INTERNATIONAL STANDARD

ISO/IEC 21122-3

First edition
2019-08

Information technology — JPEG XS low-latency lightweight image coding system —

Part 3: Transport and container formats

Reference number
ISO/IEC 21122-3:2019(E)

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

Published in Switzerland

Contents

Page

Foreword

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

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 21122 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document is part of a series of standards for a low-latency lightweight image coding system, denoted JPEG XS.

In many use cases during production or transmission of a movie, limiting the latency and the recompression loss is a more important aspect than the compression efficiency. The JPEG XS coding system offers compression and recompression of image sequences with very moderate computational resources while remaining robust under multiple compression and decompression cycles and mixing of content sources, e.g. embedding of subtitles, overlays or logos. Typical target compression ratios ensuring visually lossless quality are in the range of 2:1 to 10:1, depending on the nature of the source material. The end-to-end latency can be confined to a fraction of a frame, typically between a small number of lines down to below a single line.

This document specifies transport and container formats for JPEG XS codestreams. It also defines metadata that enriches transport protocols for transmission of image sequences, in order to facilitate transport, editing and presentation.

Information technology — JPEG XS low-latency lightweight image coding system —

Part 3: Transport and container formats

1 Scope

This document defines transport and container formats for JPEG XS codestreams as specified in ISO/IEC 21122-1. It defines file formats for working with still image and motion image sequence files on computer platforms and gives guidance on how to embed the codestream in transport streams, allowing internet-based communication.

This document uses already existing specifications for file formats and extends them for the embedding of JPEG XS codestreams.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

ISO/IEC 11578, *Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)*

ISO/IEC 11664-1, *Colorimetry — Part 1: CIE standard colorimetric observers*

ISO/IEC 14496-12, *Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 15076-1, *Image technology colour management — Architecture, profile format and data structure — Part 1: Based on ICC.1: 2010*

ISO/IEC 21122-1:2019, *JPEG XS low-latency lightweight image coding system — Part 1: Core coding system*

ISO/IEC 21122-2:2019, *JPEG XS low-latency lightweight image coding system — Part 2: Profiles and buffer models*

ISO/IEC 23008-12:2017, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format*

Rec. ITU-T H.273, *Coding-independent code points for video signal type identification*

JEITA CP-3451D, *Exchangeable image file format for digital still cameras: Exif Version 2.31*

ANSI/CTA 861-G:2016, *A DTV Profile for Uncompressed High Speed Digital Interfaces*

W3C REC-xml-20081126, *Extensible Markup Language (XML) 1.0* (Fifth Edition), W3C Recommendation

3 Terms and definitions

For the purposes of this document the terms, definitions, and abbreviated terms given in ISO/IEC 14496-12, ISO/IEC 21122-1, ISO/IEC 21122-2, ISO/IEC 23008-12 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
aux
auxiliary component channel typically used as opacity channel or alpha mask

3.2
big-endian
in order from the most significant to the least significant bits of a value representation

3.3
box
structured collection of data describing the image or the image decoding process

3.4
box content
data wrapped within the *box* (3.3) structure

3.5
box type
kind of information stored with the *box* (3.3)

3.6
byte
group of 8 bits

3.7
coding-independent code point
CICP
code point based on enumerated values for the definition of the colourspaces

Note 1 to entry: Code points defined in Rec. ITU-T H.273.

3.8
HEIF
high efficiency image file format
image file format which can embed still images and *motion sequences* (3.10)

Note 1 to entry: Based on ISO/IEC 23008-12.

3.9
JXS
still image file format with JPEG XS compressed images

3.10
motion sequence
movie
timed sequence of images

3.11
sample
<coding> single element in the two-dimensional image array which comprises a component

Note 1 to entry: This definition is used in Annex A.

[SOURCE: ISO/IEC 21122-1:2019, 3.1.42, modified – The domain and Note 1 to entry have been added.]

3.12
sample
<ISOBMFF> all the data associated with a single time

Note 1 to entry: This definition is used in Annexes B and C as data associated with one coded image in a sequence.

