INTERNATIONAL STANDARD

ISO/IEC 23008-12

Second edition
2022-09

Information technology — MPEG systems technologies —

Part 12: Image File Format

Technologies de l'information — Technologies des systèmes MPEG —

Partie 12: Format de fichier d'image

Reference number
ISO/IEC 23008-12:2022(E)

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

Published in Switzerland

Contents

Page

Foreword

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

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

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

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

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

This second edition cancels and replaces the first edition (ISO/IEC 23008-12:2017), which has been technically revised. It also incorporates the Amendments ISO/IEC 23008-12:2017/Amd.1:2020, ISO/IEC 23008-12:2017/DAmd.2:2019 and the Technical Corrigendum ISO/IEC 23008-12:2017/Cor. 1:2020.

The main changes are as follows:

- addition of Annexes L to N;
- support for predictive image coding, bursts, bracketing and region annotations.

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

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

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). It is suggested that HEIF be pronounced "heaff" (like heath with an ff ending). When the requirements of the HEVC-specific brands are applied, the file format can be referred to as the HEVC Image File Format.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

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

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.
- 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.
- 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.
- 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.
- Annex E specifies the format for encapsulating AVC-coded images, image collections, and image sequences according to the Image File Format.

- 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.

- Annex H specifies the format for encapsulating JPEG-coded images, image collections, and image sequences according to the Image File Format.

- Annex I outlines 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.

- Annex L specifies the format for encapsulating VVC-coded images, image collections, and image sequences according to the Image File Format. Annex L also specifies VVC-specific brands for a single image and an image collection as well as image sequences. Requirements on both files and file readers are specified.

- Annex M specifies the format for encapsulating EVC-coded images, image collections, and image sequences according to the Image File Format. Annex M also specifies EVC-specific brands for a single image and an image collection as well as image sequences. Requirements on both files and file readers are specified.

- Annex N contains considerations on privacy and security relating to the use of the Image File Format.

Information technology — MPEG systems technologies —

Part 12:
Image File Format

1 Scope

This document specifies the Image File Format, an interoperable storage format for a single image, a collection of images, and sequences of images.

The format defined in this document is built on tools defined in ISO/IEC 14496-12 and enables the interchange, editing, and display of images, as well as the carriage of metadata associated with those images. The Image File Format defines structures used to contain metadata, how to link that metadata to the images, and defines how metadata of certain forms is carried.

This document also specifies brands for the storage of images and image sequences conforming to High Efficiency Video Coding (HEVC), Advanced Video Coding (AVC), JPEG, Versatile Video Coding (VVC) and Essential Video Coding (EVC).

NOTE The storage of HEVC, AVC, VVC and EVC video sequences is out of scope and is provided in ISO/IEC 14496-15.

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

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

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

ISO/IEC 23090-3, *Information technology — Coded representation of immersive media — Part 3: Versatile video coding*

ISO/IEC 23094-1, *Information technology — General video coding — Part 1: Essential 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, ISO/IEC 14496-15 and the following apply.

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

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

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

3.1.1
alternate group
group of *entities* (3.1.11) 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 (3.1.18) that is associated with the *item property* (3.1.28) through the `ItemPropertiesBox`

3.1.3
auxiliary image
image (3.1.16) that may not be intended to be displayed but provides supplemental information, such as transparency data, complementing a respective *master image* (3.1.29)

3.1.4
coded image
coded representation of an *image* (3.1.16)

3.1.5
coded image item
item (3.1.27) 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* (3.1.12) than `'irot'`, `'clap'`, and `'imir'`

3.1.7
derived image
representation of an *image* (3.1.16) as an *operation* (3.1.33) on other images

3.1.8
derived image item
item (3.1.27) whose data is a *derived image* (3.1.7)

3.1.9
derived region item
item (3.1.27) whose data is a representation of the shape, position and size of a region within an *image item* (3.1.18), with which it is associated via item reference, as an *operation* (3.1.33) on other *region items* (3.1.39)

3.1.10
descriptive item property
item property (3.1.28) that describes rather than transforms the associated item

3.1.11
entity
item or track

3.1.12
essential item property
item property (3.1.28) that readers are required to process

3.1.13
HEVC image item
image item (3.1.18) of type `'hvc1'` or `'lhv1'`

3.1.14
hidden image
image (3.1.16) that is not intended to be displayed

3.1.15
hidden sample
sample that is not intended to be displayed

3.1.16
image
one or more arrays of pixels of different colour components described by an *image item* (3.1.18) or a sample

3.1.17
image collection
set of *images* (3.1.16) stored as *items* (3.1.27)of a single file according to this document

3.1.18
image item
coded image item (3.1.5) or *derived image item* (3.1.8)

3.1.19
Image File Format
file format specified by this document

3.1.20
image property
item property (3.1.28) for an *image item* (3.1.18)

3.1.21
image sequence
sequence of *coded images* (3.1.4) which may be associated with advisory timing and in which images may use *inter prediction* (3.1.24)

3.1.22
image sequence track
track that contains an *image sequence* (3.1.21)

3.1.23
input image
image (3.1.16) that is used as an input for the *operation* (3.1.33) of the *derived image item* (3.1.8)

3.1.24
inter prediction
prediction derived in a manner that is dependent on data elements (e.g. sample values or motion vectors) of *images* (3.1.16) other than the current image

3.1.25
intra coding
coding of an *image* (3.1.16) that may use *intra prediction* (3.1.26) and does not use *inter prediction* (3.1.24)

3.1.26
intra prediction
prediction derived from only data elements (e.g. sample values) of the same decoded image

3.1.27
item
data that does not require timed processing, as opposed to sample data, and is described by the boxes contained in a `MetaBox`

3.1.28
item property
descriptive or transformative information about an *item* (3.1.27) as stored in the item properties array

3.1.29
master image
image that is stored as an *item* (3.1.27) and is not an *auxiliary image* (3.1.3) or a thumbnail image

3.1.30
master image sequence
sequence of images that is stored as an *image sequence track* (3.1.22) and is not an *auxiliary image* (3.1.3) sequence or a thumbnail *image sequence* (3.1.21)

3.1.31
metadata item
item (3.1.27) containing metadata that may for example describe an *image item* (3.1.18)

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.

3.1.32
non-essential item property
item property (3.1.28) that readers are allowed to ignore

3.1.33
operation
for a *derived image item* (3.1.8), manipulation, identified by the item type, that produces a *reconstructed image* (3.1.37) from a set of *input images* (3.1.23). For a *derived region item* (3.1.9), manipulation, identified by the item type, that produces the shape, position and size of regions of an *image* (3.1.16) from a set of regions from input *region items* (3.1.39).

3.1.34
output image
image (3.1.16) that results when the reconstructed image of the *image item* (3.1.18) is transformed according to the *transformative item properties* (3.1.42) of the image item

3.1.35
pre-derived coded image
coded image (3.1.4) that has been derived from one or more other images

3.1.36
predictively coded image item
image item (3.1.18) that has a decoding dependency to one or more other *coded image items* (3.1.5)

3.1.37
reconstructed image
image (3.1.16) that results when the *coded image item* (3.1.5) is decoded or when the *operation* (3.1.33) of the *derived image item* (3.1.8), if any, is applied

3.1.38
reference image
image (3.1.16) that may be used as a reference for *inter prediction* (3.1.24) of another image

3.1.39
region item
item (3.1.27) whose data defines the shape, position and size of a region within an *image item* (3.1.18) with which it is associated via item reference

3.1.40
source image item
image item (3.1.18) 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.

3.1.41
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

3.1.42
transformative item property
item property (3.1.28) 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 degrees of a reconstructed image of an image item.

3.1.43
unique ID
identifier for either an item, an entity group or a track that fulfils the requirements of the `'unif'` brand

Note 1 to entry: Requirements on the `'unif'` brand are specified in ISO/IEC 14496-12.

3.1.44
visual context
visual rendering surface such as a screen buffer, which may already contain visual material, and onto which an *image* (3.1.16) can be rendered

3.2 Abbreviated terms

ASCII	American Standard Code for Information Interchange
AVC	Advanced Video Coding (Rec. ITU-T H.264 | ISO/IEC 14496-10)
DCF	Design rule for Camera File system (JEITA CP-3461)
EVC	Essential Video Coding (ISO/IEC 23094-1)
Exif	Exchangeable Image File Format (JEITA CP-3451)
HDR	high dynamic range
HEIF	High Efficiency Image File Format (this document: ISO/IEC 23008-12)
HEVC	High Efficiency Video Coding (Rec. ITU-T H.265 | ISO/IEC 23008-2)
MD5	Message Digest algorithm 5
MIME	Multi-purpose Internet Mail Extensions
NAL	network abstraction layer

PPS	picture parameter set
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
VVC	Versatile Video Coding (Rec. ITU-T H.266 | ISO/IEC 23090-3)
XML	Extensible Markup Language
XMP	Extensible Metadata Platform

4 Overview

This document specifies the following two forms of storage:

a) the storage of a single coded image or a collection of coded images, possibly with derived images; coded images are normally independently coded except when the 'pred' brand is signalled. In such latter case, coded images may be independently coded or may have been coded with inter prediction;

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), 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').

