
**Information technology — High
efficiency coding and media delivery
in heterogeneous environments —**

**Part 12:
Image File Format**

*Technologies de l'information — Codage à haute efficacité et livraison
des médias dans des environnements hétérogènes —*

Partie 12: Format de fichier d'image



IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-12:2017



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Contents

	Page
Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	5
4 Overview	5
5 General requirements	7
5.1 General requirements on files	7
5.2 General requirements on readers	7
5.3 Multi-purpose files	7
5.4 Other boxes	7
6 Single image and image collection	7
6.1 General	7
6.2 Derivation from the ISO base media file format	8
6.3 Derivation of an output image of an image item	8
6.4 Roles of images	9
6.4.1 General	9
6.4.2 Hidden images	9
6.4.3 Cover image	9
6.4.4 Thumbnail images	9
6.4.5 Auxiliary images	9
6.4.6 Master images	9
6.4.7 Pre-derived coded images	10
6.4.8 Multi-layer images	10
6.5 Image properties	10
6.5.1 General	10
6.5.2 Decoder configuration and initialization	11
6.5.3 Image spatial extents	11
6.5.4 Pixel aspect ratio	12
6.5.5 Colour information	12
6.5.6 Pixel information	13
6.5.7 Relative location	13
6.5.8 Image properties for auxiliary images	14
6.5.9 Clean aperture	15
6.5.10 Image rotation	15
6.5.11 Layer selection	16
6.5.12 Image mirroring	16
6.6 Derived images and derived image items	17
6.6.1 General	17
6.6.2 Derived image types and derived image item types	17
6.7 Image metadata	19
6.8 Relating an untimed item to a timed sequence	20
6.8.1 'equiv' entity group	20
6.8.2 'equiv' sample group	20
7 Image sequences	21
7.1 General	21
7.2 Derivation from the ISO base media file format	21
7.2.1 Track Header box	21
7.2.2 Handler type	22
7.2.3 Coding Constraints box	22

7.3	Presentation of an image sequence track.....	23
7.4	Sample groups.....	23
7.4.1	Direct reference samples list.....	23
7.5	Other tracks.....	25
7.5.1	General.....	25
7.5.2	Thumbnail image sequence track.....	25
7.5.3	Auxiliary image sequence track.....	25
8	Metadata support.....	26
8.1	General.....	26
8.2	Metadata for image items.....	26
8.3	Metadata for image sequence tracks.....	26
8.4	Integrity checks.....	26
8.4.1	General.....	26
8.4.2	Syntax.....	27
8.4.3	Semantics.....	27
9	Extensions to the ISO base media file format.....	28
9.1	General.....	28
9.2	ItemInfoEntry.....	28
9.3	Item Properties Box.....	28
9.3.1	Definition.....	28
9.3.2	Syntax.....	29
9.3.3	Semantics.....	29
9.4	Entity grouping.....	30
9.4.1	General.....	30
9.4.2	Groups List box.....	30
9.4.3	Entity to Group box.....	30
9.5	Additional track references.....	31
9.6	Repeating edits.....	31
9.6.1	Definition.....	31
9.6.2	Syntax.....	32
9.6.3	Semantics.....	32
9.7	Sample-to-item sample grouping.....	32
9.7.1	Definition.....	32
9.7.2	Syntax.....	33
9.7.3	Semantics.....	33
10	Image File Format brands.....	33
10.1	General.....	33
10.2	Image and image collection brands.....	34
10.2.1	'mif1' structural brand.....	34
10.3	Image sequence brands.....	35
10.3.1	'msf1' structural brand.....	35
	Annex A (normative) Storage of externally specified metadata.....	37
	Annex B (normative) HEVC Image File Format.....	39
	Annex C (normative) High efficiency image file MIME type registration.....	51
	Annex D (normative) High efficiency image sequence file MIME type registration.....	54
	Annex E (normative) AVC in the Image File Format.....	56
	Annex F (normative) Advanced coding image MIME type registration.....	61
	Annex G (normative) Advanced coding sequence MIME type registration.....	63
	Annex H (normative) JPEG in the Image File Format.....	64
	Annex I (informative) Guidelines for specifying storage of image coding formats.....	67
	Annex J (informative) Examples of image collections.....	68

Annex K (informative) Guidelines for progressive refinement	70
Bibliography	72

IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-12:2017

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.

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 ISO documents 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 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).

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

For an explanation on 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 the following URL: www.iso.org/iso/foreword.html.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology, SC 29, Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23008 series can be found on the ISO website.

Introduction

The Image File Format is designed to enable the interchange of images and image sequences, as well as their associated metadata. It forms part of a family of specifications that are box-structured and is built using tools defined in the ISO base media file format. This document specifies both structural brands that can be used with any codec and brands specific to High Efficiency Video Coding (HEVC). The file format specified in this document is referred to as the High Efficiency Image File Format (HEIF). When the requirements of the HEVC-specific brands are obeyed, the file format can be referred to as the HEVC Image File Format.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-12:2017

IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-12:2017

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 12: Image File Format

1 Scope

The formats defined in this document enable the interchange, editing, and display of images, as well as the carriage of metadata associated with those images.

The Image File Format builds on tools defined in ISO/IEC 14496-12 to define an interoperable storage format for a single image, a collection of images, and sequences of images.

This document specifies brands for the storage of images and image sequences conforming to High Efficiency Video Coding (HEVC).

NOTE The storage of HEVC video sequences is out of scope and is handled by ISO/IEC 14496-15.

This format defines normative structures used to contain metadata, how to link that metadata to the images, and defines how metadata of certain forms is carried.

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 10918-1:1994, *Information technology Digital compression and coding of continuous-tone still images — Part 1: Requirements and guidelines*

ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*

ISO/IEC 14496-12:2015, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 14496-15, *Information technology — Coding of audio-visual objects — Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format*

ISO/IEC 23008-2:2015, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

For the purposes of [Annex B](#) and [Annex E](#), the terms, definitions, and abbreviated terms specified in ISO/IEC 14496-15 apply.

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

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1

alternate group

group of *entities* ([3.1.10](#)) that are alternatives to each other and out of which only one should be selected for processing

3.1.2

associated image item

image item ([3.1.17](#)) that is associated with the *item property* ([3.1.26](#)) through the `ItemPropertiesBox`

3.1.3

auxiliary image

image ([3.1.15](#)) that may not be intended to be displayed but provides supplemental information, such as transparency data, complementing a respective *master image* ([3.1.27](#))

3.1.4

coded image

coded representation of an *image* ([3.1.15](#))

3.1.5

coded image item

item ([3.1.25](#)) whose data is a *coded image* ([3.1.4](#))

3.1.6

crop-rotate-mirror derived image item

derived image item ([3.1.8](#)) of type 'iden' that is not associated with any other types of *essential item properties* ([3.1.11](#)) than 'irot', 'clap', and 'imir'

3.1.7

derived image

representation of an *image* ([3.1.15](#)) as an *operation* ([3.1.31](#)) on other images

3.1.8

derived image item

item ([3.1.25](#)) whose data is a *derived image* ([3.1.7](#))

3.1.9

descriptive item property

item property ([3.1.26](#)) that describes rather than transforms the associated item

3.1.10

entity

item or track

3.1.11

essential item property

item property ([3.1.26](#)) that readers are required to process

3.1.12

HEVC image item

image item ([3.1.17](#)) of type 'hvc1' or 'lhv1'

3.1.13

hidden image

image ([3.1.15](#)) that is not intended to be displayed

3.1.14**hidden sample**

sample that is not intended to be displayed

3.1.15**image**

one or more arrays of pixels of different colour components described by an *image item* (3.1.17) or a sample

3.1.16**image collection**

set of *images* (3.1.15) stored as *items* (3.1.25) of a single file according to this specification

3.1.17**image item**

coded image item (3.1.5) or *derived image item* (3.1.8)

3.1.18**image property**

item property (3.1.26) for an *image item* (3.1.17)

3.1.19**image sequence**

sequence of *coded images* (3.1.4) which may be associated with advisory timing and in which images may use *inter prediction* (3.1.22)

3.1.20**image sequence track**

track that contains an *image sequence* (3.1.19)

3.1.21**input image**

image (3.1.15) that is used as an input for the *operation* (3.1.31) of the *derived image item* (3.1.8)

3.1.22**inter prediction**

prediction derived in a manner that is dependent on data elements (e.g. sample values or motion vectors) of *images* (3.1.15) other than the current image

3.1.23**intra coding**

coding of an *image* (3.1.15) that may use *intra prediction* (3.1.24) and does not use *inter prediction* (3.1.22)

3.1.24**intra prediction**

prediction derived from only data elements (e.g. sample values) of the same decoded image

3.1.25**item**

data that does not require timed processing, as opposed to sample data, and is described by the boxes contained in a *MetaBox*

3.1.26**item property**

descriptive or transformative information about an *item* (3.1.25) as stored in the item properties array

3.1.27**master image**

image that is stored as an *item* (3.1.25) and is not an *auxiliary image* (3.1.3) or a *thumbnail image* (3.1.37)

3.1.28

master image sequence

sequence of images that is stored as an *image sequence track* (3.1.20) and is not an *auxiliary image* (3.1.3) sequence or a thumbnail *image sequence* (3.1.19)

3.1.29

metadata item

item (3.1.25) containing metadata that may for example describe an *image item* (3.1.17)

Note 1 to entry: ISO/IEC 14496-12 uses the terms *item* and *metadata item* interchangeably to refer to an item of any type. This document overrides the *metadata item* definition of the ISO base media file format.

3.1.30

non-essential item property

item property (3.1.26) that readers are allowed to ignore

3.1.31

operation

for a *derived image item* (3.1.8), manipulation, identified by the item type, that produces a *reconstructed image* (3.1.34) from a set of *input images* (3.1.21)

3.1.32

output image

image (3.1.15) that results when the *reconstructed image* (3.1.34) of the *image item* (3.1.17) is transformed according to the *transformative item properties* (3.1.39) of the *image item* (3.1.17)

3.1.33

pre-derived coded image

coded image (3.1.4) that has been derived from one or more other images

3.1.34

reconstructed image

image (3.1.15) that results when the *coded image item* (3.1.5) is decoded or when the *operation* (3.1.31) of the *derived image item* (3.1.8), if any, is applied

3.1.35

reference image

image (3.1.15) that may be used as a reference for *inter prediction* (3.1.22) of another image

3.1.36

source image item

image item (3.1.17) referred to by the 'dimg' item reference from the *derived image item* (3.1.8) or from another derived image item that is a source image item for the derived image item

Note 1 to entry: In other words, an image item is a source image item for a derived image item when it is required for deriving the output image of the derived image item.

Note 2 to entry: The definition of the source image item is recursive: an image item is a source image item for a particular derived image item, when the output image of the image item is used as an input image for any derived image item in the 'dimg'-item-reference-linked chain of derived image items ending at that particular derived image item, inclusive.

3.1.37

thumbnail image

smaller-resolution representation of an *image* (3.1.15)

3.1.38

time-parallel sample

sample in the reference track that has the same or, when a sample with the same decoding time is not available, the closest preceding decoding time relative to that of the particular sample in the particular track

3.1.39**transformative item property**

item property (3.1.26) that transforms the reconstructed representation of the item content

Note 1 to entry: A transformative item property may, for example, specify rotation by 90°, 180°, or 270° of a reconstructed image of an image item.

3.2 Abbreviated terms

ASCII	American Standard Code for Information Interchange
AVC	Advanced Video Coding
DCF	Design rule for Camera File system
Exif	Exchangeable Image File Format
HDR	High Dynamic Range
HEIF	High Efficiency Image File Format
HEVC	High Efficiency Video Coding
MD5	Message Digest algorithm 5
MIME	Multi-purpose Internet Mail Extensions
NAL	Network Abstraction Layer
PPS	Picture Parameter Set
RBSP	Raw Byte Sequence Payload
SEI	Supplemental Enhancement Information
SPS	Sequence Parameter Set
TIFF	Tagged Image File Format
URN	Uniform Resource Name
UTF-8	Universal Character Set Transformation Format — 8-bit
VCL	Video Coding Layer
VPS	Video Parameter Set
XML	Extensible Markup Language
XMP	Extensible Metadata Platform

4 Overview

The Image File Format specifies the following two forms of storage:

- a) the storage of a single coded image or a collection of independently coded images, possibly with derived images;
- b) the storage of image sequences, which can be indicated to be displayed as a timed sequence or by other means, such as a gallery of images, and in which the coded images may be dependent on other coded images in the same sequence.

A file may use both structures and may also use the structures of the ISO base media file format, enabling a single file to be constructed to meet a variety of needs (e.g. a single image for printing and a record of the image burst that was used to synthesize that image).

In general, the single image support is used for simpler cases, particularly when neither timing nor coding dependency is required. If advisory timing or other tools from the ISO base media file format available for tracks are needed (e.g. sample grouping), or images have been coded with inter prediction, then the second approach is needed.

Brands are defined in order to specify what is required to be present in the file, and what reader support is required to decode under that brand (including support for features that are optional for writers). External specifications may also define brands, which may impose additional constraints on the files or the readers. The brands with which a file is compatible are recorded in the file in the usual way using the `FileTypeBox` ('ftyp').

This document is organized as follows.

[Clause 5](#) specifies general requirements on files and file readers conforming to the Image File Format.

[Clause 6](#) specifies the file structures for the storage of a single image and an image collection. Additionally, general requirements that shall be supported in all files using the Image File Format for the storage of a single image or an image collection are specified.

[Clause 7](#) specifies the file structures for the storage of image sequences. Additionally, general requirements that shall be supported in all files using the Image File Format for the storage of image sequences are specified.

[Clause 8](#) specifies the metadata structures for a single image, an image collection, and image sequences.

[Clause 9](#) specifies enhancements to the ISO base media file format (ISO/IEC 14496-12) and may be moved into that specification in the future. [Clauses 6, 7](#) and [8](#) include particular subclauses of [Clause 9](#) by reference into their specifications.

[Clause 10](#) specifies structural brands for a single image and an image collection, as well as image sequences. Requirements on both files and file readers are specified.

[Annex A](#) specifies the format for storing Exif, XMP, and MPEG-7 metadata in files conforming to the Image File Format. Storage of Exif, XMP, or MPEG-7 metadata in files conforming to the Image File Format shall conform to the specifications of [Annex A](#).

[Annex B](#) specifies the format for encapsulating HEVC-coded images, image collections, and image sequences according to the Image File Format. [Annex B](#) also specifies HEVC-specific brands for a single image and an image collection as well as image sequences. Requirements on both files and file readers are specified. Storage of HEVC-coded images, image collections, and image sequences in files conforming to the Image File Format shall conform to the specifications of [Annex B](#).

[Annex C](#) and [Annex D](#) specify the MIME type registration for a single image or an image collection, and image sequences, respectively, for the structural and HEVC-specific brands. When MIME types are used for files conforming to the HEVC-specific Image File Format brands, the MIME types shall conform the specifications of [Annex C](#) and [Annex D](#) for a single image or an image collection, and image sequences, respectively.