3.13
superbox
box (3.3) that carries other boxes as payload data

3.14
UTF-8
variable size character encoding

Note 1 to entry: The encoding is defined in ISO/IEC 10646.

4 Symbols and abbreviated terms

4.1 Symbols

N_c	number of components in an image
Plev	level a particular codestream conforms to
Ppih	profile a particular codestream conforms to
`Picture()`	JPEG XS codestream as defined in ISO/IEC 21122-1
`Codestream_Header()`	codestream header preceding the image data in the codestream as defined in A.5.5
`Codestream_Body()`	coded image data in the codestream without `Codestream_Header()` as defined in A.5.5

4.2 Abbreviated terms

CIE	Commision Internationale de l'Eclairage
ISOBMFF	iso base media file format
LSB	least significant bit
MSB	most significant bit

4.3 Naming conventions for numerical values

Integer numbers are expressed as bit patterns, hexadecimal values, or decimal numbers. Bit patterns and hexadecimal values have both a numerical value and an associated particular length in bits.

Hexadecimal notation, indicated by prefixing the hexadecimal number by "0x", may be used instead of binary notation to denote a bit pattern having a length that is an integer multiple of 4. For example,

0x41 represents an eight-bit pattern having only its second most significant bit and its least significant bit equal to 1. Numerical values that are specified under a "**Code**" heading in tables that are referred to as "code tables" are bit pattern values (specified as a string of digits equal to 0 or 1 in which the left-most bit is considered the most-significant bit). Other numerical values not prefixed by "0x" are decimal values. When used in expressions, a hexadecimal value is interpreted as having a value equal to the value of the corresponding bit pattern evaluated as a binary representation of an unsigned integer (i.e., as the value of the number formed by prefixing the bit pattern with a sign bit equal to 0 and interpreting the result as a two's complement representation of an integer value). For example, the hexadecimal value 0xF is equivalent to the 4-bit pattern '1111' and is interpreted in expressions as being equal to the decimal number 15.

5 Conformance

This document shares common definitions for the structure of files (a sequence of objects, called boxes here, and atoms in other similar file formats), and a common definition of the general structure of an object (the size and type).

File formats representing either images, or image sequences shall be as specified in Annexes A, B and C. All these specifications require that readers ignore objects that are unrecognizable to them.

This document takes precedence over those on which it is based, in any case where there are differences or conflicts; however, no such conflicts are known to exist.

For better readability and understanding, the syntax description for the different file formats is done in the same way as in the base formats.

6 Colour specification

JPEG XS (as defined in ISO/IEC 21122-1) describes only the encoded bitstream of an image. The integrated multiple component transformation is only responsible for a decorrelation of the different colour components allowing for the reduction of the entropy in the data. In order to properly display or interpret the image, it is essential that the colourspace of that image data is properly characterized. For this purpose, the respective container format or transport channel has to signal the correct colourspace. The defined formats in this document for JPEG XS signals the colour space as specified in Rec. ITU-T H.273.

7 Organization of the document

Annex A specifies the JXS file format for still images. It is based on the box-based format defined in the JPEG 2000 family of standards, for example JPEG 2000 (Rec. ITU-T T. 800 | ISO/IEC 15444-1). The boxes carry additional information supporting the use of the codestream as still image file format.

Annex B specifies the integration of JPEG XS codestreams in the ISOBMFF (as defined in ISO/IEC 14496-12) for use of image sequences as movie in a file format.

Annex C specifies the integration of JPEG XS codestreams in the HEIF file format (as defined in ISO/IEC 23008-12) allowing the integration of both still images as well as movies in one format.

Annex D specifies the Media Type registration for JPEG XS codestreams bare any file format container.

Annex A
(normative)

Use of JPEG XS codestreams in still image file format - JXS

A.1 General

This annex defines a still image file format that applications may choose to wrap a codestream based on a JPEG XS compressed image. While not all applications will use this format, many applications will find that it meets their needs. However, those applications that do implement this file format shall implement it as described in this entire annex. This specification is based on the same syntax as the box-based file format for JPEG 2000 in ISO/IEC 15444-1:2016, Annex I or ISO/IEC 15444-2:2004, Annex M.