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 14496-12:2022, 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 ISO/IEC 14496-12:2022, Clause 4.

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. When a brand specified in 10.2 is among the compatible brands of a file, the requirements specified in Clause 6 shall be applied.

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. This interleave can be used to allow progressive display.

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 specified for the item type of the image item;
- 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. 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 `'auxl'` 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.

6.4.9 Predictively coded image items

Predictively coded image items have a decoding dependency to one or more other coded image items. An example for such an image item could be a P frame stored as an image item in a burst entity group that has IPPP... structure, with the P frames dependent only on the preceding I frames.

Capability to have predictively coded image items has certain benefits especially in content re-editing and cover image selection:

- Image sequences can be converted to image items with no transcoding.
- Any sample of an image sequence track can be selected as a cover image. The cover image does not need to be intra-coded.
- Devices that do not have a video or image encoder are capable of updating the cover image of a file containing an image sequence track.
- Storage efficiency is further achieved by re-using the predictively coded picture rather than re-encoding it as I frame and storing as an additional image item. Moreover, image quality degradation is also avoided.
- Re-encoding might not be allowed or preferred by the copyright owner. Predictively coded image items avoid the need of re-encoding of any image from an image sequence track.

Predictively coded image items are linked to the coded image items they directly and indirectly depend on by item references of type `'pred'`. The list of referenced items in item references of type `'pred'` shall indicate the decoding order. When concatenated, the encoded media data of items with `item_ID` equal to `to_item_ID` for all values of `j` from 0 to `reference_count` − 1, inclusive, in increasing order of `j`, followed by the item with `item_ID` equal to `from_item_ID` shall form a bitstream that conforms to the decoder configuration item property of the predictively coded image item.

In order to decode the predictively coded image item, there shall be no other decoding dependencies other than the image items referenced by item references of type `'pred'`.

The predictively coded image item shall be associated with exactly one `RequiredReferenceTypesProperty` containing one reference type with the value `'pred'`.

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.

When unique IDs are used, an `item_ID` value in the `ItemPropertyAssociationBox` is resolved to an item identifier whenever the embedding `MetaBox` contains an item with such identifier, and is resolved to an entity group identifier otherwise.

Properties may be associated with an entity group, but only when explicitly stated in their specification. In such case, properties apply to the entity group as a whole, and not individually to each entity within the group.

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.

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 6.3.

image_height specifies the height of the reconstructed image in pixels, as specified in 6.3.

NOTE Item properties, such as decoder configuration or layer selection, can affect the reconstructed image. As a consequence, the width and height of the reconstructed image depend on the presence and content of such properties.

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

When no pixel aspect ratio 'pasp' descriptive item property is associated with an image item, readers shall infer pixel aspect ratio 1:1 to apply for the image item.

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 for a given value of colour_type

The definition of Colour Information provided in ISO/IEC 14496-12 applies.

In addition, following definitions specific to the use of colour information in image items also applies:

- When generating an image item from the content of a visual track, the order of ColourInformationBoxes in the VisualSampleEntry should be preserved in the ItemPropertyAssociationBox(es). Similarly, when creating a visual track from an image item, the order of boxes should be preserved.

While in a visual track, the order of boxes may be important per ISO/IEC 14496-12, in this specification, the order is not relevant. Colour information with different values of colour_type are intended for different purposes. Colour information with a value of colour_type set to 'nclx' is intended to be used for some processing such as colour conversion or image derivation, while colour information carrying an ICC profile is intended to be used for processes such as display matching. As in ISO/IEC 14496-12, ICC profiles may be provided in two properties under the codes 'prof' and 'rICC' to provide unrestricted and restricted ICC profiles respectively.

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: 'lsel'

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 'lsel' item property shall be associated with the image item. Otherwise, the 'lsel' item property shall not be associated with an image item.

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

The 'lsel' 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('lsel') {
   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 how the mirroring is performed: 0 indicates that the top and bottom parts of the image are exchanged (i.e. the mirroring is vertical); 1 specifies that the left and right parts are exchanged (i.e. the mirroring is horizontal).

NOTE In Exif, orientation tag can be used to signal mirroring operations. Exif orientation tag 4 corresponds to axis = 0 of ImageMirror, and Exif orientation tag 2 corresponds to axis = 1 accordingly.

6.5.13 Image scaling

6.5.13.1 Definition

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

The image scaling 'iscl' transformative item property scales an input image.

The input image is the output of the previous transformative item property, if any, or the reconstructed image of the associated image item.

The width and height of the input image (call those input_width and input_height) are resized to a target width and height, in pixels, respectively equal to ceil((input_width * target_width_numerator) / target_width_denominator) and ceil((input_height * target_height_numerator) / target_height_ denominator), where ceil() is the ceiling function. The scaling of the input image applies to both width and height separately. The fraction may or may not be in reduced terms.

NOTE 1 Formulas above use a floating-point division, not an integer division.

NOTE 2 When the input image is the reconstructed image of the associated image item, input_width and input_height, respectively, are equal to image_width and image_height declared in the ImageSpatialExtentsProperty associated with this image item. Otherwise, input_width and input_height are equal to the width and height of the output of the previous transformative item property.

6.5.13.2 Syntax

```
aligned(8) class ImageScaling
extends ItemFullProperty('iscl', version = 0, flags = 0) {
   unsigned int(16) target_width_numerator;
   unsigned int(16) target_width_denominator;
   unsigned int(16) target_height_numerator;
   unsigned int(16) target_height_denominator;
}
```

6.5.13.3 Semantics

target_width_numerator specifies the numerator of the scaling ratio for the resized image in the horizontal dimension. The value 0 shall not be used.

target_width_denominator specifies the denominator of the scaling ratio for the resized image in the horizontal dimension. The value 0 shall not be used.

target_height_numerator specifies the numerator of the scaling ratio for the resized image in the vertical dimension. The value 0 shall not be used.

target_height_denominator specifies the denominator of the scaling ratio for the resized image in the vertical dimension. The value 0 shall not be used.

6.5.14 Content light level

6.5.14.1 Definition

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

The content light level item property provides information about the light level in the content.

6.5.14.2 Syntax

The content light level 'clli' descriptive item property has the same syntax as the ContentLightLevelBox as defined in ISO/IEC 14496-12.

6.5.14.3 Semantics

The semantics of the syntax elements within the content light level 'clli' item property are the same as those specified for the syntax elements of ContentLightLevelBox as defined in ISO/IEC 14496-12.

6.5.15 Mastering display colour volume

6.5.15.1 Definition

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

This property provides information about the colour primaries, white point, and mastering luminance in the content.

6.5.15.2 Syntax

This property has the same syntax as the `MasteringDisplayColourVolumeBox` as defined in ISO/IEC 14496-12.

6.5.15.3 Semantics

The semantics of the syntax elements within this property are the same as those specified for the syntax elements of `MasteringDisplayColourVolumeBox` as defined in ISO/IEC 14496-12.

6.5.16 Content colour volume

6.5.16.1 Definition

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

This property describes the colour volume characteristics of the associated pictures.

6.5.16.2 Syntax

This property has the same syntax as the `ContentColourVolumeBox` as defined in ISO/IEC 14496-12.

6.5.16.3 Semantics

The semantics of the syntax elements within this property are the same as those specified for the syntax elements of `ContentColourVolumeBox` as defined in ISO/IEC 14496-12.

6.5.17 Required reference types

6.5.17.1 Definition

Box type:	`'rref'`

Property type: Descriptive item property

Container: ItemPropertyContainerBox

Mandatory (per item): Yes, for a predictively coded image item. No, otherwise.

Quantity (per item): At most one

The RequiredReferenceTypesProperty descriptive item property lists the item reference types that a reader shall understand and process to decode the associated image item. The respective essential flag shall be equal to 1 in ItemPropertyAssociationBox.

NOTE In the absence of this property, required reference types are not explicitly listed, but can still exist.

6.5.17.2 Syntax

```
aligned(8) class RequiredReferenceTypesProperty
extends ItemFullProperty('rref', version = 0, flags = 0){
   unsigned int(8) reference_type_count;
   for (i=0; i< reference_type_count; i++) {
      unsigned int(32) reference_type[i];
   }
}
```

6.5.17.3 Semantics

reference_type_count indicates the number of reference types that are required to understand and process to decode the associated image item.

reference_type[i] indicates a reference type that is required to understand and process to decode the the associated image item.