[Annex E](#) specifies the format for encapsulating AVC-coded images, image collections, and image sequences according to the Image File Format. Storage of AVC-coded images, image collections, and image sequences in files conforming to the Image File Format shall conform to the specifications of [Annex E](#). [Annex F](#) and [Annex G](#) specify the MIME type registration for a single image or an image collection, and image sequences, respectively, for the AVC-specific brands. When MIME types are used for files conforming to the AVC-specific Image File Format brands, the MIME types shall conform the specifications of [Annex F](#) and [Annex G](#) for a single image or an image collection, and image sequences, respectively.

[Annex H](#) specifies the format for encapsulating JPEG-coded images, image collections, and image sequences according to the Image File Format. Storage of JPEG-coded images, image collections, and image sequences in files conforming to the Image File Format shall conform to the specifications of [Annex H](#).

[Annex I](#) contains guidelines on defining new image formats and brands.

[Annex J](#) contains informative examples of single image and image collection file structures conforming to the Image File Format.

[Annex K](#) provides guidelines for a player operation for progressive refinement and file structures enabling progressive refinement.

Throughout this document, statements appearing with the preamble "NOTE" are informative and are not an integral part of this specification.

5 General requirements

5.1 General requirements on files

All files shall conform to the definitions for an object-structured file as defined in ISO/IEC 14496-12:2015, Clause 4.

5.2 General requirements on readers

The following are the requirements for all readers conforming to this document.

- 1) They shall be able to parse object-structured files formatted according to the definitions for an object-structured file as defined in [Clause 4](#) of the ISO base media file format.
- 2) They shall parse the `FileTypeBox` and confirm that one or more brands that they support are included in the list of compatible brands. If there are no such brands, the reader should terminate parsing of the file.
- 3) They shall be able to recognize and discard boxes that are not required to be supported under the specification identified by the brand(s) under which they are operating.

5.3 Multi-purpose files

Files may be identified as compatible with other standards (using brands) than those defined in this document.

NOTE A file identified as compatible with other standards (using brands) contains the boxes specified by those standards.

5.4 Other boxes

In addition to the required boxes (and their required content), other boxes from the ISO base media file format, or other box-structured specifications, may be included as needed.

6 Single image and image collection

6.1 General

Images can be stored as items using the support for untimed data storage, called the `MetaBox` for historical reasons, in the ISO base media file format. A file may contain any number of image items.

Clause 6 specifies requirements for all files using the Image File Format for the storage of a single image or image collection. In other words, when a brand specified in [10.2](#) is among the compatible brands of a file, the requirements specified in Clause 6 shall be obeyed.

6.2 Derivation from the ISO base media file format

A `MetaBox` ('meta'), as specified in ISO/IEC 14496-12, is required at file level. That `MetaBox` shall contain the boxes specified to be mandatorily present by ISO/IEC 14496-12. Additional requirements for the boxes contained in the file-level `MetaBox` are specified in this document. The `MetaBox` containing image items and the metadata items related to the image items for the brands specified in this document shall be included in the file-level `MetaBox` and shall not be included in any `AdditionalMetadataContainerBox`. The file-level `MetaBox` shall identify as its primary item an item that is a coded image or a derived image item. The primary item should be displayed when no other information is available on the preferred displaying method of the image collection. It is recommended not to have a thumbnail image or an auxiliary image as a primary item.

The handler type for the `MetaBox` shall be 'pict'.

All three construction methods specified for the `ItemLocationBox` (by file offset, by offset into the local `ItemDataBox`, and by offset into the data of another item) are permitted by this document, but brands may restrict this. Similarly, the `DataReferenceBox` may indicate the same or another file, but derived specifications may restrict this also. The location for storing the items is specified in the ISO base media file format. For example, a `MediaDataBox` ('mdat') may be used as a general container for the items referenced from the `ItemLocationBox`.

NOTE By using extents, images can be interleaved with each other or other data, and progressive display may be possible.

[9.2](#), [9.3](#), and [9.4](#) are included in Clause 6 by reference.

6.3 Derivation of an output image of an image item

The reconstructed image of an image item is derived as follows:

- if the image item contains a coded image, the coded image is decoded and the reconstructed image is the output of the decoding process;
- otherwise (the image item is a derived image item), the operation of the derived image item is applied to the input images of the derived image item to form the reconstructed image.

The output image of an image item is derived from the reconstructed image of the image item as follows:

- if the image item has no transformative item properties, the output image is identical to the reconstructed image;
- otherwise (the image item has transformative item properties), the following applies: A sequence of transformative item properties is formed from all essential transformative item properties of the image item and any set of the non-essential transformative item properties of the image item. That sequence of transformative item properties is applied, in the order of their appearance in the `ItemPropertyAssociation` for the image item, to the reconstructed image to obtain the output image.

NOTE When an image item has non-essential transformative item properties, the pixel values of the output image might depend on the reader's capability and/or choice of applying the non-essential transformative item properties.

6.4 Roles of images

6.4.1 General

Images may be assigned different roles or purposes as specified in the following subclauses. The role or the purpose is independent of whether the image is represented by a coded image or a derived image, or how the image is coded or transformed (by a transformative item property).

In this document, the role of an image can be indicated with a qualifier, such as hidden, or thumbnail, in front of the term image, e.g. hidden image, or thumbnail image. When referring to an image item that contains an image with a specific role, the role qualifier can precede the term image item, e.g. hidden image item, or thumbnail image item.

Many of the roles specified below are not mutually exclusive. Consequently, the same image may have multiple roles.

NOTE One example is of an image with multiple roles is a hidden auxiliary image.

6.4.2 Hidden images

A hidden image item has (flags & 1) equal to 1 in its ItemInfoEntry, as specified in 9.2. Readers should not display a hidden image item.

NOTE A hidden image item can be, for example, an image item that is used as an input image for a derived image item but is never intended to be displayed itself.

The primary item shall not be a hidden image item.

Any entity group of type 'altr' that includes image items, shall either include only hidden items or only non-hidden items (i.e. a group of this type cannot contain a mix of hidden and non-hidden items).

6.4.3 Cover image

For a collection of images carried as items in a MetaBox, the primary item of the MetaBox should be displayed when no other information is available on the preference to display a collection of images.

6.4.4 Thumbnail images

A thumbnail image is a smaller-resolution representation of a master image. The thumbnail image and the master image are linked using a reference type 'thmb' from the thumbnail image to the master image. A thumbnail image shall not be linked to another thumbnail image with the 'thmb' item reference.

6.4.5 Auxiliary images

Auxiliary images are images, which are not thumbnail images, related to a master image. An example of an auxiliary image is an alpha plane, specifying transparency information, for the master image. The auxiliary image and the master image are linked using an item reference of 'aux1' from the auxiliary image to the master image.

NOTE The type of auxiliary image is identified as defined in 6.5.8.

6.4.6 Master images

A master image is an image that is not an auxiliary image or a thumbnail image. Auxiliary images and thumbnail images are associated with master images through item references. A master image typically represents a full-resolution displayable image, whereas a thumbnail image is a smaller-resolution representation of the master image and only intended to be displayed on specific occasions and an auxiliary image is typically not intended to be displayed.

6.4.7 Pre-derived coded images

If a coded image has been derived from others, for example, a composite HDR image derived from exposure-bracketed individual images, then it shall be linked to those images by item references of type 'base' from the coded image to all images it derives from. An image item including a 'base' item reference is referred to as a pre-derived coded image.

NOTE In this version of this document, the exact derivation process used to produce the image is not described.

6.4.8 Multi-layer images

Some coding formats allow coding of images in a multi-layer manner, where the coded data representing an image is partitioned into several layers. A basic version of the image can be obtained by decoding the base layer, and each enhancement layer improves the basic version with respect to one or more aspects, such as spatial resolution and bit depth. In another example, an enhancement layer provides a second view, which can be used in stereoscopic displaying.

It is possible to specify several image items from the same multi-layer coded data representing an image, where each image item represents a different subset of the layers. Only one occurrence of the coded data is needed in a file, while the same layer can be included in several image items by using extents.

Each layer in a multi-layer image is associated with layer identifier different from the layer identifiers of the other layers of the image. The assignment of layer identifiers is specific to the image coding format or the mapping of the image coding format to this specification.

The decoding of a multi-layer image item may result into one or more reconstructed images.

NOTE 1 For example, when a multi-layer image contains two views, a decoder is typically specified to decode and return the decoded images of both the views.

When the decoding of a multi-layer image item results into more than one reconstructed images, a `LayerSelectorProperty` item property shall be present for the image item.

NOTE 2 The layer given in the `LayerSelectorProperty` item property chooses the reconstructed image from the set of reconstructed images obtained as the result of decoding the image item. The chosen reconstructed image might not be presented as is but might undergo further transformation (for example, cropping) before actually being presented.

Some coding formats allow the use of decoded images of another bitstream as references for prediction. In such cases, there shall be an item reference of type 'exbl' from a scalably coded image item to the image item that is first decoded and then used as reference in the decoding of the scalably coded image item.

6.5 Image properties

6.5.1 General

6.5 specifies item properties that can be used to describe image items or to affect the output image generation as specified in 6.3. This includes properties based on metadata already defined in ISO/IEC 14496-12, such as colour information and pixel aspect ratio, as well as properties specified in this document, such as spatial extents of image items.

Properties are ordered. Transformative properties apply to the image with preceding transformations applied.

The semantics of the descriptive properties specified in 6.5 are specified for the image before the transformations, if any, are applied. Readers shall allow and ignore descriptive properties following the first transformative or unrecognized property, whichever is earlier, in the sequence associating properties with an item.

Writers should arrange the descriptive properties specified in 6.5 prior to any other properties in the sequence associating properties with an item.

Descriptive properties are non-essential, unless stated otherwise in their specification.

NOTE Each property descriptor definition below specifies whether the property is mandatory, i.e. whether the property is required to be present in the file. Note that when a property is mandatory, it might or might not be essential.

6.5.2 Decoder configuration and initialization

Box type:	Specified by the mapping of the image coding format to this document
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Specified by the mapping of the image coding format to this document
Quantity (per item):	Specified by the mapping of the image coding format to this document

There may be zero or more properties associated with a coded image item that are specific to the image coding format used by that item, that configure the decoder.

NOTE Decoder configuration properties are essential, since they are required to initialize decoders.

6.5.3 Image spatial extents

6.5.3.1 Definition

Box type:	'ispe'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes
Quantity (per item):	One

The ImageSpatialExtentsProperty documents the width and height of the associated image item. Every image item shall be associated with one property of this type, prior to the association of all transformative properties.

6.5.3.2 Syntax

```
aligned(8) class ImageSpatialExtentsProperty
extends ItemFullProperty('ispe', version = 0, flags = 0) {
    unsigned int(32) image_width;
    unsigned int(32) image_height;
}
```

6.5.3.3 Semantics

`image_width` specifies the width of the reconstructed image in pixels, as specified in [6.5.3.1](#).

`image_height` specifies the height of the reconstructed image in pixels, as specified in [6.5.3.1](#).

6.5.4 Pixel aspect ratio

6.5.4.1 Definition

Box type:	'pasp'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	At most one

6.5.4.2 Syntax

The pixel aspect ratio 'pasp' descriptive item property has the same syntax as the PixelAspectRatioBox as defined in ISO/IEC 14496-12.

6.5.4.3 Semantics

The semantics of the syntax elements within the pixel aspect ratio 'pasp' descriptive item property are the same as those specified for the syntax elements of PixelAspectRatioBox as defined in ISO/IEC 14496-12.

6.5.5 Colour information

6.5.5.1 Definition

Box type:	'colr'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	At most one

6.5.5.2 Syntax

The colour information 'colr' descriptive item property has the same syntax as the ColourInformationBox as defined in ISO/IEC 14496-12.

6.5.5.3 Semantics

The semantics of the syntax elements within colour information 'colr' descriptive item property are the same as those specified for the syntax elements of ColourInformationBox as defined in ISO/IEC 14496-12.

6.5.6 Pixel information

6.5.6.1 Definition

Box type:	'pixi'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	At most one

The `PixelInformationProperty` descriptive item property indicates the number and bit depth of colour components in the reconstructed image of the associated image item.

6.5.6.2 Syntax

```
aligned(8) class PixelInformationProperty
extends ItemFullProperty('pixi', version = 0, flags = 0){
    unsigned int (8) num_channels;
    for (i=0; i<num_channels; i++) {
        unsigned int (8) bits_per_channel;
    }
}
```

6.5.6.3 Semantics

`num_channels`: This field signals the number of channels by each pixel of the reconstructed image of the associated image item.

`bits_per_channel`: This field indicates the bits per channel for the pixels of the reconstructed image of the associated image item.

6.5.7 Relative location

6.5.7.1 Definition

Box type:	'rloc'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, if the item has a 'tbas' item reference to another image item.
Quantity (per item):	At most one

The `RelativeLocationProperty` descriptive item property is used to describe the horizontal and vertical position of the reconstructed image of the associated image item relative to the reconstructed image of the related image item identified through the 'tbas' item reference as specified below. The pixel sampling of the associated image item shall be identical to that of the related image item and the sampling grids of the associated image item and the related image item shall be aligned (i.e. not have a sub-pixel offset). Consequently, one pixel in the associated image item collocates to exactly one pixel in the related image item. The related image item shall be identified by an item reference of type 'tbas' from the associated image item to the related image item.

6.5.7.2 Syntax

```
aligned(8) class RelativeLocationProperty
extends ItemFullProperty('rloc', version = 0, flags = 0)
{
    unsigned int(32) horizontal_offset;
    unsigned int(32) vertical_offset;
}
```

6.5.7.3 Semantics

horizontal_offset specifies the horizontal offset in pixels of the left-most pixel column of the reconstructed image of the associated image item in the reconstructed image of the related image item. The left-most pixel column of the reconstructed image of the related image item has a horizontal offset equal to 0.

vertical_offset specifies the vertical offset in pixels of the top-most pixel row of the reconstructed image of the associated image item in the reconstructed image of the related image item. The top-most pixel row of the reconstructed image of the related image item has a vertical offset equal to 0.

6.5.8 Image properties for auxiliary images

6.5.8.1 Definition

Box type:	'auxc'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for an image item containing an auxiliary image
Quantity (per item):	At most one

Auxiliary images shall be associated with an `AuxiliaryTypeProperty` as defined here. `AuxiliaryTypeProperty` includes a URN identifying the type of the auxiliary image. `AuxiliaryTypeProperty` may additionally include other fields, as required by the URN.

6.5.8.2 Syntax

```
aligned(8) class AuxiliaryTypeProperty
extends ItemFullProperty('auxC', version = 0, flags) {
    string aux_type;
    template unsigned int(8) aux_subtype[];
    // until the end of the box, the semantics depend on the aux_type value
}
```

6.5.8.3 Semantics

aux_type: A null-terminated UTF-8 character string of the Uniform Resource Name (URN) used to identify the type of the associated auxiliary image item.

aux_subtype: Zero or more bytes until the end of the box. The semantics of these bytes depend on the value of **aux_type**.