This annex:

— specifies a binary container (file) for both image and metadata;

— specifies a mechanism to indicate image properties, such as the tonescale or colourspace of the image;

— specifies a mechanism by which readers may recognize the existence of intellectual property rights information in the file;

— specifies a mechanism by which metadata (including vendor-specific information) can be included in files specified by this document.

A.2 Specification of the JXS file format

A.2.1 General

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

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

A.2.2 File identification

JXS files can be identified using several mechanisms. When stored in traditional computer file systems, JXS files should be given the file extension ".jxs" (readers should allow mixed case for the alphabetic characters).

A.2.3 File organization

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

The binary structure of a box is identical to ISO/IEC 15444-1:2016 and defined in A.3.

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

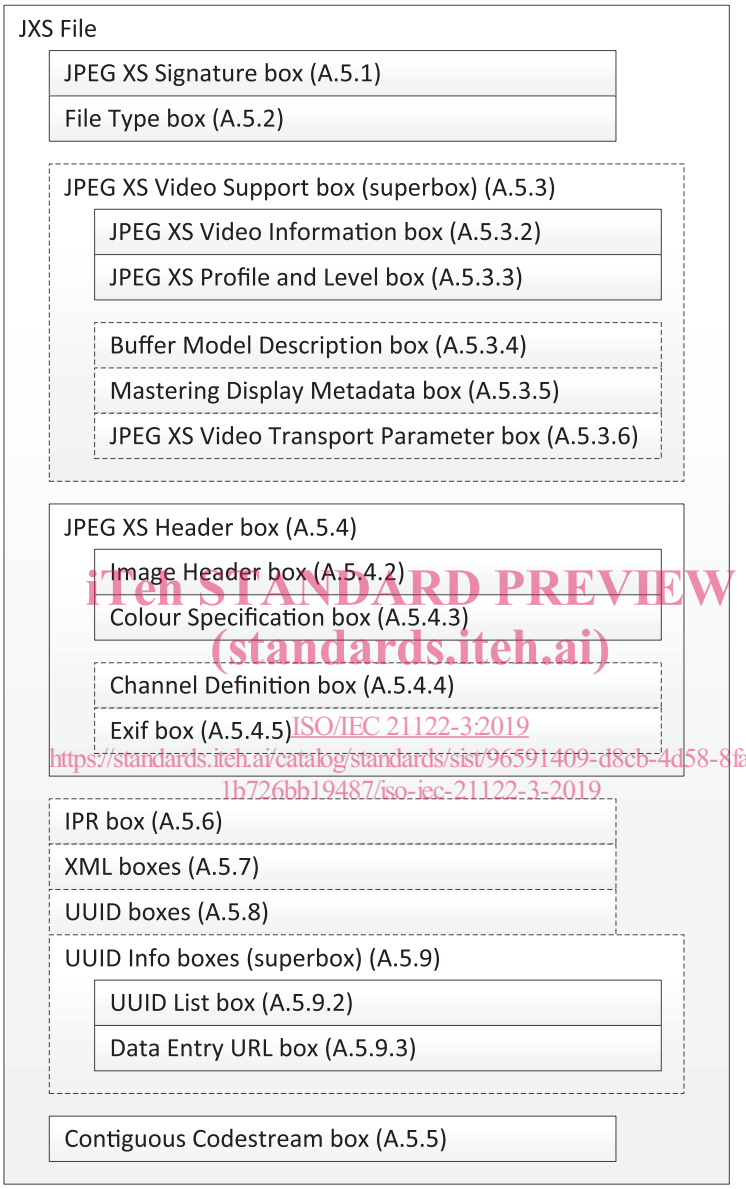

Figure A.1 — Conceptual structure of a JXS file

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

The file shown in Figure A.1 is a strict sequence of boxes. Other boxes may be found between the boxes defined in this document. However, all information contained within a JXS file shall be in the box format; byte-streams not in the box format shall not be found in the file.

As shown in Figure A.1, a JXS file contains a JPEG XS Signature box, JPEG XS Header box, and one or more Contiguous Codestream boxes. A JXS file may also contain other boxes as determined by the

file writer. For example, a JXS file may contain several XML boxes (containing metadata) between the JPEG XS Header box and the first Contiguous Codestream box.

