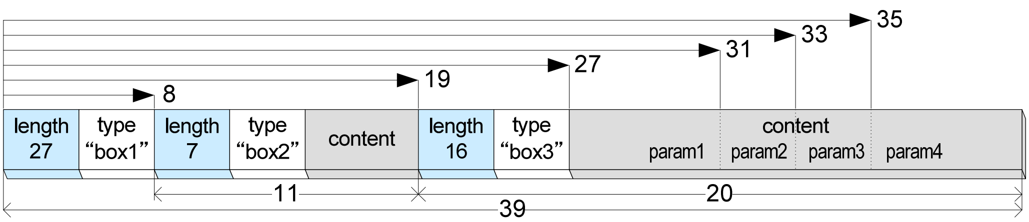
6.3 Access with the JPXML document

The two location representations, an XML location path and an absolute offset, are used to access internal image data with the JPXML document. The XML location path homologizes the target image data to 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 the 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 in 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 depicts an example of this process. After this process, the generated absolute offset is the location of the target data chunk from the very start of an image.

A target location path can be generated from a target absolute offset by 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 generators use conversion rules defined in 8.1. Figure 5 is an example of a 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 of data space for the box length and type of storage, and the "box2" is the first child of "box1" box. By using this generated JPXML document, all image data chunks described by the JPXML elements can be accessed by the XML location path.



(a) An example of part of a pseudo box-based format structure

```
 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 an 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 – Example of JPXML document accessing

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 the following common forward rules for JPXML conversion. The boxes within a superbox 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 by using 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, representing one file format or codestream.
3) The root element may have a 'name' attribute for identifying an image name.
4) All elements shall have 'length' and 'type' attributes, representing 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, representing the data length in byte.
7) All element may have an 'offset' attribute, representing the absolute offset to the data chunk in byte.
8) The box element shall have several box elements when the original box has a box inside its content.
9) The element of a 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 names.
11) The box or marker parameters may be represented in 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 converters shall use the following common inverse rules for JPXML conversion. A box element having one or more box elements shall be converted to a superbox 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 a 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 into internal contents of the data chunk.

10) All content element 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 in one of the parent contents of the box or maker.

12) The JPXML content in base64 binary shall be converted to a binary data with the base64 converter and stored in one of the 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, and shall use the following forward conversion rule for the 4CC:

1) The JPXML element name shall use a 4CC box type.

2) The alphanumeric characters 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, and shall use the following inverse conversion rule for the 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 alphanumeric name shall be directly used for the 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 a 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 and shall use the 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 at the front of the four character string, and creates a 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, and shall use the following inverse conversion rule for the 4CC.

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

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

7.4 Element name rule for a marker segment

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

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

The JPXML inverse converter to marker segments translates a JPXML document to JPEG 2000 marker segments, and shall use the 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 with a name starting with a '_' character of the five-character string shall be converted to the marker symbol code by using the inverse conversion rule for the tagged image file format.

Table 2 – Example of codestream marker conversions

Codestream marker name	Symbol name	Code	JPXML element name
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 identifies the data type of the element content data. This Recommendation | International Standard defines uses of six data types: box, marker, fourcc, location, hexbyte, and integer and time. The integer and time data types are used for the element content data type as integer and time types, respectively, and these data types are defined in the XML schema Part 2 Recommendation. The box and marker data types are used to identify the box element and the marker element, respectively. The fourcc data type indicates the element content data that shall be the 4CC data type, and the 4CC value will not converted with the rules previously described. The location indicates that the elements can have any location type value. The hexbyte indicates that the element data may be represented as a hexadecimal value in big-endian order. The XML definitions for the box, marker, fourcc, location and hexbyte data types can be defined by the following XML schema's simple type data type 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 Accessing image data

The XML location path and the JPXML document can be used to access internal binary chunk data in the image data. The following three steps access the internal binary data: the first step identifies the target element by using the location path and the JPXML document, the second step converts the target element to an absolute offset, and the third and final step accesses the internal image data with the target absolute offset position. To achieve the second step, the following rules are employed.

8.1 Rules 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 comply with these rules, there may be two approaches for calculation of the target absolute offset value; the target to root approach and the root to target approach.

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

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

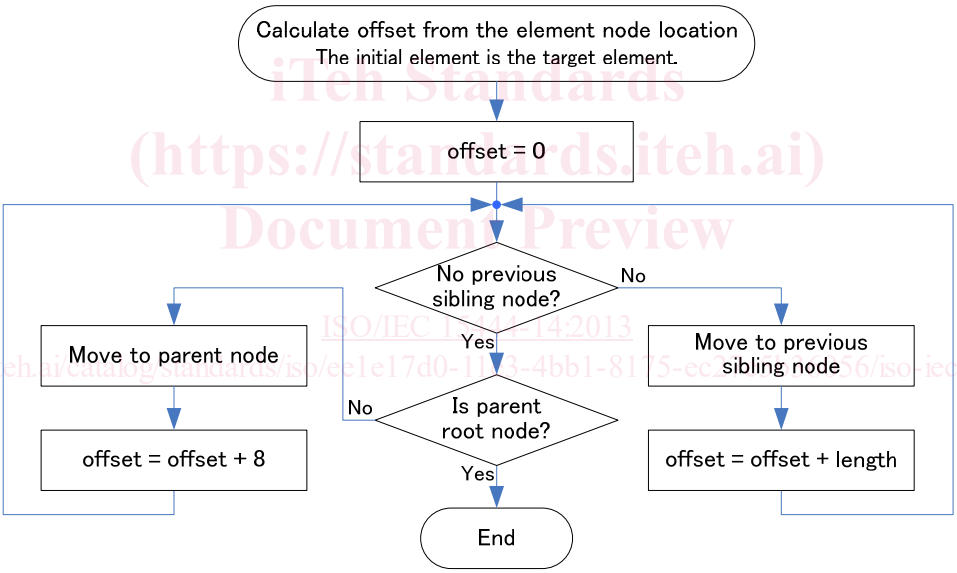


Figure 6 – Example 1 of an algorithm for calculating an offset value from the target element