6.5.18 Creation time information

6.5.18.1 Definition

Box type: 'crtt'

Property type: Descriptive item property

Container: ItemPropertyContainerBox

Mandatory (per associated item_ID): No

Quantity (per associated item_ID): At most one

The CreationTimeProperty documents the creation time of the associated item or group of entities.

6.5.18.2 Syntax

```
aligned(8) class CreationTimeProperty
extends ItemFullProperty('crtt', version = 0, flags = 0) {
   unsigned int(64)  creation_time;
}
```

6.5.18.3 Semantics

creation_time is an integer that declares the creation time of the item or group of entities (in microseconds since midnight, Jan. 1, 1904, in UTC time).

6.5.19 Modification time information

6.5.19.1 Definition

Box type:	`'mdft'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per associated item_ID):	No
Quantity (per associated item_ID):	At most one

The `ModificationTimeProperty` documents the last modification time of the associated item or group of entities.

6.5.19.2 Syntax

```
aligned(8) class ModificationTimeProperty
extends ItemFullProperty('mdft', version = 0, flags = 0) {
   unsigned int(64)  modification_time;
}
```

6.5.19.3 Semantics

`modification_time` is an integer that declares the most recent time the item or group of entities was modified (in microseconds since midnight, Jan. 1, 1904, in UTC time).

6.5.20 User description

6.5.20.1 Definition

Box type:	`'udes'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per associated item_ID):	No
Quantity (per associated item_ID):	Zero or more

The `UserDescriptionProperty` permits the association of item(s) or entity group(s) with a user-defined name, description and tags; there may be multiple such properties, which shall have different language codes.

When several instances of `UserDescriptionProperty` are associated with the same item or entity group, they represent alternatives possibly expressed in different languages and a reader should choose the most appropriate. At most one `UserDescriptionProperty` with the same `alt_lang` value should apply to the same item or entity group.

6.5.20.2 Syntax

```
aligned(8) class UserDescriptionProperty
extends ItemFullProperty('udes', version = 0, flags = 0){
   utf8string lang;
   utf8string name;
   utf8string description;
   utf8string tags;
}
```

6.5.20.3 Semantics

lang is a character string containing an RFC 5646 compliant language tag string, such as "en-US", "fr-FR", or "zh-CN“, representing the language of the text contained in name, description and tags. When lang is empty, the language is unknown/undefined.

name is a null-terminated UTF-8 character string containing human readable name for the item or group of entities. If not present (an empty string is supplied) no name is provided.

description is a null-terminated UTF-8 character string containing human readable description of the item or group of entities. If not present (an empty string is supplied) no description is provided.

tags is a null-terminated UTF-8 character string containing comma-separated user-defined tags related to the item(s). If not present (an empty string is supplied) no tags is provided.

6.5.21 Accessibility text

6.5.21.1 Definition

Box Type:	'altt'
Property Type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per an item):	No
Quantity (per an item):	Zero or more

The AccessibilityTextProperty contains a string suitable to be used as an alternate text for an image if the image cannot be displayed, similarly to alt text in HTML. The language used by the alternate text is represented by a language tag string compliant with IETF RFC 5646.

When several instances of AccessibilityTextProperty are associated with the same item, they represent alternatives possibly expressed in different languages and a reader should choose the most appropriate. At most one AccessibilityTextProperty with the same alt_lang value should apply to the same item.

6.5.21.2 Syntax

```
aligned(8) class AccessibilityTextProperty
extends ItemFullProperty('altt', version = 0, flags = 0) {
   utf8string alt_text;
   utf8string alt_lang;
}
```

6.5.21.3 Semantics

alt_text is a character string suitable to be used as an alternate text for an image if the image cannot be displayed, similarly to alt text in HTML.

alt_lang is a character string containing an RFC 5646 compliant language tag string, such as "en-US", "fr-FR", or "zh-CN“, representing the language of the text contained in alt_text. When alt_lang is empty, the language is unknown/undefined.

6.5.22 Auto Exposure Information

6.5.22.1 Definition

Box type:	'aebr'

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

The auto exposure descriptive item property defines the exposure variation of the associated image item relatively to the camera settings.

It is used to specify the properties of an image item included in an `'aebr'` entity group as specified in 6.8.6.

6.5.22.2 Syntax

```
aligned(8) class AutoExposureProperty
extends ItemFullProperty('aebr', version = 0, flags = 0) {
   int(8) exposure_step;
   int(8) exposure_numerator;
}
```

6.5.22.3 Semantics

`exposure_step` is an integer value that specifies the increment steps used during the exposure bracketing. When equals to 1, a full stop increment is used, when equals to 2, a half stop increment is used, when equals to 3, a third stop increment is used, and when equals to 4, a quarter stop increment is used. Other values are reserved.

`exposure_numerator` is an integer value specifying the exposure numerator used to compute the exposure value stop of the item.

The exposure value variation of the associated image item compared to the camera settings is expressed as a number of stops, and is computed as `exposure_numerator/exposure_step`.

6.5.23 White balance information

6.5.23.1 Definition

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

The white balance descriptive item property defines the white balance compensation on Blue/Amber bias and/or Magenta/Green bias applied to the associated image item relatively to the camera settings.

It is used to specify the properties of an image item included in a `'wbbr'` entity group as specified in 6.8.6.

6.5.23.2 Syntax

```
aligned(8) class WhiteBalanceProperty
extends ItemFullProperty('wbbr', version = 0, flags = 0) {
   unsigned int(16) blue_amber;
   int(8) green_magenta;
}
```

6.5.23.3 Semantics

`blue_amber` is an unsigned integer indicating the color temperature component of the white balance in Kelvin.

`green_magenta` is a signed integer indicating the color deviation component of white balance in unit of 1/100 Duv (distance to the blackbody locus).

NOTE A Duv of 0 indicates a light source that is neutral. A negative Duv indicates a magenta color shift, while a positive Duv indicates a green color shift.

6.5.24 Focus information

6.5.24.1 Definition

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

The focus descriptive item property defines the focus variation of the associated image item relatively to the camera settings.

It is used to specify the properties of an image item included in an `'fobr'` entity group as specified in 6.8.6.

6.5.24.2 Syntax

```
aligned(8) class FocusProperty
extends ItemFullProperty('fobr', version = 0, flags = 0) {
   unsigned int(16) focus_distance_numerator;
   unsigned int(16) focus_distance_denominator;
}
```

6.5.24.3 Semantics

The focus distance is expressed in meter as the ratio of `focus_distance_numerator` and `focus_distance_denominator`. Focus at infinity is expressed as division by zero, i.e. `focus_distance_denominator` is equal to 0 and `focus_distance_numerator` should be equal to 0.

6.5.25 Flash exposure information

6.5.25.1 Definition

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

The flash exposure descriptive item property defines the flash exposure variation of the associated image item relatively to the camera settings.

It is used to specify the properties of an image item included in an 'afbr' entity group as specified in Clause 6.8.6.

6.5.25.2 Syntax

```
aligned(8) class FlashExposureProperty
extends ItemFullProperty('afbr', version = 0, flags = 0) {
   int(8) flash_exposure_numerator;
   int(8) flash_exposure_denominator;
}
```

6.5.25.3 Semantics

flash_exposure_numerator and flash_exposure_denominator are integers representing the flash exposure value of the sample expressed in a number of f-stops as the ratio of flash_exposure_numerator and flash_exposure_denominator.

6.5.26 Depth of field information

6.5.26.1 Definition

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

The depth of field descriptive item property defines the depth of field variation of the associated image item relatively to the camera settings. It is expressed as an aperture change.

It is used to specify the properties of an image item included in a 'dobr' entity group as specified in 6.8.6.

6.5.26.2 Syntax

```
aligned(8) class DepthOfFieldProperty
extends ItemFullProperty('dobr', version = 0, flags = 0) {
   int(8) f_stop_numerator;
   int(8) f_stop_denominator;
}
```

6.5.26.3 Semantics

The depth of field variation is expressed as an aperture change in a number of stops, and is computed as f_stop_numerator/f_stop_denominator.

6.5.27 Panorama information

6.5.27.1 Definition

Box type:	'pano'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per associated item_ID):	No

Quantity (per associated item_ID): At most one

The panorama descriptive item property defines the characteristics associated with a panorama defined by a 'pano' entity group.

This item property should only be associated with an entity group with grouping_type equal to 'pano'.