A.2.4 Greyscale, colour, multi-component specification

One of the most important aspects of a file format is that it specifies the colourspace of the contained image data. In order to properly display or interpret the image data, it is essential that the colourspace of that image is properly characterized. The JXS file format provides one method to specify the colourspace of the image based on coding-independent code points (CICP). The CICP enumerated method specifies the colourspace of an image by the use of three numeric values that identifies the colourspace. The set of supported colourspaces is specified in A.5.4.3. The allowed values are a subset of the code points defined in Rec. ITU-T H.273.

A.2.5 Inclusion of auxilliary channels

In many applications, components other than the colour channels are required (Aux channel). For example, many images used on web pages contain opacity information; the browser uses this information to blend the image into the background. Another example is the use of alpha channels in video production; video mixers use this alpha channel to mix multiple images. It is thus desirable to include both the colour and auxiliary channels within a single codestream.

The JXS file format provides a means to indicate the presence of auxiliary channels (such as opacity), to define the type of these channels, and to specify the ordering and source of these. When a reader opens the JXS file, it determines the ordering and type of each component. The application shall then match the component definition and ordering from the JXS file with the component ordering as defined by the colourspace specification. Once the file components have been mapped to the colour channels, the decompressed image can be processed through any needed colourspace transformations.

How applications deal with opacity or other auxiliary channels is outside the scope of this document.

A.2.6 Metadata

One important aspect of the JXS file format is the ability to add metadata to a JXS file.

Some of the boxes provide a set of tools by which applications can add vendor-specific information to the JXS file format, like the EXIF box or the XML box. These boxes are optional in conforming files and may be ignored by conforming readers.

A.2.7 Conformance with the file format

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

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

Defining boxes for private implementation purposes is discouraged. Instead, implementation-specific metadata should be carried by UUID boxes as specified in A.5.8.

A.3 Concept of boxes

A.3.1 Key to graphical descriptions

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

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

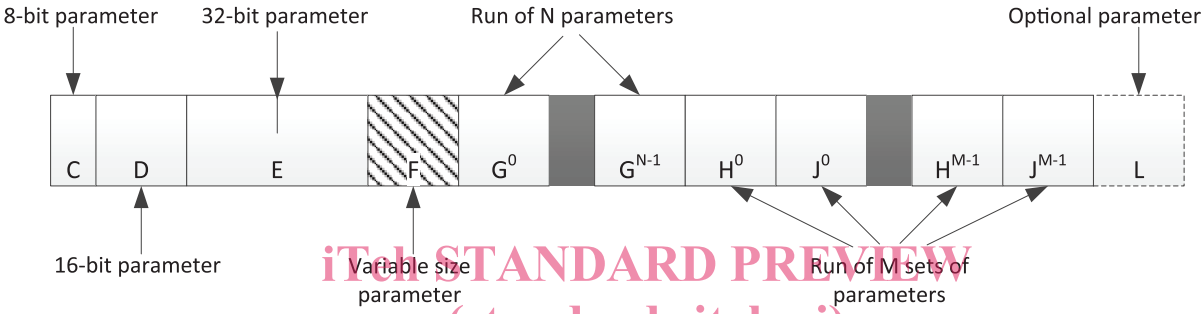

Figure A.2 — Example of the box description figures

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

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

Some boxes may carry other boxes as payload data. Such boxes are denoted as superboxes. The payload size of a superbox is given by the sum of the box lengths of all the boxes it contains.

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

For example, the superbox shown in Figure A.3 shall contain an AA box and a BB box, and the BB box shall follow the AA box. However, there may be other boxes found between boxes AA and BB. Dealing with unknown boxes is discussed in A.6.

Figure A.3 — Example of the superbox description figures

A.3.2 Box definition

Physically, each object in the file is encapsulated within a binary structure called a box. That binary structure is as in Figure A.4, and detailed in Table A.1.

Figure A.4 — Organization of a box

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

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

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

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

Table A.1 — Binary structure of a box

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

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