6.5.9 Clean aperture

6.5.9.1 Definition

Box type:	'clap'
Property type:	Transformative item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	At most one

The clean aperture transformative item property defines a cropping transformation of the input image.

6.5.9.2 Syntax

The clean aperture 'clap' transformative item property has the same syntax as the CleanApertureBox as defined in ISO/IEC 14496-12.

6.5.9.3 Semantics

The semantics of the syntax elements within the clean aperture 'clap' transformative item property are the same as those specified for the syntax elements of CleanApertureBox as defined in ISO/IEC 14496-12.

6.5.10 Image rotation

6.5.10.1 Definition

Box type:	'irot'
Property type:	Transformative item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	At most one

The image rotation 'irot' transformative item property rotates the reconstructed image of the associated image item in anti-clockwise direction in units of 90 degrees.

6.5.10.2 Syntax

```
aligned(8) class ImageRotation
extends ItemProperty('irot') {
    unsigned int (6) reserved = 0;
    unsigned int (2) angle;
}
```

6.5.10.3 Semantics

angle * 90 specifies the angle (in anti-clockwise direction) in units of degrees.

6.5.11 Layer selection

6.5.11.1 Definition

Box type:	'lssel'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	Zero or one

If the decoding of a multi-layer image item results into more than one reconstructed image, the 'lssel' item property shall be associated with the image item. Otherwise, the 'lssel' item property shall not be associated with an image item.

essential shall be equal to 1 for an 'lssel' item property.

The 'lssel' item property, if any, shall precede, in the item property association order, all transformative item properties.

This property is used to select which of the reconstructed images is described by subsequent descriptive item properties in the item property association order and manipulated by transformative item properties, if any, to generate an output image of the image item.

6.5.11.2 Syntax

```
aligned(8) class LayerSelectorProperty
extends ItemProperty('lssel') {
    unsigned int(16) layer_id;
}
```

6.5.11.3 Semantics

layer_id specifies the layer identifier of the image among the reconstructed images that is described by subsequent descriptive item properties in the item property association order and manipulated by transformative item properties, if any, to generate an output image of the image item. The semantics of layer_id are specific to the coding format and are therefore defined for each coding format for which the decoding of a multi-layer image item can result into more than one reconstructed images.

6.5.12 Image mirroring

6.5.12.1 Definition

Box type:	'imir'
Property type:	Transformative item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	Zero or one

The image mirroring 'imir' transformative item property mirrors the image about either a vertical or horizontal axis.

6.5.12.2 Syntax

```
aligned(8) class ImageMirror
extends ItemProperty('imir') {
    unsigned int (7) reserved = 0;
    unsigned int (1) axis;
}
```

6.5.12.3 Semantics

`axis` specifies a vertical (`axis = 0`) or horizontal (`axis = 1`) axis for the mirroring operation.

6.6 Derived images and derived image items

6.6.1 General

An item is a derived image item, when it includes a 'dimg' item reference to one or more other image items, which are inputs to the derivation. The reconstructed image of a derived image item is obtained as described in 6.3. The exact operation performed to obtain the reconstructed image is identified by the `item_type` of the item. The image items used as input to a derived image item are output images of other image items, which may be coded image items or derived image items.

The output image of a derived image item is obtained, also as defined in 6.3, by applying transformative item properties to the reconstructed image.

The number of `SingleItemTypeReferenceBoxes` with the box type 'dimg' and with the same value of `from_item_ID` shall not be greater than 1.

The following subclauses specify the `item_type` and the syntax of the item data for some derived image items.

6.6.2 Derived image types and derived image item types

6.6.2.1 Identity derivation

A derived image item of the `item_type` value 'iden' (identity transformation) may be used when it is desired to use transformative properties to derive an image item. The derived image item shall have no item body (i.e. no extents), and `reference_count` for the 'dimg' item reference of a 'iden' derived image item shall be equal to 1.

NOTE 1 A derived image item of type 'iden' has an empty item body, and the output image is the result of applying the transformative item properties associated with this derived image item.

NOTE 2 A derived image of type 'iden' can be used, for example, when it is desirable to have both the original version of a coded image item and a cropped version of the same coded image item (obtained through the 'clap' transformative item property) as non-hidden image items.

6.6.2.2 Image overlay derivation

6.6.2.2.1 Definition

An item with an `item_type` value of 'iovl' defines a derived image item by overlaying one or more input images in a given layering order within a larger canvas. The input images are listed in the order they are layered, i.e. the bottom-most input image first and the top-most input image last, in the `SingleItemTypeReferenceBox` of type 'dimg' for this derived image item within the `ItemReferenceBox`.

NOTE File writers need to be careful when removing an item that is marked as an input image of an image overlay item, as the content of the image overlay item might need to be rewritten.

6.6.2.2.2 Syntax

```
aligned(8) class ImageOverlay {
    unsigned int(8) version = 0;
    unsigned int(8) flags;
    for (j=0; j<4; j++) {
        unsigned int(16) canvas_fill_value;
    }
    FieldLength = ((flags & 1) + 1) * 16;
    unsigned int(FieldLength) output_width;
    unsigned int(FieldLength) output_height;
    for (i=0; i<reference_count; i++) {
        signed int(FieldLength) horizontal_offset;
        signed int(FieldLength) vertical_offset;
    }
}
```

6.6.2.2.3 Semantics

version shall be equal to 0.

(flags & 1) equal to 0 specifies that the length of the fields output_width, output_height, horizontal_offset, and vertical_offset is 16 bits. (flags & 1) equal to 1 specifies that the length of the fields output_width, output_height, horizontal_offset, and vertical_offset is 32 bits. The values of flags greater than 1 are reserved.

canvas_fill_value: Indicates the pixel value per channels used if no pixel of any input image is located at a particular pixel location. The fill values are specified as RGBA (R, G, B, and A corresponding to loop counter j equal to 0, 1, 2, and 3, respectively). The RGB values are in the sRGB color space as defined in IEC 61966-2-1. The A value is a linear opacity value ranging from 0 (fully transparent) to 65535 (fully opaque).

output_width, output_height: Specify the width and height, respectively, of the reconstructed image on which the input images are placed. The image area of the reconstructed image is referred to as the canvas.

reference_count is obtained from the SingleItemTypeReferenceBox of type 'dimg' where this item is identified by the from_item_ID field.

horizontal_offset, vertical_offset: Specifies the offset, from the top-left corner of the canvas, to which the input image is located. Pixel locations with a negative offset value are not included in the reconstructed image. Horizontal pixel locations greater than or equal to output_width are not included in the reconstructed image. Vertical pixel locations greater than or equal to output_height are not included in the reconstructed image.

6.6.2.3 Image grid derivation

6.6.2.3.1 Definition

An item with an item_type value of 'grid' defines a derived image item whose reconstructed image is formed from one or more input images in a given grid order within a larger canvas.

The input images are inserted in row-major order, top-row first, left to right, in the order of SingleItemTypeReferenceBox of type 'dimg' for this derived image item within the ItemReferenceBox. In the SingleItemTypeReferenceBox of type 'dimg', the value of from_item_ID identifies the derived image item of type 'grid', the value of reference_count shall be equal to rows*columns, and the values of to_item_ID identify the input images. All input images shall have exactly the same width and height; call those tile_width and tile_height. The tiled input images shall completely "cover" the reconstructed image grid canvas, where tile_width*columns is greater than or equal to output_width and tile_height*rows is greater than or equal to output_height.

The reconstructed image is formed by tiling the input images into a grid with a column width (potentially excluding the right-most column) equal to `tile_width` and a row height (potentially excluding the bottom-most row) equal to `tile_height`, without gap or overlap, and then trimming on the right and the bottom to the indicated `output_width` and `output_height`.

NOTE 1 If the desired input images are not of a consistent size, then derived image items that scale or crop them, as needed to make them consistent, can be used; other specifications can, however, restrict whether derived image items are permissible as input to the image grid derived image item. This document specifies cropping in 6.5.8 but does not specify a derived image item for scaling; hence, other specifications would need to specify scaling if such is desired for making input images having a consistent width and height for the image grid derived image item.

NOTE 2 File writers need to be careful when removing an item that is marked as an input image of an image grid item, as the content of the image grid item may need to be rewritten.

6.6.2.3.2 Syntax

```
aligned(8) class ImageGrid {
    unsigned int(8) version = 0;
    unsigned int(8) flags;
    FieldLength = ((flags & 1) + 1) * 16;
    unsigned int(8) rows_minus_one;
    unsigned int(8) columns_minus_one;
    unsigned int(FieldLength) output_width;
    unsigned int(FieldLength) output_height;
}
```

6.6.2.3.3 Semantics

`version` shall be equal to 0. Readers shall not process an `ImageGrid` with an unrecognized version number.

`(flags & 1)` equal to 0 specifies that the length of the fields `output_width`, `output_height`, is 16 bits. `(flags & 1)` equal to 1 specifies that the length of the fields `output_width`, `output_height`, is 32 bits. The values of `flags` greater than 1 are reserved.

`output_width`, `output_height`: Specify the width and height, respectively, of the reconstructed image on which the input images are placed. The image area of the reconstructed image is referred to as the canvas.

`rows_minus_one`, `columns_minus_one`: Specify the number of rows of input images and the number of input images per row. The value is one less than the number of rows or columns respectively. Input images populate the top row first, followed by the second and following, in the order of item references.

6.7 Image metadata

The metadata that describes an image is formed as the union of the items that refer from the metadata item to the image item using the '`cdsc`' (content describes) item reference.

NOTE A union is used so that common metadata, e.g. camera owner, model, and so on, can be stored once, and then each image can also have image-specific metadata.

[Clause 8](#) specifies file format structures related to storing metadata describing images in files conforming to the Image File Format and [Annex A](#) specifies how to store metadata of certain schematic languages in files conforming to this document.

6.8 Relating an untimed item to a timed sequence

6.8.1 'equiv' entity group

It is useful in some situations to be able to say that a given untimed image relates to a particular position in the timeline of a track. An entity group of type 'equiv' (equivalence) can be used for this purpose. Equivalent images are visually substitutable, but possibly coded differently (e.g. different resolution, compression, etc.). This differs from the 'altr' entity group in that it applies to selected images in tracks as specified below, not whole tracks.

The semantics of the 'equiv' entity group are that all the items included in an 'equiv' entity group are 'equivalent' and that the tracks in the same 'equiv' entity group include selected samples, as specified below, that are 'equivalent' to the items.

If there are multiple visual tracks, each in a different entity group (providing image bursts associated with different images) and those visual tracks each have other associated tracks (e.g. audio), then the 'msrc' track group as defined in ISO/IEC 14496-12:2015, 8.3.4.3 should be used to associate each visual track with its associated audio track.

6.8.2 'equiv' sample group

6.8.2.1 Definition

Tracks included in an 'equiv' entity group should have a sample group of type 'equiv'. The value of `grouping_type_parameter` of the 'equiv' sample group shall be equal to the `group_id` of the 'equiv' entity group. There may be several sample groups of type 'equiv', each with a different value of `grouping_type_parameter`.

All the samples marked by an 'equiv' sample group are 'equivalent' to each other and to the items in an entity group with `group_id` equal to `grouping_type_parameter` of the sample group.

In the case that an 'equiv' entity group contains one image item and one track that marks only one sample, the timing of the sample documents the time that the image item was drawn from.

There is usually at most one sample in a given track that is equivalent to a given image item.

If there is no sample in a given track that is exactly equivalent to the image item(s) in the same 'equiv' entity group but the image item(s) anyway relate to a particular time in the given track, the `time_offset` field of the sample group description entry indicates the difference of the particular time related to the image item(s) and the composition time of the sample associated with the sample group description entry. The following notation is used:

- C: the composition time of the associated sample;
- S: the media timescale of the track;
- O: `time_offset`;
- M: `timescale_multiplier`.

The identified time, in the media timescale of the track, for the image item(s) of the associated 'equiv' entity group is

$$T = C + O/(M/256)$$

6.8.2.2 Syntax

```
class VisualEquivalenceEntry() extends VisualSampleGroupEntry ('equiv')
{
    signed int(16) time_offset;
    unsigned int(16) timescale_multiplier;
}
```

6.8.2.3 Semantics

`time_offset` specifies the difference of the time related to the image item(s) in the associated 'equiv' entity group and the composition time of the sample associated with the sample group description entry, as specified above. `time_offset` is an integer expressed in the timescale resulting from the `timescale_multiplier` field. The `time_offset` shall, when positive, be less than the duration of the associated sample, unless the sample is last in its track, whereupon it may be equal to the duration but no greater. A negative offset shall only be associated with the first sample in a track.

`timescale_multiplier` specifies the timescale, as a multiplier to the media timescale of the track, that is used to indicate the difference of the time related to the image item(s) in the associated 'equiv' entity group and the composition time of the sample associated with the sample group description entry. `timescale_multiplier` is an 8.8 fixed-point value. The recommended value of the `timescale_multiplier` is 1.0 (represented as 1<<8). The value 0 for `timescale_multiplier` is reserved and shall not be used.

7 Image sequences

7.1 General

Image sequences are stored in the ISO base media file format in tracks. In order to distinguish image sequences from video, the handler type in the `HandlerBox` of the track is 'pict' to indicate an image sequence track. In particular, in an image sequence track, the timing is advisory: it may be the timing at collection (e.g. of an image burst) or the suggested display timing (e.g. for a slide show). In all other respects, the definitions and requirements for a video track apply unless otherwise specified in this document.

Clause 7 specifies requirements for all files using the Image File Format for the storage of image sequences. In other words, when a brand specified in [10.3](#) is among the compatible brands of a file, the requirements specified in Clause 7 shall be obeyed.

Files containing an image sequence should also contain a file-level `MetaBox` with a primary item that is an image item as specified in [Clause 6](#), for cases in which temporal presentation is either undesirable, or not possible (e.g. printing).

NOTE The primary item can share coded data with one of the intra-coded images in the sequence.

7.2 Derivation from the ISO base media file format

7.2.1 Track Header box

The specifications of the `TrackHeaderBox` in ISO/IEC 14496-12 apply with the following changes.

- The syntax of the `matrix` syntax element is replaced with
 - `int(32)[9] matrix;`
- The semantics of the `matrix` syntax element are specified in ISO/IEC 14496-12:2015, 6.2.2.
- The following constraints on the value of `matrix` shall be obeyed:
 - either `a` or `c` but not both shall be equal to 0x0001000 or 0xFFFF0000, while the remaining one of `a` and `c` shall be equal to 0;

- either *b* or *d* but not both shall be equal to 0x0001000 or 0xFFFF0000, while the remaining one of *b* and *d* shall be equal to 0.

NOTE 1 These combinations of *a*, *b*, *c*, and *d* values specify combinations of horizontal and vertical mirroring and counter-clockwise rotation by 0°, 90°, 180°, and 270°.

- The values of *x* and *y* are not constrained. Players are allowed to translate the image implicitly to a coordinate space with non-negative coordinates.
- *u* and *v* shall be equal to 0, and *w* shall be equal to 0x40000000.