6.5.27.2 Syntax

```
aligned(8) class PanoramaProperty
extends ItemFullProperty('pano', version = 0, flags = 0) {
   unsigned int(8) panorama_direction;
   if (panorama_direction >= 4 && panorama_direction <= 5) { // grid
      unsigned int(8) rows_minus_one;
      unsigned int(8) columns_minus_one;
   }
}
```

6.5.27.3 Semantics

panorama_direction is an unsigned integer signalling the type of panorama used and the scanning order of input images in the panorama:

- 0: left-to-right horizontal panorama
- 1: right-to-left horizontal panorama
- 2: bottom-to-top vertical panorama
- 3: top-to-bottom vertical panorama
- 4: grid panorama in raster scan order, i.e. rows and columns are organised from left-to-right and top-to-bottom starting from the top-left corner.
- 5: grid panorama in continuous order, i.e. starting from the top-left corner, the first row is organised from left-to-right, then the second row is organised from right-to-left, the third row is organised from left-to-right and so on.

other values are undefined.

rows_minus_one is an unsigned integer that specifies the number of rows in the grid minus one.

columns_minus_one is an unsigned integer that specifies the number of columns in the grid minus one.

6.5.28 Sub-sample information

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

Sub-sample information for coded images may be given using an associated item property that is exactly identical to SubSampleInformationBox as defined in ISO/IEC 14496-12 and for the coding format of the associated coded image item.

NOTE The definition of sub-sample information for a NAL unit-based coding format is as defined for this coding format in 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 SubSampleInformationBoxes may be associated with the same coded image item.

When more than one SubSampleInformationBox is associated with the same coded image item, the value of flags shall differ in each of these SubSampleInformationBoxes. The semantics of flags, if any, shall be supplied for a given coding format. If flags have no semantics for a given coding format, the flags shall be 0

6.5.29 Target output layer set

6.5.29.1 Definition

Box type:	'tols'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No, unless otherwise specified by the coding format
Quantity (per item):	Zero or one for a coded image item

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

essential shall be equal to 1 for a TargetOlsProperty.

6.5.29.2 Syntax

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

6.5.29.3 Semantics

target_ols_idx is the output layer set index to be provided to the decoding process. The semantics of target_ols_idx are specific to the coding format of the image item and are therefore defined for each coding format that requires its use.

6.5.30 Wipe transition effect

6.5.30.1 Definition

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

The WipeTransitionEffectProperty documents a suggested wipe transition effect to apply between the display of two consecutive items of a slideshow entity group.

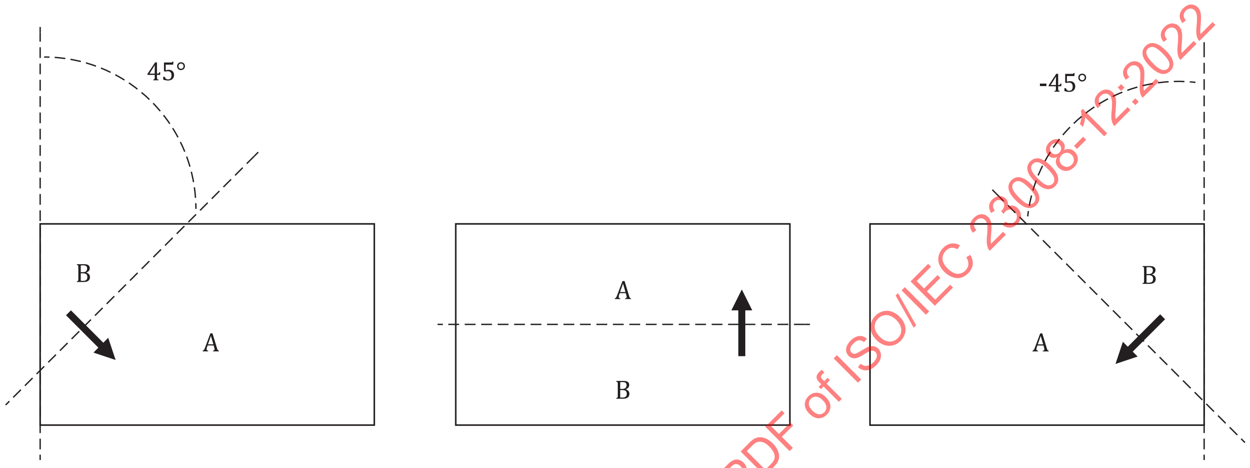
A wipe transition effect consists in an image that gradually replaces another image from one side to another.

NOTE The precise rendering process is deliberately left to the player implementation.

This item property shall be associated with the first of the two consecutive items involved in the transition.

When present for an image item, this item property shall be the last signalled item property for the image item.

Figure 1 illustrates three wipe transition effects from an image A to an image B:



a) starting from the left top corner and following a 45° moving boundary (dash line) toward the opposite side

b) from bottom following a horizontal moving boundary toward the top

c) starting from the right top corner and following a -45° moving boundary toward the opposite side

Figure 1 — Examples of wipe transition effects

6.5.30.2 Syntax

```
aligned(8) class WipeTransitionEffectProperty
extends ItemFullProperty('wipe', version=0, flags=0) {
   unsigned int(8) transition_direction;
}
```

6.5.30.3 Semantics

`transition_direction` identifies the transition direction to apply. It takes one of the following values:

0: from-left; following a vertical moving boundary.

1: from-right; following a vertical moving boundary.

2: from-top; following a horizontal moving boundary.

3: from-bottom; following a horizontal moving boundary.

4: from-left-top; following a 45° moving boundary.

5: from-right-top; following a -45° moving boundary.

6: from-left-bottom; following a -45° moving boundary.

7: from-right-bottom; following a 45° moving boundary.

Other values are reserved.

6.5.31 Zoom transition effect

6.5.31.1 Definition

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

The `ZoomTransitionEffectProperty` documents a suggested zoom transition effect to apply between the display of two consecutive items of a slideshow entity group.

A zoom transition effect consists in a new image that replaces another by zoom-in or zoom-out from the middle of the preceding image with a special shape `transition_shape`.

NOTE The precise rendering process is deliberately left to the player implementation.

This item property shall be associated with the first of the two consecutive items involved in the transition.

When present for an image item, this item property shall be the last signalled item property for the image item.

6.5.31.2 Syntax

```
aligned(8) class ZoomTransitionEffectProperty
extends ItemFullProperty('zoom', version=0, flags=0) {
   unsigned int(1) transition_direction;
   unsigned int(7) transition_shape;
}
```

6.5.31.3 Semantics

`transition_direction` identifies the transition direction to apply. It takes one of the following values:

0: in; (zoom-in using the shape defined by `transition_shape`)

1: out; (zoom-out using the shape defined by `transition_shape`)

`transition_shape` identifies the transition shape to apply. It takes one of the following values

0: rectangular;

1: circle;

2: star;

Other values are reserved.

NOTE Details on transition shape are deliberately left to the player implementation.

6.5.32 Fade transition effect

6.5.32.1 Definition

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

The FadeTransitionEffectProperty documents a suggested fade transition effect to apply between the display of two consecutive items of a slideshow entity group.

A fade transition effect consists in an image that replaces another by either first gradually transitioning to a white or black image and then gradually transitioning from this white or black image to the new image, or by gradually fading in while the other image is gradually fading out.

NOTE The precise rendering process is deliberately left to the player implementation.

This item property shall be associated with the first of the two consecutive items involved in the transition.

When present for an image item, this item property shall be the last signalled item property for the image item.

6.5.32.2 Syntax

```
aligned(8) class FadeTransitionEffectProperty
extends ItemFullProperty('fade', version=0, flags=0) {
   unsigned int(8) transition_direction;
}
```

6.5.32.3 Semantics

transition_direction identifies the type of transition. It takes one of the following values:

0: through-white; the transition is through a white image.

1: through-black; the transition is through a black image.

2: dissolve; the transition is a mutual fading in and out.

Other values are reserved.

6.5.33 Split transition effect

6.5.33.1 Definition

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

The `SplitTransitionEffectProperty` documents a suggested split transition effect to apply between the display of two consecutive items of a slideshow entity group.

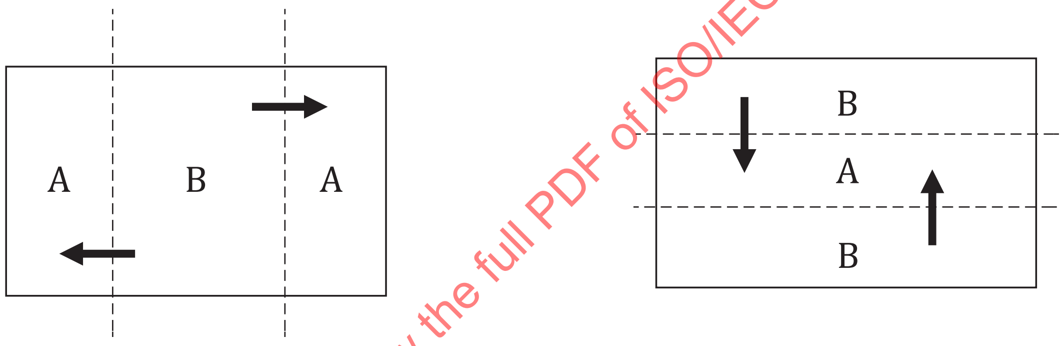
A split transition effect consists in splitting one of the two consecutive images vertically or horizontally in two equal parts and moving both parts horizontally or vertically, respectively, depending on `transition_direction`. In a vertical-in or horizontal-in transition, the second image of the two consecutive images is split and each part moves over the first image from outside toward the inside, each part starting from an opposite edge of the first image. Conversely, in a vertical-out or horizontal-out, the first image of the two consecutive images is split and each part moves in opposite directions from inside toward the outside revealing the second image.

NOTE The precise rendering process is deliberately left to the player implementation.

This item property shall be associated with the first of the two consecutive items involved in the transition.

When present for an image item, this item property shall be the last signalled item property for the image item.

Figure 2 illustrates two split transition effects from an image A to an image B:



a) following a vertical-out, where image A is split vertically and moved out

b) following a horizontal-in, where image B is split horizontally and moved in

Figure 2 — Examples of split transition effects

6.5.33.2 Syntax

```
aligned(8) class SplitTransitionEffectProperty
extends ItemFullProperty('splt', version=0, flags=0) {
   unsigned int(8) transition_direction;
}
```

6.5.33.3 Semantics

`transition_direction` identifies the transitioning direction to apply. It takes one of the following values:

0: vertical-in;

1: vertical-out;

2: horizontal-in;

3: horizontal-out;

Other values are reserved.

6.5.34 Suggested transition period

6.5.34.1 Definition

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

The `SuggestedTransitionPeriodProperty` documents the suggested transition effect duration (from the start to the end of the transition) to apply between the display of two consecutive items of a slideshow entity group.

This item property shall be associated with the first of the two consecutive items involved in the transition.

6.5.34.2 Syntax

```
aligned(8) class SuggestedTransitionPeriodProperty
extends ItemFullProperty('stpe', version=0, flags=0) {
   unsigned int(8) transition_period;
}
```

6.5.34.3 Semantics

`transition_period` indicates the recommended transition period, in units of 1/16 seconds, i.e., the time period from the start to the end of the transition.

6.5.35 Suggested time display duration

6.5.35.1 Definition

Box type:	`'ssld'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per associated item_ID):	No
Quantity (per associated item_ID):	At most one

The `SuggestedTimeDisplayDurationProperty` documents the suggested time display duration to apply to an image item of a slideshow entity group.

The `SuggestedTimeDisplayDurationProperty` may be associated with either a slideshow entity group or an image item of a slideshow entity group. When this property is associated with a slideshow entity group, it documents the default suggested time display duration to apply to image items belonging to the slideshow entity group. When this property is associated with an image item, it documents the suggested time display duration to apply to the image item and supersedes the default value if defined.

When a transition effect applies at the beginning or at the end of an image item display, `duration` represents the time period between the end of the transition from the previous image item and the start of the transition to the next image item. When no transition effect applies, `duration` represents the time period between the start and the end of display of the image item.

6.5.35.2 Syntax

```
aligned(8) class SuggestedTimeDisplayDurationProperty
extends ItemFullProperty('ssld', version=0, flags=0) {
   unsigned int(16) duration;
}
```

6.5.35.3 Semantics

`duration` indicates the recommended image item display duration, in units of 1/16 seconds. The value 0 is reserved.

6.5.36 Ambient viewing environment

6.5.36.1 Definition

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

This property provides information about the characteristics of the nominal ambient viewing environment for the display of the associated content.

6.5.36.2 Syntax

This property has the same syntax as the `AmbientViewingEnvironmentBox` as defined in ISO/IEC 14496-12.

6.5.36.3 Semantics

The semantics of the syntax elements within this property are the same as those specified for the syntax elements of `AmbientViewingEnvironmentBox` as defined in ISO/IEC 14496-12.

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 clauses specify the item_type and the syntax of the item data for some derived image items.

NOTE This document does not require the colour space of the input images for grid or overlay derived image items to be identical. Derived specifications could constrain the colour space to avoid the need for colour space conversions in the derivation process.

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 When removing an item that is marked as an input image of an image overlay item, 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;
   }
   unsigned int FieldLength = ((flags & 1) + 1) * 16; // this is a temporary, non-parsable
variable
   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 and scaling in 6.5.13.

NOTE 2 When removing an item that is marked as an input image of an image grid item, the content of the image grid item might need to be rewritten.

6.6.2.3.2 Syntax

```
aligned(8) class ImageGrid {
   unsigned int(8) version = 0;
   unsigned int(8) flags;
   unsigned int FieldLength = ((flags & 1) + 1) * 16; // this is a temporary, non-parsable
variable
   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 Entity and sample groups

6.8.1 Relating an untimed item to a timed sequence

6.8.1.1 'eqiv' 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 'eqiv' (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 'eqiv' entity group are that all the items included in an 'eqiv' entity group are ‘equivalent’ and that the tracks in the same 'eqiv' 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 ISO/IEC 14496-12:2022, 8.3.4.3 should be used to associate each visual track with its associated audio track.

6.8.1.2 'eqiv' sample group

6.8.1.2.1 Definition

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

All the samples marked by an 'eqiv' 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 'eqiv' 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 'eqiv' 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,

O: time_offset

M: timescale_multiplier.

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

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

6.8.1.2.2 Syntax

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

6.8.1.2.3 Semantics

time_offset specifies the difference of the time related to the image item(s) in the associated 'eqiv' 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 'eqiv' 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.

6.8.2 Burst images

6.8.2.1 Overview

Burst images are a series of rapid succession images. Burst images can contain any number of images and may benefit fully from the image sensor resolution or image processing pipeline, hence providing access to high-resolution original versions of individual images of a burst capture.

An image burst may be stored as an image sequence track or a set of image items. The latter enables flexible and advanced burst image storage and retrieval use cases such as storage of individual burst images as derived image items.

6.8.2.2 'brst' entity group

The burst image entity group ('brst') indicates a set of images that form a temporal burst image set. The entity_id values of the image items in a 'brst' entity group shall be listed in a temporally increasing order.

When an image sequence track is included in a 'brst' entity group, there shall be only one entity_id present in the 'brst' entity group. An image sequence track may be included in a 'brst' entity group to indicate that it contains a burst-captured image sequence as opposed to other types of multi-image captures, such as focal or exposure stacks.

There may be multiple 'brst' entity groupings in the same file with different group_id values.

6.8.3 'tsyn' entity group

A time-synchronized capture entity group ('tsyn') contains entities that were synchronously captured. A single 'tsyn' entity group shall include entity_id values that either resolve to image items or to image sequence tracks, but not a mixture of both. A 'tsyn' entity group including image items indicates that the image items were simultaneously captured spanning the same time. A 'tsyn' entity group including image sequence tracks indicates that all tracks in the group, if played using the timing in the file, are in sync.

Tracks included in the same 'tsyn' entity group shall have the same duration.

There may be multiple 'tsyn' entity groupings in the same file with different group_id values.

6.8.4 'iaug' entity group

An audio-to-image entity group ('iaug') associates an audio track with an image item. When unique IDs are used, an audio-to-image entity group may associate an entity group with an audio track. When displaying an image item mapped to an audio-to-image entity group or an image item from an entity group mapped to an audio-to-image entity group, a reader should also play the related audio track provided that audio playback is enabled (e.g. based on user preferences).

When the audio track duration is longer than the viewing duration, it may be truncated. When the audio track duration is shorter than the viewing duration, then if the audio track has an edit list with the repeat flag set, the audio is repeated; otherwise if the least significant bit of the flags field of the audio-to-image entity group is set, the audio is repeated for the whole viewing duration; otherwise the audio stops after its playback is complete. If the audio track associated with an entity group is already playing when a new image item from the same entity group is displayed, the audio track should continue to play without restarting from the beginning.

The number of entities in an audio-to-image entity group shall be exactly 2, one of the entity_id values shall indicate an image item or an entity group, and the other entity_id value shall indicate an audio track. The number of audio-to-image entity groups including a particular item_ID value or a particular group_ID value shall not be greater than 1.

If the file contains alternative audio tracks to be played with an image item or with image items from an entity group, the audio tracks should have the same alternate_group value in their TrackHeaderBox and any one of these audio tracks should be included in audio-to-image entity group. A reader processing an audio-to-image entity group should check the availability of alternative audio tracks in the same file based on the alternate_group syntax element and select the audio track among the alternatives.

When the FileTypeBox includes a brand specified in this document, and either the file does not include video or image sequence tracks or when video or image sequence tracks are not played, the playback of an audio track should only be started when it is present in an audio-to-image entity group and the image item in the same audio-to-image entity group or an image item of the entity group in the same audio-to-image entity group is displayed.

When the displayed image item is a derived image item, only the audio track that is associated with the final derived image item or with an entity group containing the final derived image item should be played back, and audio tracks that are associated with image items or with entity groups containing image items utilized during the derivation of the derived image item should not be played back. An entity group which is referenced by the audio-to-image entity group shall not contain `entity_id` values which refer to other entity groups also referenced by an audio-to-image entity group.

When an image item is displayed and several audio tracks are associated with an image item both directly via an audio-to-image entity group referring to this image item and indirectly via an audio-to-image entity group referring to an entity group containing this image item, the reader should select the audio track that best suits its needs and display context.

When an audio track is not meant to be played back without the image item or without an image item from the entity group associated with the audio track using the audio-to-image entity group, `track_in_movie` should be equal to 0 in the `TrackHeaderBox` of the audio track.

6.8.5 `'ster'` entity grouping

`'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.

NOTE When a `'ster'` entity group indicates that the primary item contains one view of a stereo pair, the primary item is intended to be displayed in monoscopic viewing of that stereo pair.

6.8.6 Bracketed sets/logically group of images at capture-time

6.8.6.1 Overview

It is useful in some situations to keep and make explicit the relationship between the images for carrying the initial intent of the photographer during all the photo processing workflow. Image items and samples from sequence of images (`'pict'` track) may be logically grouped during capture-time for many different purposes, for instance, any kind of bracketing such as exposure, white-balance, flash exposure, depth-of-field, focus.

For this purpose, image items are grouped using an `EntityToGroupBox` with a specific `grouping_type` value that characterizes the purpose of grouping. Samples from images sequences are grouped using sample grouping with a dedicated `grouping_type` value and group description that characterize the purpose of grouping. `grouping_type` values for bracketed sets defined in this document are listed in the following Table 1.

In addition, for each `EntityToGroupBox` with a given `grouping_type` value, an item property may be associated with each item to provide item-specific parameters for this item within the entity group (e.g. exposure value of an item in an auto exposure bracketing set). The same FourCC code is used for the `grouping_type` of a given `EntityToGroupBox` and for the box type of the item property that provides the parameters specific to an item within a corresponding entity group.

NOTE This allows one HEIF reader to easily retrieve the item property associated with a given group of items among all item properties. (e.g. the parameters of items pertaining to an `EntityToGroupBox('aebr')` are provided by an `ItemFullProperty('aebr')` where `'aebr'` is the FourCC for auto exposure bracketing set).

Table 1 — Bracketed set types

`grouping_type` semantics	FourCC codes
Auto Exposure bracketing	`aebr`
White balance bracketing	`wbbr`
Focus bracketing	`fobr`

Table 1 *(continued)*

`grouping_type` semantics	FourCC codes
Flash Exposure bracketing	`afbr`
Depth of field bracketing	`dobr`

The following subclauses provide more details on each grouping of images.

6.8.6.2 Auto exposure bracketing

6.8.6.2.1 `'aebr'` entity group

The auto exposure bracketing entity group (`'aebr'`) indicates a set of image items that were captured with varying exposure settings. The relative exposure setting for each image item in the entity group should be defined using an auto exposure item property `'aebr'` as specified in 6.5.22.

6.8.6.2.2 `'aebr'` sample group

6.8.6.2.2.1 Definition

The auto exposure bracketing sample group (`'aebr'`) indicates a set of samples that were captured with varying exposure settings. There may be several sample groups of type `'aebr'`, each with a different value of grouping_type_parameter. `AutoExposureBracketingEntry` provides the relative exposure setting for samples associated with this entry in the sample group.

6.8.6.2.2.2 Syntax