NOTE 2 As implied in ISO/IEC 14496-12, when a `CleanApertureBox` is present in a sample entry, the clipping specified by the `CleanApertureBox` takes place before applying the rotation specified by the matrix syntax element.

7.2.2 Handler type

The handler type of an image sequence track shall be 'pict'. When the syntax and semantics of features of the ISO base media file format are applied to a track with the 'pict' handler type, the specifications for a track with the 'vide' handler type apply, unless otherwise specified in this document. Specifically, when `handler_type` is equal to 'pict', the `VisualSampleEntry` structure is used in the `SampleDescriptionBox` and the `VisualSampleGroupEntry` structure is used in the `SampleGroupDescriptionBox`. The sample entry shall be used as specified for storage in a track with the handler type 'vide'.

7.2.3 Coding Constraints box

7.2.3.1 General

The `CodingConstraintsBox` shall be present in the sample description entry for tracks with `handler_type` equal to 'pict' and may be present for other tracks. The `CodingConstraintsBox` includes fields specifying that certain constraints are obeyed in the samples of the track, as specified by the semantics of the fields below.

7.2.3.2 Definition

Box type: 'ccst'
 Container: Sample entry
 Mandatory: Yes
 Quantity: One

7.2.3.3 Syntax

```
class CodingConstraintsBox extends FullBox('ccst', version = 0, flags = 0){
    unsigned int (1) all_ref_pics_intra;
    unsigned int (1) intra_pred_used;
    unsigned int (4) max_ref_per_pic;
    unsigned int(26) reserved;
}
```

7.2.3.4 Semantics

`all_ref_pics_intra`: This flag when set to one indicates the restriction that samples that are not sync samples, if any, are predicted only from sync samples.

NOTE 1 When there are inter predicted images in the track and `all_ref_pics_intra` is equal to 1, then these images are all predicted from intra coded images.

`intra_pred_used` equal to 0 indicates that intra prediction is not used in the inter predicted images referring to the sample entry containing this `CodingConstraintsBox`. `intra_pred_used` equal to 1 indicates that intra prediction may or may not be used in the inter predicted images referring to the sample entry containing this `CodingConstraintsBox`.

NOTE 2 A decoder that is interested in only a specific region of an inter predicted image can choose not to decode any of the coding units outside the boundary of its region of interest when no intra prediction is used.

`max_ref_per_pic` indicates the maximum number of reference images that may be used for decoding any single image within an image sequence. The value 15 is reserved to indicate that any number of reference images permitted by the sample entry may be used.

`reserved` shall be equal to 0 in files conforming to this version of this specification. The value of `reserved` shall be ignored by readers.

7.3 Presentation of an image sequence track

An image sequence track is presented like any other media track, as specified in ISO/IEC 14496-12, with the additional specifications below.

The `TimeToSampleBox` of a track with handler type 'pict' (and their matching structures in movie fragments) may or may not provide conforming timing according to the codec being used. An `EditListBox` may be used in such a track. It can be indicated as follows whether an image sequence track is intended to be displayed as a timed sequence or by other means, such as a gallery of images. If either (a) an `EditListBox` is present, and `edit(s)` indicate the playback of more than one sample or (b) an `EditListBox` is not present, then a file reader should attempt to follow the provided timing in presenting the image sequence. Otherwise (an `EditListBox` indicates the playback of zero or one samples), a file reader should display all the samples that are not hidden samples but ignore their timing. A hidden sample is a sample for which all of the following are true.

- The image sequence track includes version 1 of the `CompositionOffsetBox`.
- The value of `sample_offset` is equal to -2^{31} .
- The `CompositionToDecodeBox` is contained in the `SampleTableBox` of the image sequence track.
- The value of `leastDecodeToDisplayDelta` field in the `CompositionToDecodeBox` is greater than -2^{31} .

The specifications of 9.6 apply for repeating edits.

7.4 Sample groups

7.4.1 Direct reference samples list

7.4.1.1 Definition

The following terms are defined for the specification of this sample group:

- direct reference sample (for a second image): sample, as defined by ISO/IEC 14496-12, containing a first image that may be used as a reference for inter prediction of the second image;
- indirect reference sample (for a second image): sample, as defined by ISO/IEC 14496-12, that is not a direct reference sample for the second image and is a direct reference sample for a third image contained in a reference sample for the second image;
- reference sample: direct reference sample or indirect reference sample;
- non-reference sample: sample, as defined by ISO/IEC 14496-12, containing an image that is not used as a reference for inter prediction of another image.

A sample that is predicted from other samples requires all its reference samples to be decoded prior to its decoding. The direct reference samples list 'refs' is a sample group that identifies all the direct reference samples for a sample.

This sample group entry consists of fields: (a) `sample_id`, and (b) a list of `direct_reference_sample_id` values. For samples that may be used as a reference for predicting other samples, the `sample_id` field is a positive integer and unique. A non-reference sample is given a `sample_id` value of zero. A list of `direct_reference_sample_id` lists the `sample_id` values of all the direct reference samples for a sample belonging to the group.

NOTE 1 A sample that is mapped to a 'refs' sample group entry with `sample_id` not equal to 0 and with `num_direct_reference_samples` equal to 0 is not required to be a sync sample but can contain an intra-coded image that does not qualify as a sync sample.

When samples that are not sync samples are present in an image sequence track, the 'refs' sample group should be present for the track.

When the 'refs' sample group is present, it is required that the sequence of samples consisting of the following samples in decoding order conforms to the requirements imposed by the sample entry, without the processing of the other samples in the track:

- a set of samples s_i for i equal to 1 to N , inclusive, where N is any value from 1 to the number of samples in the track, inclusive;
- the direct and indirect reference samples r_j for samples s_i for each i from 1 to N , inclusive;
- each sync sample q_k such that q_k is not among s_i or r_j for any values of i and j , and q_k is the previous sync sample, in decoding order, for at least one sample s_i or r_j that is not a sync sample.

NOTE 2 Consequently, the sequence of samples consisting of s_i , r_j , and q_k for all values of i , j , and k , in decoding order, can be decoded with a decoder that conforms to the indicated sample entry.

The version of the `SampleGroupDescriptionBox` for the 'refs' sample group shall be greater than or equal to 1.

7.4.1.2 Syntax

```
class DirectReferenceSamplesList()
extends VisualSampleGroupEntry ('refs') {
    unsigned int(32) sample_id;
    unsigned int(8) num_direct_reference_samples;
    for(i = 0; i < num_direct_reference_samples; i++) {
        unsigned int(32) direct_reference_sample_id;
    }
}
```

7.4.1.3 Semantics

sample_id: When the sample group entry corresponds to a reference sample, the value of this field shall be a positive integer. The value for this field shall be zero for non-reference samples.

num_direct_reference_samples: The number of the direct reference samples required for decoding an inter-predicted sample.

NOTE Only the direct reference samples are counted in `num_direct_reference_samples`. When a first sample is used as a reference for inter prediction of a second sample, and the second sample is used as a reference for inter prediction of a third sample, but the first sample is not (directly) used as a reference for inter prediction of the third sample, the first sample is not included in the value of `num_direct_reference_samples`.

direct_reference_sample_id: The value of this field shall be set to the `sample_id` values of the direct reference samples that a sample belonging to this group may be predicted from.

7.5 Other tracks

7.5.1 General

This document does not preclude other tracks, for example, those defined by ISO/IEC 14496-12 or ISO/IEC 14496-15, to be present in the file. If other tracks are present in the file, the support for such track may be documented by using the `FileTypeInfoBox`.

7.5.2 Thumbnail image sequence track

When present, thumbnails for samples carried in a track with handler type 'pict' are carried in another linked track also with a handler type 'pict'. A track reference of type 'thmb' is used to link the thumbnail track to the track, as specified in 9.5. Linking of individual samples in the image track and the related sample in thumbnail track is done using sample timings. The sample in a thumbnail image sequence track that is associated with a sample in a master image sequence track is the time-parallel sample in the thumbnail image sequence track relative to the sample in the master image sequence track, as defined in 3.1.29. Not all samples in a master image sequence track need to have a corresponding sample in a thumbnail track. The `track_in_preview` flag of the `TrackHeaderBox` 'tkhd' of the thumbnail track should be set to 1. Furthermore, the thumbnail track and the image sequence track should be signalled to be a part of the same alternate group, by setting the same integer value for field `alternate_group` in the track headers of these tracks.

7.5.3 Auxiliary image sequence track

7.5.3.1 General

Any number of tracks with handler type 'auxv' may be included in files containing image sequence tracks.

As auxiliary image sequence tracks are not intended to be displayed as such, the `track_in_movie` flag in `TrackHeaderBox` of auxiliary image sequences tracks should be equal to 0.

The master image sequence track is linked to the auxiliary track using the 'aux1' track reference included in the auxiliary track.

The nature of the auxiliary track is announced by the `AuxiliaryTypeInfoBox` that shall be included in the sample entry of the auxiliary track.

Linking of samples in the auxiliary image sequence and its master image sequence is handled by the values encoded in the `TimeToSampleBox` ('stts'). The sample in an auxiliary image sequence track that is associated with a sample in a master image sequence track is the time-parallel sample in the auxiliary image sequence track relative to the sample in the master image sequence track, as defined in 3.1.39.

7.5.3.2 Definition

Box type:	'auxi'
Container:	Sample entry
Mandatory:	Yes (for an auxiliary image sequence track)
Quantity:	One

7.5.3.3 Syntax

```
aligned(8) class AuxiliaryTypeInfoBox
extends FullBox ('auxi') {
    string aux_track_type;
}
```

7.5.3.4 Semantics

`aux_track_type`: A null-terminated UTF-8 character string of the Uniform Resource Name (URN) used to identify the type of the auxiliary images associated with this sample entry. The URN identifies the resource and the index in the resource which identifies the type of the auxiliary images associated with this sample entry.

8 Metadata support

8.1 General

This clause specifies file format structures related to the storage of metadata. This clause also contains the metadata structures specified in the Image File Format. In addition, this document supports the carriage of images and image sequences along with metadata written in various metadata schematic languages. Examples of such schematic languages include Exif and MPEG-7. [Annex A](#) specifies how to store metadata of certain schematic languages in files conforming to this document. Metadata according to [Annex A](#) may or may not be present in files conforming to this document. Additionally, other forms of metadata may be present, identified using suitable item type and MIME type values.

Metadata specified in [Annex A](#) or according to the item type and MIME type values is descriptive and does not normatively affect the presentation. In particular, an image item can be rotated by 90°, 180°, or 270° using the '`irotd`' transformative item property. Rotation metadata, e.g. according to [Annex A](#), is ignored in the displaying process.

8.2 Metadata for image items

Metadata items are linked to the images they describe by item references of type '`cdsc`'. Metadata items take an appropriate type, and may use the tools for items, including content protection. The metadata applicable to an image is the union of the metadata thus linked, though it should be noted that the same information may be expressed in multiple ways.

8.3 Metadata for image sequence tracks

Timed metadata tracks may be used to define metadata for image sequences. They are linked to the image sequence by a track reference of type '`cdsc`'.

When metadata tracks are used to describe data, a metadata sample describing a media sample is the time-parallel sample in the metadata track relative to the media sample, as defined in [3.1.39](#).

If two or more metadata tracks linked to an image sequence track are parts of the same alternate group, any one of these metadata tracks can be parsed to obtain applicable metadata for the image sequence track. Metadata tracks that are linked to an image sequence track and that are not parts of the same alternate group are complementary, and applicable metadata for an image in the image sequence track is the union of the contents of the time-parallel samples in these metadata tracks to the sample containing the image in question.

A track level `MetaBox` can be used to describe contents that are specific to the image sequence as a whole.

When samples of an image sequence have to be linked to one or more metadata items contained in a `MetaBox` in track, the sample grouping `SampleToMetadataItemEntry` as defined in [9.7](#) is used.

8.4 Integrity checks

8.4.1 General

While this document provides a way of describing metadata about some or all of a group of images in an image sequence, it is not required for an entity adhering to this document to understand the metadata.

When an entity does not understand the schematic language used for describing the metadata, it shall ignore the contents of the metadata items using that schematic language in their syntax.

Since the data can be modified by an entity that does not understand the metadata, a situation where the metadata no longer describes the actual image samples carried in the track can occur.

In order to warn an entity conforming to this document that the metadata might be out of sync with respect to the images in the image sequence, the `DataIntegrity` item is defined.

When a track is edited by an entity that understands both this integrity data and the metadata formats in use, it should adjust the integrity data and/or the metadata items describing the samples as needed.

The `DataIntegrity` item shall only occur in a `MetaBox` in a track (not in a movie or file-level `MetaBox`). The `DataIntegrity` item has the `item_type` value 'mint'. In this version of the Image File Format, the `DataIntegrity` item shall consist of one or more `MD5IntegrityBoxes`, although readers shall allow and ignore other types of contained boxes too. There shall be at least one item reference of type 'mint' from the `DataIntegrity` item to a metadata item that describes or is otherwise associated with the samples indicated by the `MD5IntegrityBoxes` included in the `DataIntegrity` item.

8.4.2 Syntax

```
aligned(8) class DataIntegrity {
    while (MoreDataInItem())
        MD5IntegrityBox();
}
aligned(8) class MD5IntegrityBox()
extends FullBox('md5i', version = 0, flags) {
    unsigned int(8)[16] input_MD5;
    unsigned int(32) input_4cc;
    if (input_4cc == 'sgpd') {
        unsigned int(32) grouping_type;
        if (flags&1)
            unsigned int(32) grouping_type_parameter;
        unsigned int(32) num_entries;
        for(i=0; i<num_entries; i++) {
            unsigned int(32) group_description_index[i];
        }
    }
}
```

8.4.3 Semantics

`MoreDataInItem()` is specified to return 0 when no more data follows in this item, and is otherwise specified to return 1.

`input_MD5`: the 128 bit MD5 digest. The value of this field is the digest in network byte order, as specified in IETF RFC 1864 prior to base64 conversion to a string.

`input_4cc`: The input data over which the MD5 string is computed, which shall be one of the following.

- 'stsz': The MD5 digest is computed over the concatenated 32-bit sizes of the samples, in decoding order, in the track.
- 'trak': The MD5 digest is computed over the concatenated bytes of the samples, in decoding order, that are in the track.
- 'sgpd': The MD5 digest is computed over concatenated bytes of the samples, in decoding order, that are mapped to any group of the indicated type, and, if defined for the given grouping type, with the indicated `grouping_type_parameter`.
- `grouping_type`: If sample groups are used for the MD5 calculation, this field reflects the `grouping_type` of the sample groups considered for the MD5 calculations.

- `grouping_type_parameter`: If sample groups are used for the MD5 calculation, and a grouping type parameter is defined for the indicated grouping type, this field reflects the `grouping_type_parameter` to select, of the given sample group, to be considered for the MD5 calculations. When sample groups are used for the MD5 calculation and `grouping_type_parameter` is absent, version 0 of the `SampleToGroupBox` of the given sample group is referred.
- `num_entries` equal to 0 specifies that the MD5 checksum is derived from all samples mapped to the identified sample group. `num_entries` greater than 0 specifies that the MD5 checksum is derived from samples mapped to indicated sample group description indices.
- `group_description_index[i]`, when present, specifies that the MD5 checksum is derived from samples mapped to the sample group description index equal to `group_description_index[i]`.

9 Extensions to the ISO base media file format

9.1 General

The technologies in this clause are enhancements to the ISO base media file format (ISO/IEC 14496-12) and may be moved into that specification in the future.

9.2 ItemInfoEntry

`ItemInfoEntry` is specified in the ISO base media file format, and additionally the specifications of this clause apply.

The `flags` field of `ItemInfoEntry` with version greater than or equal to 2 is specified as follows:

(`flags & 1`) equal to 1 indicates that the item is not intended to be a part of the presentation. For example, when (`flags & 1`) is equal to 1 for an image item, the image item should not be displayed. (`flags & 1`) equal to 0 indicates that the item is intended to be a part of the presentation.

9.3 Item Properties Box

9.3.1 Definition

Box type: 'iprp'
 Container: `MetaBox('meta')`
 Mandatory: No
 Quantity: Zero or one

The `ItemPropertiesBox` enables the association of any item with an ordered set of item properties. Item properties are small data records.

The `ItemPropertiesBox` consists of two parts: `ItemPropertyContainerBox` that contains an implicitly indexed list of item properties, and one or more `ItemPropertyAssociation` boxes that associate items with item properties.

Each item property is a `Box` or `FullBox`. The `boxtype` of the item property specifies the property type. The `FreeSpaceBox` as defined in ISO/IEC 14496-12 may occur in the `ItemPropertyContainerBox`. It has no meaning, and, though it occupies an index value, should not be associated with any item.

Each property association may be marked as either essential or non-essential. A reader shall not process an item that is associated with a property that is not recognized or not supported by the reader.

and that is marked as essential to the item. A reader may ignore an associated item property that is marked non-essential to the item.

Specifications deriving from this document may specify property types and the respective item property box definitions as well as constraints and requirements for the property associations.

When defining item properties, it is recommended that they be small. When large data records need to be associated with an item, a separate item and item reference are more suitable.

Each ItemPropertyAssociation box shall be ordered by increasing item_ID, and there shall be at most one association box for each item_ID, in any ItemPropertyAssociation box. The version 0 should be used unless 32-bit item_ID values are needed; similarly, flags should be equal to 0 unless there are more than 127 properties in the ItemPropertyContainerBox. There shall be at most one ItemPropertyAssociation box with a given pair of values of version and flags.

9.3.2 Syntax

```
aligned(8) class ItemProperty(property_type)
    extends Box(property_type)
{
}
aligned(8) class ItemFullProperty(property_type, version, flags)
    extends FullBox(property_type, version, flags)
{
}
aligned(8) class ItemPropertyContainerBox
    extends Box('ipco')
{
    properties      ItemProperty() []; // boxes derived from
    // ItemProperty or ItemFullProperty, to fill box
}
aligned(8) class ItemPropertyAssociation
    extends FullBox('ipma', version, flags)
{
    unsigned int(32) entry_count;
    for(i = 0; i < entry_count; i++) {
        if (version < 1)
            unsigned int(16) item_ID;
        else
            unsigned int(32) item_ID;
        unsigned int(8) association_count;
        for (i=0; i<association_count; i++) {
            bit(1) essential;
            if (flags & 1)
                unsigned int(15) property_index;
            else
                unsigned int(7) property_index;
        }
    }
}
aligned(8) class ItemPropertiesBox
    extends Box('iprp') {
    ItemPropertyContainerBox property_container;
    ItemPropertyAssociation association[];
}
```

9.3.3 Semantics

item_ID identifies the item with which properties are associated.

essential when set to 1 indicates that the associated property is essential to the item, otherwise it is non-essential.

property_index is either 0 indicating that no property is associated (the essential indicator shall also be 0), or is the 1-based index of the associated property box in the ItemPropertyContainerBox contained in the same ItemPropertiesBox.

9.4 Entity grouping

9.4.1 General

An entity group is a grouping of items, which may also group tracks. The entities in an entity group share a particular characteristic or have a particular relationship, as indicated by the grouping type.

Entity groups are indicated in `GroupsListBox`. Entity groups specified in `GroupsListBox` of a file-level `MetaBox` refer to tracks or file-level items. Entity groups specified in `GroupsListBox` of a movie-level `MetaBox` refer to movie-level items. Entity groups specified in `GroupsListBox` of a track-level `MetaBox` refer to track-level items of that track.

`GroupsListBox` contains `EntityToGroupBoxes`, each specifying one entity group.

9.4.2 Groups List box

9.4.2.1 Definition

Box type:	'grpl'
Container:	<code>MetaBox</code> ('meta') that is not contained in <code>AdditionalMetadataContainerBox</code>
Mandatory:	No
Quantity:	Zero or one

The `GroupsListBox` includes the entity groups specified for the file. This box contains a set of full boxes, each called an `EntityToGroupBox`, with four-character codes denoting a defined grouping type.

The `GroupsListBox` shall not be present in `AdditionalMetadataContainerBox`.

When `GroupsListBox` is present in a file-level `MetaBox`, there shall be no `item_ID` value in `ItemInfoBox` in any file-level `MetaBox` that is equal to the `track_ID` value in any `TrackHeaderBox`.

9.4.2.2 Syntax

```
aligned(8) class GroupsListBox extends Box('grpl') {
}
```

9.4.3 Entity to Group box

9.4.3.1 Definition

Box type:	As specified below with the <code>grouping_type</code> value for the <code>EntityToGroupBox</code>
Container:	<code>GroupsListBox</code>
Mandatory:	No
Quantity:	Zero or one

The `EntityToGroupBox` specifies an entity group.

The box type (`grouping_type`) indicates the grouping type of the entity group. Each `grouping_type` code is associated with semantics that describe the grouping. The following `grouping_type` values are specified.

'altr': The items and tracks mapped to this grouping are alternatives to each other, and only one of them should be played (when the mapped items and tracks are part of the presentation, e.g. are displayable image items or tracks) or processed by other means (when the mapped items or tracks are not part of the presentation, e.g. are metadata). A player should select the first entity from the list of `entity_id` values that it can process (e.g. decode and play for mapped items and tracks that are part of the presentation) and that suits the application needs. Any `entity_id` value shall be mapped to only one grouping of type 'altr'. An alternate group of entities consists of those items and tracks that are mapped to the same entity group of type 'altr'.

'ster': The output images of the image items form a stereo pair suitable for displaying on a stereoscopic display. The entity group shall contain exactly two `entity_id` values that point to image items and shall contain no `entity_id` values that point to tracks. The first listed `entity_id` value (with `i` equal to 0) indicates the left view and the second `entity_id` value indicates the right view.

9.4.3.2 Syntax

```
aligned(8) class EntityToGroupBox(grouping_type, version, flags)
extends FullBox(grouping_type, version, flags) {
    unsigned int(32) group_id;
    unsigned int(32) num_entities_in_group;
    for(i=0; i<num_entities_in_group; i++)
        unsigned int(32) entity_id;
}
```

9.4.3.3 Semantics

`group_id` is a non-negative integer assigned to the particular grouping that shall not be equal to any `group_id` value of any other `EntityToGroupBox`, any `item_ID` value of the hierarchy level (file, movie, or track) that contains the `GroupsListBox`, or any `track_ID` value (when the `GroupsListBox` is contained in the file level).

`num_entities_in_group` specifies the number of `entity_id` values mapped to this entity group.

`entity_id` is resolved to an item, when an item with `item_ID` equal to `entity_id` is present in the hierarchy level (file, movie or track) that contains the `GroupsListBox`, or to a track, when a track with `track_ID` equal to `entity_id` is present and the `GroupsListBox` is contained in the file level.

9.5 Additional track references

The following additional `reference_type` values are specified to be used within `TrackReferenceBox`.

'thmb': This track contains thumbnails for the referenced track. A thumbnail track shall not be linked to another thumbnail track with the 'thmb' item reference.

'aux1': This track contains auxiliary media for the indicated track (e.g. depth map or alpha plane for video).

9.6 Repeating edits

9.6.1 Definition

To indicate that media is repeated, an enhancement to the `EditListBox` is used. The edit list maps the media timeline to the presentation timeline. The semantics of the flags field of the `EditListBox` as defined in ISO/IEC 14496-12 are extended as follows. When $(\text{flags} \ \& \ 1)$ is equal to 1, the entire edit list is repeated a sufficient number of times to equal the track duration.

NOTE The number of times the edit list is repeated does not need to be an integer. In other words, the last repetition of the edit list may be cut to match the track duration.

If the track duration is unknown/indefinite, the edit list is repeated indefinitely.

The full syntax is reproduced below, for convenience; however, the ISO base media file format specification is the normative specification of the `EditListBox` for all features except the flags.

9.6.2 Syntax

```
aligned(8) class EditListBox extends FullBox('elst', version, flags) {
    unsigned int(32) entry_count;
    for (i=1; i <= entry_count; i++) {
        if (version==1) {
            unsigned int(64) segment_duration;
            int(64) media_time;
        } else { // version==0
            unsigned int(32) segment_duration;
            int(32) media_time;
        }
        int(16) media_rate_integer;
        int(16) media_rate_fraction = 0;
    }
}
```

9.6.3 Semantics

flags specifies repetition of the edit list as follows. (flags & 1) equal to 0 specifies that the edit list is not repeated, while (flags & 1) equal to 1 specifies that the edit list is repeated. The values of flags greater than 1 are reserved. When an `EditListBox` indicates the playback of zero or one samples, (flags & 1) shall be equal to 0.

NOTE When the edit list is repeated, media at time 0 resulting from the edit list follows immediately the media having the largest time resulting from the edit list. In other words, the edit list is repeated seamlessly.

version is an integer that specifies the version of this box (0 or 1).

entry_count is an integer that gives the number of entries in the following table.

segment_duration is an integer that specifies the duration of this edit segment in units of the timescale in the `MovieHeaderBox`.

media_time is an integer containing the starting time within the media of this edit segment (in media time scale units, in composition time). If this field is set to -1, it is an empty edit. The last edit in a track shall never be an empty edit. Any difference between the duration in the `MovieHeaderBox` and the track's duration is expressed as an implicit empty edit at the end.

media_rate specifies the relative rate at which to play the media corresponding to this edit segment. If this value is 0, then the edit is specifying a 'dwell': the media at media-time is presented for the segment-duration. Otherwise, this field shall contain the value 1.

9.7 Sample-to-item sample grouping

9.7.1 Definition

Samples of a track can be linked to one or more metadata items using the sample-to-item sample grouping. The `MetaBox` containing the referred items is resolved as specified in the semantics below.

The sample-to-item sample grouping is allowed for any types of tracks, and its syntax and semantics are unchanged regardless of the track handler type.

In the absence of this sample group, the entire track-level `MetaBox`, if any, is applicable to every sample.

9.7.2 Syntax

```
class SampleToMetadataItemEntry()
extends SampleGroupDescriptionEntry('stmi') {
    unsigned int(32) meta_box_handler_type;
    unsigned int(32) num_items;
    for(i = 0; i < num_items; i++) {
        unsigned int(32) item_id[i];
    }
}
```

9.7.3 Semantics

`meta_box_handler_type` informs about the type of metadata schema used by the `MetaBox` which is referenced by the items in this sample group. When there are multiple `MetaBoxes` with the same handler types, the `MetaBox` referred to in this sample group entry is the first `MetaBox` fulfilling one of the following ordered constraints:

- a `MetaBox` included in the current track, not contained in `AdditionalMetadataContainerBox`, and with `handler_type` equal to `meta_box_handler_type`;
- a `MetaBox` included in the current track, contained in `AdditionalMetadataContainerBox`, and with `handler_type` equal to `meta_box_handler_type`;
- a `MetaBox` included in `MovieBox`, not contained in `AdditionalMetadataContainerBox`, and with `handler_type` equal to `meta_box_handler_type`;
- a `MetaBox` included in `MovieBox`, contained in `AdditionalMetadataContainerBox`, and with `handler_type` equal to `meta_box_handler_type`;
- a `MetaBox` included in the root level of the file, not contained in `AdditionalMetadataContainerBox`, and with `handler_type` equal to `meta_box_handler_type`;
- a `MetaBox` included in the root level of the file, contained in `AdditionalMetadataContainerBox`, and with `handler_type` equal to `meta_box_handler_type`.

`num_items` counts the number of items referenced by this sample group.

`item_id[i]` specifies the `item_ID` value of an item that applies to or is valid for the sample mapped to this sample group description entry.

10 Image File Format brands

10.1 General

Both structural and codec-specific brands are specified for the Image File Format.

Codec specific brand names enable file players to identify the required decoding capability by inspecting the `FileTypeBox` rather than in-depth investigation of all the `profile_idc` values included in the decoder configuration record.

When any of the brands specified in this document is in the `major_brand`, the `minor_version` shall be set to zero when writing the file and ignored by readers.

10.2 Image and image collection brands

10.2.1 'mifl' structural brand

10.2.1.1 Requirements on files

Files shall contain the brand 'mifl' in the compatible brands array of the FileTypeBox.

When the 'mifl' brand is present among the compatible brands array of the FileTypeBox, the file may be identified by MIME type defined in [Annex C](#). When this brand is the major brand, the defined file extension and MIME type should be used.

The following boxes are required in a file under the 'mifl' brand. The Version column in the following table lists the versions of the boxes allowed by this brand. Other versions of the boxes shall not be present.

NOTE A '-' in the Version column indicates that the box is a container box.

Hierarchy of boxes			Version	Box description
ftyp			—	file type and compatibility
meta			0	metadata
	hdlr		0	handler, declares the metadata (handler) type
	iloc		0, 1, 2	item location
	iinf		0, 1	item information
		infe	2, 3	item information entry
	pitm		0, 1	primary item reference
	iprp		—	item properties

Note particularly that the brand 'mifl' does not mandate a MovieBox ('moov').

10.2.1.2 Requirements on readers

Support for the following boxes is required under the 'mifl' brand. The Version column in the following table specifies the versions of the boxes that shall be supported by the readers of the 'mifl' brand.

Hierarchy of boxes			Version	Box description
ftyp			—	file type and compatibility
mdat			—	media data container
free			—	free space
skip			—	free space
meta			0	metadata
	hdlr		0	handler, declares the metadata (handler) type
	dinf		—	data information box, container
		dref	0	data reference box, declares source(s) of items
	iloc		0, 1, 2	item location
	iinf		0, 1	item information
		infe	2, 3	item information entry
	iref		0, 1	item reference box

Hierarchy of boxes			Version	Box description
ftyp			—	<i>file type and compatibility</i>
	pitm		0, 1	<i>primary item reference</i>
	idat		—	<i>item data</i>
	iprp		—	<i>item properties</i>

The boxes that declare item protection shall be recognized, and processed to the extent that readers shall determine when an item is protected. No support for any specific protection scheme is required. Readers should fail to decode items that are protected by an unrecognized scheme. The Version column in the following table specifies the versions of the boxes that shall be recognized by the readers of the 'mif1' brand.