```
aligned(8) class AutoExposureBracketingEntry
extends VisualSampleGroupEntry('aebr') {
   int(8) exposure_step;
   int(8) exposure_numerator;
}
```

6.8.6.2.2.3 Semantics

`exposure_step` is an integer value that specifies the increment steps used during the exposure bracketing. When equals to 1, a full stop increment is used, when equals to 2, a half stop increment is used, when equals to 3, a third stop increment is used, and when equals to 4, a quarter stop increment is used. Other values are reserved.

`exposure_numerator` is an integer value specifying the exposure numerator used to compute the exposure value stop of the item.

The exposure value variation of the image item compared to the Automatic Exposure camera settings is expressed as a number of stops that is computed as `exposure_numerator/exposure_step`.

6.8.6.3 White balance bracketing

6.8.6.3.1 `'wbbr'` entity group

The white balance bracketing entity group (`'wbbr'`) indicates a set of image items that were captured with varying white balance settings. The relative white balance setting for each image item in the entity group should be defined using a white balance item property `'wbbr'` as specified in 6.5.23.

6.8.6.3.2 `'wbbr'` sample group

6.8.6.3.2.1 Definition

The white balance bracketing sample group (`'wbbr'`) indicates a set of samples that were captured with varying white balance settings. There may be several sample groups of type `'wbbr'`, each with a different value of `grouping_type_parameter`. `WhiteBalanceBracketingEntry` provides the white balance compensation on Blue/Amber bias and/or Magenta/Green bias for samples associated with this entry in the sample group.

6.8.6.3.2.2 Syntax

```
aligned(8) class WhiteBalanceBracketingEntry
extends VisualSampleGroupEntry('wbbr') {
   unsigned int(16) blue_amber;
   int(8) green_magenta;
}
```

6.8.6.3.2.3 Semantics

`blue_amber` is an unsigned integer indicating the color temperature component of the white balance in Kelvin.

`green_magenta` is a signed integer indicating the color deviation component of white balance in unit of 1/100 Duv (distance to the blackbody locus).

NOTE A Duv of 0 indicates a light source that is neutral. A negative Duv indicates a magenta color shift, while a positive Duv indicates a green color shift.

6.8.6.4 Focus bracketing

6.8.6.4.1 `'fobr'` entity group

The focus bracketing exposure bracketing entity group (`'fobr'`) indicates a set of image items that were captured with varying focus settings. The relative focus setting for each image item in the entity group should be defined using a focus item property `'fobr'` as specified in 6.5.24.

6.8.6.4.2 `'fobr'` sample group

6.8.6.4.2.1 Definition

The focus bracketing exposure bracketing sample group (`'fobr'`) indicates a set of samples that were captured with varying focus settings. There may be several sample groups of type `'fobr'`, each with a different value of `grouping_type_parameter`. `FocusBracketingEntry` provides the relative focus setting for samples associated with this entry in the sample group.

6.8.6.4.2.2 Syntax

```
aligned(8) class FocusBracketingEntry
extends VisualSampleGroupEntry('fobr') {
   unsigned int(16) focus_distance_numerator;
   unsigned int(16) focus_distance_denominator;
}
```

6.8.6.4.2.3 Semantics

The focus distance is expressed in meter as the ratio of `focus_distance_numerator` and `focus_distance_denominator`. Focus at infinity is expressed as division by zero, i.e. `focus_distance_denominator` is equal to 0 and `focus_distance_numerator` should be equal to 0.

6.8.6.5 Flash exposure bracketing

6.8.6.5.1 'afbr' entity group

The flash exposure bracketing entity group ('afbr') indicates a set of image items that were captured with varying flash exposure settings. The relative flash exposure setting for each image item in the entity group should be defined using a flash exposure item property 'afbr' as specified in 6.5.25.

6.8.6.5.2 'afbr' sample group

6.8.6.5.2.1 Definition

The flash exposure bracketing sample group ('afbr') indicates a set of samples that were captured with varying flash exposure settings. There may be several sample groups of type 'afbr', each with a different value of grouping_type_parameter. FlashExposureBracketingEntry provides the relative flash exposure setting for samples associated with this entry in the sample group.

6.8.6.5.2.2 Syntax

```
aligned(8) class FlashExposureBracketingEntry
extends VisualSampleGroupEntry('afbr') {
   int(8) flash_exposure_numerator;
   int(8) flash_exposure_denominator;
}
```

6.8.6.5.2.3 Semantics

flash_exposure_numerator and flash_exposure_denominator are integers representing the flash exposure value of the sample expressed in a number of f-stops as the ratio of flash_exposure_numerator and flash_exposure_denominator.

6.8.6.6 Depth of field bracketing

6.8.6.6.1 'dobr' entity group

The depth of field bracketing entity group ('dobr') indicates a set of image items that were captured with varying depth of field settings. The relative depth of field setting for each image item in the entity group should be defined using a depth of field item property 'dobr' as specified in 6.5.26.

6.8.6.6.2 'dobr' sample group

6.8.6.6.2.1 Definition

The depth of field bracketing sample group ('dobr') indicates a set of samples that were captured with varying depth of field settings. There may be several sample groups of type 'dobr', each with a different value of grouping_type_parameter. DepthOfFieldBracketingEntry provides the relative depth of field setting for samples associated with this entry in the sample group.

6.8.6.6.2.2 Syntax