Hierarchy of boxes			Version	Box description
ipro			0	<i>item protection</i>
	sinf		—	<i>protection scheme information box</i>
		frma	—	<i>original format box</i>
		schm	0	<i>scheme type box</i>
		schI	—	<i>scheme information box</i>

Readers shall support all the construction methods of the `ItemLocationBox` and the construction of the data of items from multiple extents.

Any reader conforming to the 'mif1' brand shall support displaying of at least the image included in the primary item provided that the reader supports the item type of that image and, when that image is described by a derived image item, the item types of the source image items of that image item.

Readers shall recognize the following item properties.

Four-character code	Name of the property
ispe	<i>image spatial extents</i>
pasp	<i>pixel aspect ratio</i>
colr	<i>colour information</i>
pixl	<i>pixel information</i>
rloc	<i>relative location</i>
auxC	<i>image properties for auxiliary images</i>
clap	<i>clean aperture</i>
irotr	<i>image rotation</i>
imir	<i>image mirroring</i>

10.3 Image sequence brands

10.3.1 'msf1' structural brand

10.3.1.1 Requirements on files

Files shall contain the brand 'msf1' in the compatible brands array of the `FileTypeBox`.

When the 'msf1' brand is present among the compatible brands array of the `FileTypeBox`, the file may be identified by MIME type defined in [Annex D](#). When this brand is the major brand, the defined file extension and MIME type should be used.

Additionally, the compatible brands array may contain codec specific brands, such as those described in C.4.2. The codec specific brand announces to the reader, the facilities required from the reader to decode the coded media stream properly.

At least one track of handler type 'pict', as defined in 7.2, is required.

It is required that 'iso8' is present among the compatible brands array.

10.3.1.2 Requirements on readers

Readers shall support tracks with handler type 'pict', as defined in 7.2.

Structures required by the 'iso8' brand shall be supported.

The EditListBox repetition, as specified in 9.6, shall be supported.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-12:2017

Annex A (normative)

Storage of externally specified metadata

A.1 General

This annex specifies the format to store metadata complying with Exif (JEITA CP-3451B), XMP (ISO 16684-1), or MPEG-7 (ISO/IEC 15938-3) in files conforming to the Image File Format. When Exif, XMP, or MPEG-7 metadata is associated with items or tracks conforming to the Image File Format, the metadata shall follow the specifications of this annex. However, it is not required for a reader conforming to this document to understand Exif, XMP, or MPEG-7 metadata.

A.2 Exif

A.2.1 Untimed Exif metadata

Exif data is stored always as an 'Exif block', with the following structure:

```
aligned(8) class ExifDataBlock() {
    unsigned int(32) exif_tiff_header_offset;
    unsigned int(8) exif_payload[];
}
```

`exif_tiff_header_offset` is an offset in bytes from the first byte of `exif_payload` to the first byte of the TIFF Header of the Exif metadata, as specified in JEITA CP-3451B. If the TIFF Header is the first byte of the payload, the value is 0. Otherwise, it is a positive number skipping any other bytes before the TIFF Header (e.g. `exif_payload` is formatted as specified for the DCF thumbnail file in JEITA CP-3461B).

`exif_payload` is a variable sized array of bytes holding the Exif compliant metadata to be parsed by the reader. This is compliant with JEITA CP-3451B or JEITA CP-3461B and shall have as part of it a TIFF Header with referenced Image File Directories (IFDs). There may be additional bytes before or after this Exif data, but the all data shall be contained in the size indicated by the item size. `exif_payload` should not contain fields that use file-absolute offsets, because it is allowed to modify a file so that the location of item data is changed.

When untimed Exif metadata is stored as a metadata item the `item_type` value shall be 'Exif'.

A.2.2 Exif metadata in tracks

When Exif metadata is stored in a metadata track, the sample entry type is 'Exif'. The Exif metadata track is linked via a 'cdsc' track reference to the track it describes.

Exif metadata that is true for the entire track may be stored in a `MetaBox` in the `TrackBox`, in one or more items of type 'Exif'.

It is not required that every sample be a 'sync sample'. The metadata that applies to the corresponding time interval of the linked track is formed by the union of the following:

- for sync samples, the metadata in the `MetaBox` and the metadata that is the metadata sample data;
- for non-sync samples, the metadata in the `MetaBox`, the metadata that is the sample data of the preceding sync sample, and the metadata that is the sample data.

When such a union is formed, any duplicate metadata items are replaced, in the order given.

Each sample is precisely an ExifDataBlock.

A.3 XMP metadata

For image items, XMP metadata shall be stored as an item of `item_type` value 'mime' and content type 'application/rdf+xml'. The body of the item shall be a valid XMP document, in XML form.

For image sequences which use the track structure, XMP metadata that is true for the entire track can be stored in an item of type 'mime' and content type 'application/rdf+xml' embedded in a track level `MetaBox`.

When XMP data is carried in a metadata track, the track handler is 'meta' and shall use the XML metadata sample entry using the XMP namespace. The XMP metadata track is linked via a 'cdsc' track reference to the track it describes.

A.4 MPEG-7 metadata

MPEG-7 metadata is stored as an item of `item_type` value 'mime' and using the XML document MIME type.

The body of the item shall be an MPEG-7 document, in XML form.

The storage of MPEG-7 in tracks is defined in ISO/IEC 14496-14.

Annex B (normative)

HEVC Image File Format

B.1 General

This annex derives a format to encapsulate HEVC-coded images, image collections, and image sequences from the Image File Format specified above. HEVC-specific brands for a single image and an image collection as well as image sequences are specified in [B.4](#).

B.2 HEVC images and image collections

B.2.1 General

[B.2](#) specifies requirements for files containing HEVC-coded image items. When a brand specified in [B.4.1](#) is among the compatible brands of a file, the requirements specified in [B.2](#) shall be obeyed.

The specifications of [Clause 6](#) apply.

B.2.2 Image data

B.2.2.1 Definition

B.2.2.1.1 General

There shall be no inter prediction between HEVC image items, except for the case of an external base layer (see below). If inter prediction between images exist, the HEVC pictures shall be stored according to [B.3](#).

HEVCItemData is structurally identical to the syntax defined in ISO/IEC 14496-15 for an HEVC sample. HEVCItemData shall not contain any extractors or aggregators defined in ISO/IEC 14496-15.

NOTE Functionality similar to sharing NAL units through extractors between samples of different tracks can be achieved in image items through the use of extents.

B.2.2.1.2 Image item of type 'hvc1'

An item of type 'hvc1' consists of the NAL units of an HEVC bitstream that are length-delimited as specified below, and the bitstream contains exactly one access unit.

NAL units with `nuh_layer_id` greater than 0 may be present in items of type 'hvc1'. Readers shall ignore NAL units with `nuh_layer_id` greater than 0 in an item of type 'hvc1'.

NOTE The base layer picture of HEVC items of type 'hvc1' may be an IDR, CRA or BLA picture as defined ISO/IEC 23008-2.

B.2.2.1.3 Image item of type 'lhv1'

An item of type 'lhv1' consists of the NAL units of an HEVC bitstream that are length-delimited as specified below and the bitstream contains exactly one access unit.

NOTE An item of type 'lhv1' consists of an initial IRAP access unit as defined ISO/IEC 23008-2, can contain more than one coded picture, and contains at most one coded picture with any specific value of `nuh_layer_id`.

All image items of type 'lhv1' shall have an associated property called the 'oinf' property. The 'oinf' property provides a summary of the high-level characteristics of the bitstream containing the image item, similar to the 'oinf' sample grouping of ISO/IEC 14496-15.

All image items of type 'lhv1' shall have an associated TargetOlsProperty item property. TargetOlsProperty provides the output layer set index to be used as input for the decoding process of coded image item.

The 'lhv1' image item shall include the layers that are included in the layer set identified by the associated TargetOlsProperty and may include other layers too.

When LayerSelectorProperty is associated with an image item of type 'lhv1', it shall contain layer_id that is among the nuh_layer_id values of the output layers of the output layer set identified by TargetOlsProperty associated with the same image item.

When there is an 'exbl' item reference from an image item of type 'lhv1' to another image item, the decoded pixel array of that other image item serves as the decoded picture with nuh_layer_id equal to 0 for the decoding of the 'lhv1' image item. Moreover, the variable BIlrapPicFlag, as specified by ISO/IEC 23008-2, is set equal to 1 and nal_unit_type for the decoded picture with nuh_layer_id equal to 0 is set equal to IDR_W_RADL, as specified by ISO/IEC 23008-2, for the decoding of the 'lhv1' image item.

B.2.2.2 Syntax

```
aligned(8) class HEVCItemData
{
    for (i=0; i<item_size; ) // item_size from summing the extents
                           // in the ItemLocationBox
    {
        unsigned int((DecoderConfigurationRecord.LengthSizeMinusOne+1)*8)
            NALUnitLength;
        bit(NALUnitLength * 8) NALUnit;
        i += (DecoderConfigurationRecord.LengthSizeMinusOne+1) +
            NALUnitLength;
    }
}
```

B.2.2.3 Semantics

In the syntax above, the following applies.

- The value of item_size is equal to the sum of the extent_length values of each extent of the item, as specified in the ItemLocationBox.
- DecoderConfigurationRecord indicates the record in the associated configuration initialization property.

NALUnitLength indicates the size of a NAL unit measured in bytes. The length field includes the size of both the two-byte NAL header and the RBSP payload but does not include the length field itself.

NALUnit contains a single NAL unit. The syntax of a NAL unit is specified in ISO/IEC 23008-2 and includes both the two-byte NAL unit header and the variable-length RBSP payload.

B.2.3 Image properties

B.2.3.1 HEVC configuration item property

Box type:	'hvcC'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for an image item of type 'hvc1'
Quantity (per item):	One for an image item of type 'hvc1'

Each HEVC image item of type 'hvc1' shall have an associated property that is exactly identical to the HEVCConfigurationBox as defined in ISO/IEC 14496-15.

essential shall be equal to 1 for an 'hvcC' item property associated with an image item of type 'hvc1'.

B.2.3.2 Sub-sample item property

Box type:	'subs'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	Zero or more for an HEVC image item

Sub-sample information for HEVC coded images may be given using an associated property that is exactly identical to SubSampleInformationBox for HEVC as defined in ISO/IEC 14496-12 and ISO/IEC 14496-15. The entry_count field of the SubSampleInformationBox shall be equal to 1, and the sample_delta field of the SubSampleInformationBox shall be equal to 0.

Zero or more properties of type 'subs' may be linked to the same HEVC image item.

B.2.3.3 Layered HEVC configuration item property

Box type:	'lhvc'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for an image item of type 'lhv1'
Quantity (per item):	One for an image item of type 'lhv1'

Each HEVC image item of type 'lhv1' shall have an associated property that is exactly identical to the LHEVCConfigurationBox as defined in ISO/IEC 14496-15.

essential shall be equal to 1 for an 'lhvc' item property associated with an image item of type 'lhv1'.

B.2.3.4 Operating points information property

B.2.3.4.1 Definition

Box type:	'oinf'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for an image item of type 'lhv1'
Quantity (per item):	One for an image item of type 'lhv1'

The operating points information property 'oinf' is similar to the operating points information sample group specified in ISO/IEC 14496-15 but applies to image items.

Image items originating from the same bitstream shall be associated with the same 'oinf' property. The 'oinf' property informs about the different operating points provided by a bitstream and their constitution. Each operating point is related to an output layer set and a combination of a profile, level and tier. For each operating point, the 'oinf' property provides the minimum and maximum width and height of output pictures, the chroma format, and the bit-depth. `TargetOlsProperty` associated with an image item provides the output layer set index that can be used to select which operating-point-specific information of the 'oinf' property applies to the image item. The property also provides the dependency information between layers and the scalability types in the bitstream.

B.2.3.4.2 Syntax

```
aligned(8) class OperatingPointsInformationProperty
extends ItemFullProperty('oinf', version = 0, flags = 0){
    OperatingPointsRecord; // specified in ISO/IEC 14496-15
}
```

B.2.3.4.3 Semantics

The semantics of `OperatingPointsRecord` are specified in ISO/IEC 14496-15. When included in `OperatingPointsInformationProperty`, the values of the syntax elements of `OperatingPointsRecord` are constrained as follows.

`frame_rate_info_flag` shall be equal to 0. Consequently, `avgFrameRate` and `constantFrameRate` are not present and their semantics are not specified.

`bit_rate_info_flag` shall be equal to 0. Consequently, `maxBitRate` and `avgBitRate` are not present and their semantics are not specified.

B.2.3.5 Target output layer set property

B.2.3.5.1 Definition

Box type:	'tols'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	Yes, for an image item of type 'lhv1'
Quantity (per item):	One for an image item of type 'lhv1', zero otherwise

`TargetOlsProperty` provides the output layer set index to be used as input for the decoding process of coded image item.

essential shall be equal to 1 for an 'tols' item property.

B.2.3.5.2 Syntax

```
aligned(8) class TargetOlsProperty
extends ItemFullProperty('tols', version = 0, flags = 0){
    unsigned int(16) target_ols_index;
}
```

B.2.3.5.3 Semantics

`target_ols_idx` provides the output layer set index to be provided to the L-HEVC decoding process as the value of `TargetOlsIdx` variable, specified in clause F.8 of HEVC.

NOTE Output layer set index equal to 0 indicates an output layer set consisting of the base layer only. It is discouraged to have 'lhv1' image items with an associated `TargetOlsProperty` with `target_ols_idx` equal to 0 present in files. Instead, the inclusion of the respective 'hvc1' image items is encouraged.

B.2.4 HEVC auxiliary images

B.2.4.1 General

The following URNs are specified for `aux_type` of `AuxiliaryTypeProperty` in this annex.

urn:mpeg:hevc:2015:auxid:xxx. This URN points to ISO/IEC 23008-2:2015, Table F.2. The xxx in the URN string is the decimal string representation of an integer identifying the auxiliary image type that is equal to the `AuxId` value specified in ISO/IEC 23008-2:2015, Table F.2.

NOTE 1 ISO/IEC 23008-2:2015, Table F.2 specifies auxiliary picture types for auxiliary pictures within auxiliary picture layers. `urn:mpeg:hevc:2015:auxid:xxx` may be used auxiliary pictures stored as items with an item reference of type 'aux1' to another image.

An HEVC coded auxiliary image uses the `item_type` value 'hvc1' or 'lhv1'.

`HEVCAuxConfigSubType` structure specified below replaces the `aux_subtype` byte array in `AuxiliaryTypeProperty`.

NOTE 2 The auxiliary image is treated like a master image. Among other things, this means that the initialization data for HEVC coded auxiliary images is provided by a property. The same image item may be associated with an `AuxiliaryTypeProperty` and `HEVCConfigurationBox`.

`HevcAuxConfigSubType` can include SEI messages that are specific to the auxiliary image type and can provide information relevant for interpreting the auxiliary image.

B.2.4.2 Syntax

```
aligned(8) class HEVCAuxConfigSubType {
    unsigned int(32) sei_msg_len;
    for (i=0; i<sei_msg_len; ){
        unsigned int((DecoderConfigurationRecord.LengthSizeMinusOne+1)*8)
        nalu_len;
        bit(nalu_len * 8) nal_unit;
        i += (DecoderConfigurationRecord.LengthSizeMinusOne+1) + nalu_len;
    }
}
```

B.2.4.3 Semantics

In the syntax above, `DecoderConfigurationRecord` indicates the record in the associated configuration item property.

`sei_msg_len`: the sum of the sizes of zero or more SEI NAL units, preceded by their length.

nal_len: The size of a NAL unit measured in bytes. The length field includes the size of both the two-byte NAL unit header and the RBSP payload but does not include the length field itself.

nal_unit: A single SEI NAL unit including both the two-byte NAL unit header and the variable length encapsulated byte stream payload.

The label `urn:mpeg:hevc:2015:auxid:1` indicates an alpha plane. It is recommended to provide, within `nal_unit`, the alpha channel information SEI message specified in ISO/IEC 23008-2, when `aux_type` is equal to `urn:mpeg:hevc:2015:auxid:1`. `urn:mpeg:hevc:2015:auxid:2` indicates a depth image. It is recommended to provide, within `nal_unit`, the depth representation information SEI message specified in ISO/IEC 23008-2, when `aux_type` is equal to `urn:mpeg:hevc:2015:auxid:2`.

B.2.5 HEVC tile Items

ISO/IEC 23008-2 allows partitioning of a picture into tiles. ISO/IEC 23008-2 includes the exact definition and properties of a tile, while some properties are informatively repeated in the following: A tile is a rectangular portion of the picture within a particular tile column and a particular tile row. A tile column is a rectangular region of the picture having a height equal to the height of the picture. A tile row is a rectangular region of the picture having a width equal to the width of the picture. A tile can be decoded independently of other tiles, as there are no intra prediction or entropy decoding dependencies between tiles.

NOTE 1 Although tiles may be independently decoded, a tile may require adjacent tiles to be decoded for an exact reconstruction of pixel data, because of HEVC loop filtering. It is the task of the file reader/decoder to identify whether HEVC loop filtering is disabled across tiles when processing HEVC tile items.

An HEVC tile item shall be stored as an item of type `'hvt1'`, and formatted as a series of NAL units preceded by length fields, as defined in [B.2.2.1.3](#).

An HEVC tile item consists of the following NAL units in decoding order specified by ISO/IEC 23008-2:

- a set of VCL NAL units containing one or more tiles, as defined in ISO/IEC 23008-2, such that the tiles contained in the set of VCL NAL units represent a rectangular array of pixels;
- associated non-VCL NAL units (if any) for the set of VCL NAL units, as defined in ISO/IEC 23008-2.

NOTE 2 Typically, if the HEVC tile item consists of a single tile, only the slice(s) used to code this tile will be found in the HEVC tile item.

The VCL NAL units of an HEVC tile item shall form a rectangular array of pixels without holes. Tiles shall appear in raster-scan order in an HEVC tile item.

The slice type of VCL NAL units in HEVC tile items shall be equal to I (standing for intra coded slices).

NOTE 3 HEVC tile items can be included in a file to allow fast data fetching without analysing NAL unit layout of the image. For finer-grain and/or more generic indication of tiles, the sub-sample information specified in [B.2.3.2](#) can be used. For instance, the sub-sample information is suitable to indicate the tiles that are contained within one VCL NAL unit.

When a VCL NAL unit included in an HEVC tile item is a dependent slice segment of a particular slice and the independent slice segment of that particular slice is not included in the same HEVC tile item, there shall be an item reference of type `'dpnd'` identifying the HEVC tile item in the `from_item_ID` field and the HEVC tile item containing the independent slice segment of that particular slice in the `to_item_ID` field. All dependent slice segments of an HEVC tile item identified by `from_item_ID` of any `'dpnd'` item reference shall be parts of the same slice. There shall be exactly one independent slice segment in the tile HEVC tile item identified by `to_item_ID` of any `'dpnd'` item reference.

NOTE 4 If a single slice with a single segment NAL unit carries a non-rectangular set of tiles, the resulting set of tiles cannot be expressed as an HEVC tile item.

NOTE 5 If a slice carries a non-rectangular set of tiles with one slice segment per tile, an HEVC tile item can typically be formed from a subset of the tiles in the slice, possibly together with tiles from another slice. For example, if an HEVC picture has a 2×2 regular tiling with one slice for the first three tiles with one slice segment per tile and one slice for the last tile, HEVC tile items can be used and the items corresponding to the second and third HEVC tile items shall have an item reference of type 'dpnd' to the first HEVC tile item.

Each HEVC tile item shall be associated with one HEVCConfigurationBox, one ImageSpatialExtentsProperty and one RelativeLocationProperty.

The HEVCConfigurationBox shall contain all parameter sets required for decoding the tiles present in the HEVC tile item.

The RelativeLocationProperty shall indicate the position of the HEVC tile item within the respective HEVC image item.

NOTE 6 The respective HEVC image item of each HEVC tile item is identified as documented in 6.5.7.

The image_width and image_height of the ImageSpatialExtentsProperty shall be set according to the width and height of the HEVC tile item.

B.3 HEVC image sequences

B.3.1 General

B.3 specifies requirements for all files containing one or more HEVC-coded image sequence tracks. When a brand specified in B.4.2 is among the compatible brands of a file, the requirements specified in B.3 shall be obeyed.

The specifications of Clause 7 apply.

B.3.2 Derivation from ISO/IEC 14496-12 and ISO/IEC 14496-15

The sample entry of type 'hvc1', 'hvc2', or 'lhv1' shall be used for an image sequence track coded with HEVC, as specified in ISO/IEC 14496-15.

The HEVCSampleEntry or LHEVCSampleEntry shall be used as specified in ISO/IEC 14496-15.

NOTE As specified in 7.2.3.1, a CodingConstraintsBox is required to be present in the HEVCSampleEntry and LHEVCSampleEntry, in addition to the boxes required by ISO/IEC 14496-15 to be contained in the HEVCSampleEntry and LHEVCSampleEntry, respectively.

For a track containing an HEVC image sequence, either all samples shall be sync samples or the all_ref_pics_intra field in the CodingConstraintsBox specified in 7.2.3 shall be set to one.

B.3.3 Auxiliary HEVC image sequence tracks

The SEI messages for the auxiliary channel follow the same principle as any other SEI message for the sample entry, i.e. they may be included in the decoder configuration record of the sample entry types specified for HEVC or its multi-layer extensions in ISO/IEC 14496-15. When aux_track_type is equal to 'urn:mpeg:hevc:2015:auxid:xxx' (where xxx is a positive integer), as specified in B.2.3.3, an HEVC SEI message describing the auxiliary image sequence should be included in the sample entry.

B.4 HEVC-specific brands

B.4.1 HEVC image and image collection brands

B.4.1.1 General

The brands 'heic' and 'heix' are specified in the following subclauses. Unless otherwise stated, the specifications apply to both 'heic' and 'heix' brands.

A coded image item is specified to conform to the 'heic' brand when all of the following constraints are true.

- The item has type 'hvc1' and conforms to the specifications in [B.2](#).
- The item is not associated with any other types of essential item properties than 'hvcC', 'irot', 'clap', and 'imir'.

The content of the item conforms to the Main profile or the Main Still Picture profile of HEVC.

A coded image item is specified to conform to the 'heix' brand when all of the following constraints are true.

- The item has type 'hvc1' and conforms to the specifications in [B.2](#).
- The item is not associated with any other types of essential item properties than 'hvcC', 'irot', 'clap', and 'imir'.
- The content of the item conforms to the Main 10 profile or any of the format range extensions profiles of HEVC.

B.4.1.2 Requirements on files

Files shall include 'mif1' among the compatible brands and hence conform to the specifications in [10.2.1.1](#). Additionally, files shall comply with the specifications in [B.2](#).

The files conforming to the 'heic' and 'heix' brands shall additionally be constrained as follows.

Each file including 'heic' as a compatible brand shall contain an item that is present in the file, is either the primary item or any item from the alternate group containing the primary item, and fulfils one of the following constraints.

- The item is a coded image item conforming to the 'heic' brand as specified in [B.4.1.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item that is present in the file and conforms to the 'heic' brand as specified in [B.4.1.1](#).

Each file including 'heix' as a compatible brand shall contain an item that is present in the file, is either the primary item or any item from the alternate group containing the primary item, and fulfils one of the following constraints.

- The item is a coded image item conforming to the 'heix' brand as specified in [B.4.1.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item that is present in the file and conforms to the 'heix' brand as specified in [B.4.1.1](#).

B.4.1.3 Requirements on readers

The requirements on readers specified in [10.2.1.2](#) shall be supported.

Readers conforming to the 'heic' brand shall support displaying an item that is either the primary item or any item from the alternate group containing the primary item and fulfils one of the following constraints.

- The item is a coded image item conforming to the 'heic' brand as specified in [B.4.1.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'heic' brand as specified in [B.4.1.1](#).

Readers conforming to the 'heix' brand shall support displaying an item that is either the primary item or any item from the alternate group containing the primary item and fulfils one of the following constraints.

- The item is a coded image item conforming to the 'heix' brand as specified in [B.4.1.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'heix' brand as specified in [B.4.1.1](#).

Readers conforming to the 'heic' brand or the 'heix' brand are recommended but not required to decode all levels as specified by ISO/IEC 23008-2.

File readers should support displaying of an image with opacity information specified by an associated auxiliary image of `aux_type` equal to `urn:mpeg:hevc:2015:auxid:1`.

B.4.2 HEVC image sequence brands

B.4.2.1 General

The brands 'hevc' and 'hevX' are specified in the following subclauses. Unless otherwise stated, the specifications apply to both 'hevc' and 'hevX' brands.

B.4.2.2 Requirements on files

Files shall include 'msf1' among the compatible brands and hence conform to the specifications in [10.3.1.1](#). Additionally, files shall comply with the specifications in [B.3](#). `Track_enabled` shall be equal to 1 and `Track_in_movie` shall be equal to 1 for at least one image sequence track conforming to with the specifications in [B.3](#).

When the 'hevc' brand is among the compatible brands, there shall be an image sequence track with 'hvc1' sample entry type, `Track_enabled` equal to 1, `Track_in_movie` equal to 1, and each sample entry having a `data_reference_index` value such that it is mapped to a `DataEntryBox` with `(entry_flags & 1)` equal to 1, for which `general_profile_idc` is equal to 1 or `(general_profile_compatibility_flags & 2(32-1))` is greater than 0.

NOTE 1 In other words, when the 'hevc' brand is among the compatible brands, at least one viewable image sequence track is an HEVC image sequence track and contains a bitstream conforming to the Main profile of HEVC.

When the 'hevX' brand is among the compatible brands, there shall be an image sequence track with 'hvc1' sample entry type, `Track_enabled` equal to 1 and `Track_in_movie` equal to 1, and each sample entry having a `data_reference_index` value such that it is mapped to a `DataEntryBox` with `(entry_flags & 1)` equal to 1, for which either one of the following is true:

- `general_profile_idc` is equal to 2 or `(general_profile_compatibility_flags & 2(32-2))` is greater than 0;
- `general_profile_idc` is equal to 4 or `(general_profile_compatibility_flags & 2(32-4))` is greater than 0.

NOTE 2 In other words, when the 'hevX' brand is among the compatible brands, at least one viewable image sequence track is an HEVC image sequence track and contains a bitstream conforming to the Main 10 profile or any of the format range extensions profiles of HEVC.

B.4.2.3 Requirements on readers

The requirements on readers specified in [10.3.1.2](#) shall be supported.

Readers for the 'hevc' brand shall be able to display an image sequence track with 'hvc1' sample entry type, `Track_enabled` equal to 1 and `Track_in_movie` equal to 1, for which `general_profile_idc` is equal to 1 or $(\text{general_profile_compatibility_flags} \& 2^{(32-1)})$ is greater than 0.

Readers for the 'hev1' brand shall be able to display an image sequence track with 'hvc1' sample entry type, `Track_enabled` equal to 1 and `Track_in_movie` equal to 1, for which either one of the following is true:

- `general_profile_idc` is equal to 2 or $(\text{general_profile_compatibility_flags} \& 2^{(32-2)})$ is greater than 0;
- `general_profile_idc` is equal to 4 or $(\text{general_profile_compatibility_flags} \& 2^{(32-4)})$ is greater than 0.

Readers shall support all values allowed by [7.2.1](#) for the matrix syntax element of the `TrackHeaderBox` and shall obey the `CleanApertureBox` of the visual sample entry when displaying an image sequence track with 'hvc1' sample entry.

NOTE In other words, readers are required to support rotation by 0°, 90°, 180°, and 270° and mirroring, as controlled by the matrix syntax element, as well as cropping, as controlled by the `CleanApertureBox`.

Displaying of an image sequence track with opacity information specified by an associated auxiliary track of `aux_track_type` equal to `urn:mpeg:hevc:2015:auxid:1`, as specified in [B.3.3](#), should be supported.

B.4.3 L-HEVC image and image collection brands

B.4.3.1 General

The brands 'heim' and 'heis' are specified in the following subclauses. Unless otherwise stated, the specifications apply to both 'heim' and 'heis' brands.

A coded image item is specified to conform to the 'heim' brand when all of the following constraints are true.

- The item has type 'lhv1' and conforms to the specifications in [B.2](#).
- The item is not associated with any other types of essential item properties than 'lhvC', 'irot', 'clap', 'imir', 'lsel', and 'tols'.
- Each layer of the item conforms to the Main profile or the Multiview Main profile of HEVC.

A coded image item is specified to conform to the 'heis' brand when all of the following constraints are true.

- The item has type 'lhv1' and conforms to the specifications in [B.2](#).
- The item is not associated with any other types of essential item properties than 'lhvC', 'irot', 'clap', 'imir', 'lsel', and 'tols'.
- Each layer of the item conforms to the Main profile, Main 10 profile, the Scalable Main profile, or the Scalable Main 10 profile of HEVC.

B.4.3.2 Requirements on files

Files shall include 'mif1' among the compatible brands and hence conform to the specifications in [10.2.1.1](#). Additionally, files shall comply with the specifications in [B.2](#).

The files conforming to the 'heim' and 'heis' brands shall additionally be constrained as follows:

Files including 'heim' as a compatible brand shall contain an item that is present in the file, is either the primary item or any item from the alternate group containing the primary item, and fulfills one of the following constraints.

- The item is a coded image item conforming to the 'heim' brand as specified in [B.4.3.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'heim' brand as specified in [B.4.3.1](#).

Files including 'heis' as a compatible brand shall contain an item that is present in the file is either the primary item or any item from the alternate group containing the primary item, and fulfills one of the following constraints.