```
aligned(8) class DepthOfFieldBracketingEntry
extends VisualSampleGroupEntry('dobr') {
   int(8) f_stop_numerator;
   int(8) f_stop_denominator;
}
```

6.8.6.6.2.3 Semantics

The depth of field variation is expressed as an aperture change in a number of stops, and is computed as f_stop_numerator/f_stop_denominator.

6.8.7 User-defined image collections

6.8.7.1 'albc' entity group

The Album Collection entity group ('albc') indicates a set of entities that form an album of images.

Human readable description may be associated with an Album Collection entity group using a user-description item property 'udes'.

NOTE Human-readable description with alternatives languages can be obtained by associating multiple user-description item properties with different lang attributes.

There may be multiple 'albc' entity groupings in the same file with different group_id values, and the same image may belong to multiple album collections.

6.8.7.2 'favc' entity group

The Favorites Collection entity group ('favc') indicates a set of entities that form a collection of favorites images.

Human readable description may be associated with a Favorites Collection entity group using a user-description item property 'udes'.

NOTE Human-readable description with alternatives languages can be obtained by associating multiple user-description item properties with different lang attributes.

There may be multiple 'favc' entity groupings in the same file with different group_id values, and the same image may belong to multiple favorites collections.

6.8.8 Panorama

6.8.8.1 'pano' entity group

The panorama entity group ('pano') indicates a set of images that were captured in order to create a panorama.

The entity_id may refer to an item, to a track, or, when unique IDs are used, to another EntityToGroupBox representing a bracketing set of images.

NOTE The latter is useful to describe enhanced panorama (e.g. an HDR panorama for which each entity composing the panorama is an auto exposure bracketing entity group 'aebr').

Entities in the panorama entity group or samples in an image sequence track referred by a panorama entity group are listed in increasing panorama order. The panorama direction (e.g. left-to-right, right-to-left, etc.) corresponding to the panorama order should be declared using a panorama item property 'pano' associated with the entity group.

When an image sequence track is included in a 'pano' entity group, there shall be only one entity_id present in the 'pano' entity group. An image sequence track may be included in a 'pano' entity group to indicate that it contains captured image sequence destined to form a panorama. An image sequence track present in the 'pano' entity group may contain other bracketing sample grouping (e.g. when the panorama capture is coupled with an auto exposure bracketing to form an HDR panorama).

There may be multiple 'pano' entity groupings in the same file with different group_id values.

6.8.8.2 'pano' sample group

6.8.8.2.1 Definition

The panorama sample group ('pano') indicates a set of samples that were captured in order to create a panorama.

A panorama is composed of frames strictly ordered by increasing frame number. A frame in a panorama may correspond to one sample or to a set of samples (i.e. samples with same frame number). The PanoramaEntry provides the frame number within the panorama of samples associated with this entry in the sample group.

The relationship of samples composing a frame in the panorama may be signalled using a bracketing sample group (e.g. auto exposure bracketing sample group).

NOTE This is useful to describe enhanced panorama (e.g. an HDR panorama for which each frame composing the panorama is an auto exposure bracketing sample group 'aebr'). There can be several sample groups of type 'pano', each with a different value of grouping_type_parameter to indicate multiple panorama within a same track.

6.8.8.2.2 Syntax

```
aligned(8) class PanoramaEntry
extends VisualSampleGroupEntry('pano') {
   unsigned int(16) frame_number;
}
```

6.8.8.2.3 Semantics

frame_number is an unsigned integer representing a frame number in a panorama (in increasing order following the panorama direction).

NOTE The panorama direction is provided by the panorama item property 'pano' associated with the 'pano' entity group referencing the track as described in 6.8.8.1.

6.8.9 Slideshow

6.8.9.1 'slid' entity group

The slideshow entity group ('slid') indicates a set of entities that are intended to form a slideshow. This entity group shall contain entity_id values that point to image items and shall contain no entity_id values that point to tracks.

NOTE 1 An input image item can be a derived image item (e.g. identity, overlay or grid) for advanced slideshows requiring composition of images (possibly onto a canvas).

The entity_id values of the input slideshow images in a slideshow entity group shall be listed in increasing display order.

There may be multiple slideshow entity groups in the same file with different group_id values.

At most one transition effect item property may be associated with an image item of the slideshow entity group to document the transition effect to apply between this image item and the consecutive image item in the list of entities.

NOTE 2 When a same image needs to be included in different slideshows, it is possible to associate it with a different transition effect in a different slideshow either by using a derived image item of type 'iden' in a different slideshow associated with a different transition effect or by having two items that share same data (via ItemLocationBox) but having a different transition effect in different slideshows.

Transition effect item properties should only be marked as essential if they truly are so, as an unrecognized transition property that is marked as essential might impede displaying of a single image.

In most slideshows the transition effects are 'nice to have' but they should not prevent displaying an image if the reader doesn't understand the transition effect.

Audio may be associated with a slideshow entity group by referencing an audio track and the slideshow entity group in an audio-to-image entity group.

A SuggestedTimeDisplayDurationProperty may be associated with the slideshow entity group and/or with the image items belonging to the slideshow entity group to document respectively a default suggested time display duration or the suggested time display duration of image items of the slideshow entity group.

NOTE 3 The SuggestedTimeDisplayDurationProperty can be used by clients that wish to provide an interaction-free slideshow.

When there is associated audio with the slideshow, the audio duration should be congruent with the sum of the suggested time display durations and transition effect durations (possibly including audio repetitions or truncation as defined in audio-to-image entity group)

6.9 Auxiliary image item types and sample formats

6.9.1 CICP-compliant alpha plane

The semantics provided in this subclause are applicable to auxiliary images stored as image items or as part of image sequences or video.

Alpha planes specified in this subclause are identified by “urn:mpeg:mpegB:cicp:systems:auxiliary:alpha” as the aux_type value, for images, or the aux_track_type value, for tracks, of the AuxiliaryTypeProperty (respectively AuxiliaryTypeInfoBox).

The alpha plane format specified in this subclause is preferred (under any brand specified in Clause 10) to the alpha plane format identified by "urn:mpeg:hevc:2015:auxid:1" as specified in Annex B.

Alpha planes should be encoded in monochrome format (i.e., 4:0:0 chroma format) if possible; if they are encoded in colour, they shall be encoded in a colour format with a luma plane and chroma planes (e.g. as 4:2:0 YCbCr) in which case only the luma plane is relevant, and the chroma planes should be ignored in rendering.

The following semantics apply to alpha planes:

— sample value 0 means that the co-located pixel in the master image is transparent (i.e. won't be displayed)

— the maximum sample value (e.g. 255 for 8-bit sample values) means that the co-located pixel in the master image is opaque, i.e. fully covers the background image

— the sample values of the alpha plane divided by the maximum value (e.g. by 255 for 8-bit sample values) provides the multiplier to be used to obtain the intensity for the associated master image

NOTE The term “sample value” used above is to be interpreted as “luma sample value” if encoded with separate luma and chroma planes.

An item reference (respectively track reference) of type 'prem' from the master image item (respectively master image sequence track) to the auxiliary image item (respectively auxiliary image sequence track) signals that the master image(s) is (are) pre-multiplied by the alpha value. If the item or track reference is not present, the master image(s) are not pre-multiplied by the alpha value.

If a master image is rendered onto a visual context by taking an associated alpha plane into account, the following applies:

- When the width or the height of the alpha plane differs from the width or the height of the master image, respectively, the alpha plane is resized to have the same width and height as those of the master image.
- If the master image is not pre-multiplied, the visual context is updated by performing the following operation for each co-located pixel of the master image and the visual context:

  $v_u = m \times \alpha + v_i \times (1 - \alpha)$

- Otherwise (the master image is pre-multiplied), the visual context is updated by performing the following operation for each co-located pixel of the master image and the visual context:

  $v_u = m + v_i \times (1 - \alpha)$

where

v_u is a pixel value in the updated visual context,

m is a pixel value in the master image,

α is an alpha plane value, scaled into the range of 0 (fully transparent) to 1 (fully opaque), inclusive, and

v_i is a pixel value in the visual context given as input to the process.

6.9.2 CICP-compliant depth map

The semantics provided in this subclause are applicable to auxiliary images stored as image items or as part of image sequences or video.

Depth maps specified in this subclause are identified by “urn:mpeg:mpegB:cicp:systems:auxiliary:depth” as the aux_type value, for images, or the aux_track_type value, for tracks, of the AuxiliaryTypeProperty (respectively AuxiliaryTypeInfoBox).

The depth map format specified in this subclause is preferred to the depth map format identified by "urn:mpeg:hevc:2015:auxid:2" as specified in Annex B.

Depth maps should be encoded in monochrome format (i.e., 4:0:0 chroma format) if possible; if they are encoded in colour, they shall be encoded in a colour format with a luma plane and chroma planes (e.g. as 4:2:0 YCbCr) in which case only the luma plane is relevant, and the chroma planes should be ignored in rendering.

6.10 Region items and region annotations

6.10.1 Region item

6.10.1.1 Definition

An item with an item_type value of 'rgan' is a region item that defines one or more regions of an image.

A region item allows associating a same set of item properties or other items or both with each individual region it defines inside an image. Item properties should only be associated with a region item when the property value for the region differs from the matching (explicit or implied) property value for the whole image.

The region item is associated with the image item inside which the regions are defined using an item reference of type 'cdsc' from the region item to the image item.

The geometries of the regions described by the region item are specified in the data of the region item. These geometries define the shape, position and size of the regions inside a reference space that is mapped to the image item with which the region item is associated after any transformative item property is applied to the image item.

The reference space is defined as a 2D coordinate system with the origin (0,0) located at the top-left corner and a maximum size defined by reference_width and reference_height; the x-axis is oriented from left to right and the y-axis from top to bottom. The geometry of a region inside the associated image item is obtained after applying the implicit resampling caused by the difference between the size of the reference space and the size of the associated image item. If the region item has transformative item properties, then the implicit resampling shall be performed on the region item before the first of its transformative item properties is applied.

The geometry of a region described by the region item can be represented either by:

- a point
- a polyline
- a rectangle
- an ellipse
- a polygon
- a mask stored in another image item
- a mask defined in the data of the region item.

Line segments for both polygons and polylines shall not cross (including the implicit closing line for polygons).

The pixels of the associated image item that are part of a region depend on the geometry of the region as follows:

- When the geometry of a region is represented by a point, the pixel located at this point, if it exists, is part of the region.
- When the geometry of a region is represented by a rectangle, an ellipse, or a polygon, the pixels that are inside (including the boundary) of the rectangle, ellipse, or polygon are part of the region.
- When the geometry of a region is represented as a mask defined in the data of the region item (i.e., when geometry_type equals 5), the pixels of the image item corresponding to pixels with the value 1 in the mask image are part of the region.
- When the geometry of a region is represented by a polyline, the pixels that would be coloured if a one-pixel-wide line were drawn along the polyline (even minimally) are part of the region.
- When the geometry of a region is represented by a mask stored in another image item (i.e., when geometry_type equals 4), the image item containing the mask shall be identified by an item reference of type 'mask' from the region item to the image item containing the mask. The image item containing the mask shall be one of the following:
  - A mask item as defined in 6.10.2 or a derived image item of a mask item.
  - An image item that is encoded in monochrome format (i.e. 4:0:0 chroma format).
  - An image item that is encoded in colour. In such a case, it shall be encoded in a colour format with a luma plane and chroma planes (e.g. as 4:2:0 YCbCr). Since only the luma plane is relevant, the chroma planes should be ignored.

A region may be empty if it falls entirely outside the image. An empty region should be ignored.

The mask contained in an image item applies to a region after performing the implicit resampling caused by the difference between the size of the region documented by `width` and `height` and the size of the image item containing the mask.

The following semantics apply to a mask stored as a mask item or as an image item:

- the maximum sample value (e.g. 255 for 8-bit sample values) in the mask image means that the corresponding pixel in the associated image item is part of the region;
- the minimum sample value (i.e. 0) in the mask image means that the corresponding pixel in the associated image item is not part of the region;
- other sample values (e.g. 1 to 254 for 8-bit sample values) is representative of a probability that the corresponding pixel in the associated image item is part of the region. Higher pixel values represent higher probability values.

NOTE 1 The term “sample value” used above is to be interpreted as “luma sample value” if the mask is stored as an image item encoded with separate luma and chroma planes.

NOTE 2 Non-zero mask pixel values can have application-dependent semantics. For example, they can represent confidence scores for each pixel of an AI-generated mask, or saliency score indication per pixel in the defined region.

An alpha plane auxiliary image item may be referenced as a mask by a region item. In such case, the reference space size should be equal to the alpha plane size and the coordinates (x, y) of the region should be (0, 0).

When a region item describes several regions represented by a mask stored in an image item, there shall be one reference for each such region inside the item reference of type `'mask'`. The Nth reference in the item reference of type `'mask'` identifies the mask for the Nth region with `geometry_type` equals to 4, in declaration order, in the region item.

NOTE 3 If the same mask stored as an image item is used for multiple regions declared in a region item, this image item will be referenced several times in the item reference of type `'mask'`.

When the mask is stored as an image item, transformative item properties associated with the image item may apply to the mask before it is used to define the geometry of a region inside the referenced image item.

6.10.1.2 Syntax