- The item is a coded image item conforming to the 'heis' brand as specified in [B.4.3.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'heis' brand as specified in [B.4.3.1](#).

B.4.3.3 Requirements on readers

The requirements on readers specified in [10.2.1.2](#) shall be supported.

Readers conforming to the 'heim' brand shall support displaying an item that is either the primary item or any item from the alternate group containing the primary item and fulfils one of the following constraints.

- The item is a coded image item conforming to the 'heim' brand as specified in [B.4.3.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'heim' brand as specified in [B.4.3.1](#).

Readers conforming to the 'heis' brand shall support displaying an item that is either the primary item or any item from the alternate group containing the primary item and fulfils one of the following constraints.

- The item is a coded image item conforming to the 'heis' brand as specified in [B.4.3.1](#).
- The item is a crop-rotate-mirror derived image item, and each source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'heis' brand as specified in [B.4.3.1](#).

Readers conforming to the 'heim' brand or the 'heis' brand are recommended but not required to decode all levels as specified by ISO/IEC 23008-2.

File readers should support displaying of an image with opacity information specified by an associated auxiliary image of `aux_type` equal to `urn:mpeg:hevc:2015:auxid:1`.

B.4.4 L-HEVC image sequence brands

B.4.4.1 General

The brands 'hevm' and 'hevs' are specified in the following subclauses. Unless otherwise stated, the specifications apply to both 'hevm' and 'hevs' brands.

B.4.4.2 Requirements on files

Files shall include 'msf1' among the compatible brands and hence conform to the specifications in [10.3.1.1](#). Additionally, files shall comply with the specifications in [B.3](#). `Track_enabled` shall be equal

to 1 and `Track_in_movie` shall be equal to 1 for at least one image sequence track conforming to the specifications in [B.3](#).

When the 'hevm' brand is among the compatible brands, there shall be an image sequence track with 'hvc1' or 'hvc2' sample entry type with an L-HEVC configuration record included, `Track_enabled` equal to 1, `Track_in_movie` equal to 1, and each sample entry having a `data_reference_index` value such that it is mapped to a `DataEntryBox` with `(entry_flags & 1)` equal to 1. Moreover, each layer in the reconstructed access units of the image sequence track shall conform to the Main profile or the Multiview Main profile.

When the 'hevs' brand is among the compatible brands, there shall be an image sequence track with 'hvc1' or 'hvc2' sample entry type with an L-HEVC configuration record included, `Track_enabled` equal to 1 and `Track_in_movie` equal to 1, and each sample entry having a `data_reference_index` value such that it is mapped to a `DataEntryBox` with `(entry_flags & 1)` equal to 1. Moreover, each layer in the reconstructed access units of the image sequence track shall conform to the Main profile, Main 10 profile, Scalable Main profile, or the Scalable Main 10 profile.

B.4.4.3 Requirements on readers

The requirements on readers specified in [10.3.1.2](#) shall be supported.

Readers for the 'hevm' brand shall be able to display an image sequence track with 'hvc1' or 'hvc2' sample entry type with an L-HEVC configuration record included, `Track_enabled` equal to 1 and `Track_in_movie` equal to 1, for which each layer in the reconstructed access units of the image sequence track conforms to the Main profile or the Multiview Main profile.

Readers for the 'hevs' brand shall be able to display an image sequence track with 'hvc1' or 'hvc2' sample entry type with an L-HEVC configuration record included, `Track_enabled` equal to 1 and `Track_in_movie` equal to 1, for which each layer in the reconstructed access units of the image sequence track conforms to the Main profile, Main 10 profile, Scalable Main profile, or the Scalable Main 10 profile.

Readers shall support all values allowed by [7.2.1](#) for the matrix syntax element of the `TrackHeaderBox` and shall obey the `CleanApertureBox` of the visual sample entry when displaying an image sequence track with 'hvc1' or 'hvc2' sample entry.

NOTE In other words, readers are required to support rotation by 0°, 90°, 180°, and 270° and mirroring, as controlled by the matrix syntax element, as well as cropping, as controlled by the `CleanApertureBox`.

Displaying of an image sequence track with opacity information specified by an associated auxiliary track of `aux_track_type` equal to `urn:mpeg:hevc:2015:auxid:1`, as specified in [B.3.3](#), should be supported.

Annex C (normative)

High efficiency image file MIME type registration

C.1 General

The file extension and MIME type of a file in deriving from the ISO base media file format usually reflect the major brand in the `FileTypeBox`. When the major brand indicates a brand related to [Clause 6](#) or [B.2](#) (single image and image collection), the MIME type defined here should be used. When such a brand is a compatible brand, this MIME type may also be used.

C.2 Registration

MIME media type name: image

MIME subtype name: heif, heic

The semantics of the subtypes are as follows:

heif: High efficiency image file containing one or more image items using any coding format

heic: High efficiency image file conforming to the requirements for the 'heic', 'heix', 'heim', or 'heis' brand (and hence containing one or more HEVC coded image items)

The use of subtype values is constrained as follows:

The MIME subtype name MAY be 'heic' only if the file conforms to the requirements of the 'heic', 'heix', 'heim', or 'heis' brand, and contains at least one of those brands as a compatible brand. Otherwise, the MIME subtype name SHALL be 'heif' or the MIME subtype specified for a derived format to which the file conforms.

Required parameters: none

Optional parameters:

profiles: Specified by RFC 6381 and its successors.

codecs: As specified for files derived from ISO/IEC 14496-12 and derived specifications. The specifications of the ISO base media file format name space of RFC 6381 apply for the codecs parameter. For HEVC, the format of a list item included in the value of the codecs parameter is specified in ISO/IEC 14496-15.

itemtypes: One or more comma-separated item descriptions.

Each item description corresponds to an image item included in the file. An item description SHOULD be present for the primary item of the file and MAY be present for other image items of the file.

Each item description starts with an item type string and is followed by a plus-separated ('+') list of zero or more item property strings.

An item type string starts with the four-character

item_type value of the item and MAY be followed by zero or more dot-separated ('.') qualifiers specified below.

When the item type is a four-character code of a coded image, it MAY be followed by a dot-separated ('.') value, as specified for the codecs parameter of the ISO base media file format name space in RFC 6381. For the item type 'hvc1', the value after the '.' is the profile-tier-level value as specified in ISO/IEC 14496-15.

When the item type is a four-character code of a derived image item, it MAY be followed by a dot-separated ('.') pixel count value that is the positive decimal integer indicating the number of pixels that is required for the input images of the derived image item and the reconstructed image itself. For the item type 'hvc1', the pixel count value SHALL be present for an item description, when that pixel count value is greater than twice the largest pixel count inferred from the profile-tier-level value of any coded image of the same item description list.

An item property string consists of the box-type of an item property marked as essential. The list of the item property strings SHALL indicate the entire set of item properties that are marked as essential. The item property strings SHALL appear in the order they are associated with the image item in the file.

lhvcpt1: syntax and semantics are identical to those specified for lhvcpt1 optional MIME parameter in ISO/IEC 14496-15 for the L-HEVC sample entry types.

dependencies: a list of comma-separated URLs from the DataReferenceBoxes in the top-level MetaBox and all tracks. The DataReferenceBoxes indicating a reference to the same file as the container file MUST NOT be listed. The URLs SHOULD be relative whenever possible. Note that the URLs are often, but not required to be, relative, and that some characters in URLs may require escaping in some situations.

Encoding considerations: as for video/mp4

Security considerations: See section 5 of RFC 4337

Interoperability considerations: -

Published specification: ISO/IEC 23008-12

Applications: Multimedia

Additional information:

Magic number(s): none
File extension(s): heif (for subtype heif),
heic (for subtype heic)
Macintosh File Type Code(s): None

Person to contact for info: David Singer, singer@apple.com

Intended usage: Common

Author/Change controller: David Singer, ISO/IEC/JTC1/SC29/WG11 file format chair

C.3 Examples (informative)

Content-Type: image/heic; itemtypes=hvc1.A1.80.L93.B0+hvcC+irot

An image rotating by a multiple of 90 degrees an associated image that is a non-frame-packed HEVC Main profile image at the Main tier, level 3.1.

Content-Type: image/heic; itemtypes=hvc1.A1.80.L93.B0+hvcC,iden+irot

Two items, one of which is a derived image item obtained by rotation, and the other is a non-frame-packed HEVC Main profile image at the Main tier, level 3.1.

Content-Type: image/heic; itemtypes=hvc1.A1.80.L93.B0+hvcC; profiles=heic

An image file where the primary item of the file is a coded image that may or may not be associated with transformative item properties that are marked as non-essential. The coded image is a progressive, non-frame-packed HEVC Main profile image at the Main tier, level 3.1.

Content-Type: image/heic; itemtypes=grid.3686400,hvc1.A1.80.L93.B0+hvcC,hvc1.A1.80.L93.B0+hvcC

A grid of two images of size 1280x720, and two non-frame-packed HEVC Main profile image at the Main tier, level 3.1.

Annex D (normative)

High efficiency image sequence file MIME type registration

D.1 General

The file extension and MIME type of a file deriving from the ISO base media file format usually reflect the major brand in the `FileTypeBox`. When the major brand indicates a brand related to [Clause 7](#) or [B.3](#) (image sequences), the MIME type defined here should be used. When such a brand is a compatible brand, this MIME type may also be used.

D.2 Registration

MIME media type name: image

MIME subtype name: heif-sequence, heic-sequence

The semantics of the subtypes are as follows:

heif-sequence: High efficiency image file containing one or more image sequences using any coding format

heic-sequence: High efficiency image file containing one or more HEVC coded image sequences

The use of subtype values is constrained as follows:
The MIME subtype name MAY be 'heic-sequence' only if the file conforms to the requirements of the 'hevc', 'hev1', 'hevm', or 'hevs' brand, and contains that brand as a compatible brand. Otherwise, the MIME subtype name SHALL be 'heif-sequence' or the MIME subtype specified for a derived format to which the file conforms.

Required parameters: none

Optional parameters:

profiles: Specified by RFC 6381 and its successors.

codecs: As specified for files derived from ISO/IEC 14496-12 and derived specifications. The specifications of the ISO base media file format name space of RFC 6381 apply for the codecs parameter. For HEVC, the format of a list item included in the value of the codecs parameter is specified in ISO/IEC 14496-15.

When the codecs parameter is present, the first list item SHOULD represent a track having the handler type 'pict'. Other list items represent other tracks.

lhevcp1: syntax and semantics are identical to those specified for lhevcp1 optional MIME parameter in ISO/IEC 14496-15 for the L-HEVC sample entry types.

dependencies:
as specified for the dependencies optional MIME parameter of image/heif and image/heic MIME types.

Encoding considerations: as for video/mp4

Security considerations: See section 5 of RFC 4337

Interoperability considerations: A reference implementation exists within the ISO/IEC 14496 community.

Published specification: ISO/IEC 23008-12

Applications: Multimedia

Additional information:

 Magic number(s): none

 File extension(s): heifs (for subtype heif-sequence), heics
 (for subtype heic-sequence)

 Macintosh File Type Code(s): None

Person to contact for info: David Singer, singer@apple.com

Intended usage: Common

Author/Change controller: David Singer, ISO/IEC/JTC1/SC29/WG11 file format chair

IECNORM.COM : Click to view the full PDF of ISO/IEC 23008-12:2017

Annex E (normative)

AVC in the Image File Format

E.1 Overview

This annex derives a format to encapsulate AVC-coded images, image collections, and image sequences from the Image File Format specified above. AVC-specific brands for a single image and an image collection as well as image sequences are specified in [E.4](#).

E.2 AVC images and image collections

E.2.1 General

[E.3](#) specifies requirements for files containing AVC-coded image items. When a brand specified in [E.4.1](#) is among the compatible brands of a file, the requirements specified in [E.3](#) and its subclauses shall be obeyed.

The specifications of [Clause 6](#) and its subclauses apply.

E.2.2 Image data

E.2.2.1 Definition

An item of type 'avc1' consists of the NAL units of an AVC bitstream that are length-delimited as specified below, and the bitstream contains exactly one access unit.

NOTE 1 AVC items are normally IDR pictures as defined ISO/IEC 14496-10.

There shall be no inter prediction between AVC image items. If inter prediction between images exist, the AVC pictures shall be stored according to [E.4](#).

The AVCIItemData is structurally identical to the syntax defined in ISO/IEC 14496-15 for an AVC sample. AVCIItemData shall not contain any extractors or aggregators defined in ISO/IEC 14496-15.

NOTE 2 Functionality similar to sharing NAL units through extractors between samples of different tracks can be achieved in image items through the use of extents.

E.2.2.2 Syntax

```
aligned(8) class AVCIItemData
{
    for (i=0; i<item_size; ) // item_size from summing the extents
                           // in the ItemLocationBox
    {
        unsigned int((DecoderConfigurationRecord.LengthSizeMinusOne+1)*8)
            NALUnitLength;
        bit(NALUnitLength * 8) NALUnit;
        i += (DecoderConfigurationRecord.LengthSizeMinusOne+1) +
            NALUnitLength;
    }
}
```

E.2.2.3 Semantics

In the syntax above, the following applies.

- The value of `item_size` is equal to the sum of the `extent_length` values of each extent of the item, as specified in the `ItemLocationBox`.
- `DecoderConfigurationRecord` indicates the record in the associated configuration initialization property.

`NALUnitLength` indicates the size of a NAL unit measured in bytes. The length field includes the size of both the two-byte NAL header and the RBSP payload but does not include the length field itself.

`NALUnit` contains a single NAL unit. The syntax of a NAL unit is specified in ISO/IEC 14496-10 and includes both the two-byte NAL unit header and the variable-length RBSP payload.

E.2.3 AVC configuration item property

Box type:	'avcC'
Property type:	Descriptive item property
Container:	<code>ItemPropertyContainerBox</code>
Mandatory (per item):	Yes, for an image item of type 'avc1'
Quantity (per item):	One for an image item of type 'avc1'

Each AVC image item shall have an associated property that is exactly identical to the `AVCConfigurationBox` as defined in ISO/IEC 14496-15.

`essential` shall be equal to 1 for an 'avcC' item property associated with an image item of type 'avc1'.

E.2.4 Sub-sample item property

Box type:	'subs'
Property type:	Descriptive item property
Container:	<code>ItemPropertyContainerBox</code>
Mandatory (per item):	No
Quantity (per item):	Zero or more for an image item of type 'avc1'

Sub-sample information for AVC coded images may be given using an associated property that is exactly identical to `SubSampleInformationBox` for AVC as defined in ISO/IEC 14496-12 and ISO/IEC 14496-15. The `entry_count` field of the `SubSampleInformationBox` shall be equal to 1, and the `sample_delta` field of the `SubSampleInformationBox` shall be equal to 0.

Zero or more properties of type 'subs' may be linked to the same item of type 'avc1'.

E.2.5 AVC auxiliary images

The URNs specified for HEVC in [B.2.5](#) may also be used with AVC.

An AVC coded auxiliary image uses the `item_type` value 'avc1'.