```
aligned (8) class RegionItem {
   unsigned int(8) version = 0;
   unsigned int(8) flags;
   unsigned int field_size = ((flags & 1) + 1) * 16; // this is a temporary, non-parsable
variable
   unsigned int(field_size) reference_width;
   unsigned int(field_size) reference_height;
   unsigned int(8) region_count;
   for (r=0; r < region_count; r++) {
      unsigned int(8) geometry_type;
      if (geometry_type == 0) {
         // point
         signed int(field_size) x;
         signed int(field_size) y;
      }
      else if (geometry_type == 1) {
         // rectangle
         signed int(field_size) x;
         signed int(field_size) y;
         unsigned int(field_size) width;
         unsigned int(field_size) height;
      }
```

```
        else if (geometry_type == 2) {
           // ellipse
           signed int(field_size) x;
           signed int(field_size) y;
           unsigned int(field_size) radius_x;
           unsigned int(field_size) radius_y;
        }
        else if (geometry_type == 3 || geometry_type == 6) {
           // polygon or polyline
           unsigned int(field size) point_count;
           for (i=0; i < point_count; i++) {
              signed int(field_size) px;
              signed int(field_size) py;
           }
        }
        else if (geometry_type == 4) {
           // referenced mask
           signed int(field_size) x;
           signed int(field_size) y;
           unsigned int(field_size) width;
           unsigned int(field_size) height;
        }
        else if (geometry_type == 5) {
           // inline mask
           signed int(field_size) x;
           signed int(field_size) y;
           unsigned int(field_size) width;
           unsigned int(field_size) height;
           unsigned int(8) mask_coding_method;
           if (mask_coding_method != 0)
              unsigned int(32) mask_coding_parameters;
           bit(8) data[];
        }
     }
  }
```

6.10.1.3 Semantics

version shall be equal to 0.

(flags & 1) equal to 0 specifies that the length of the fields x, y, width, height, radius_x, radius_y, point_count, px, and py is 16 bits. (flags & 1) equal to 1 specifies that the length of the fields x, y, width, height, radius_x, radius_y, point_count, px, and py is 32 bits. The values of flags greater than 1 are reserved.

reference_width, reference_height specify, in pixel units, the width and height, respectively, of the reference space on which the regions are placed.

geometry_type specifies the type of the geometry of a region. The following values for geometry_type are defined:

0: the region is described as a point.

1: the region is described as a rectangle.

2: the region is described as an ellipse.

3: the region is described as a polygon.

4: the region is described as a mask defined in a referenced image item.

5: the region is described as a mask defined inside the data of the region item.

6: the region is described as a polyline.

Other values are reserved.

x, y specify the coordinates of the point composing the region relatively to the reference space when its geometry is a point. x, y specify the top, left corner of the region relatively to the reference space when its geometry is a rectangle or a mask. x, y specify the centre of the region relatively to the reference space when its geometry is an ellipse. The value (x = 0, y = 0) represents the position of the top-left pixel in the reference space.

NOTE 1 Negative values for the x or y fields enable to specify points, top-left corners, and/or centres that are outside the image. This can be useful for updating region annotations during the edition of an HEIF file.

width, height specify, relatively to the reference space, the width and the height of the region when its geometry is a rectangle or a mask. When geometry_type equals 4, the value 0 indicates that the corresponding width or height value is provided by the ImageSpatialExtentsProperty associated with the item containing the mask. When geometry_type does not equal 4, the value 0 is reserved.

radius_x specifies, relatively to the reference space, the radius on x-axis of the region when its geometry is an ellipse.

radius_y specifies, relatively to the reference space, the radius on y-axis of the region when its geometry is an ellipse.

point_count is the number of points contained in a polygon or a polyline.

NOTE 2 A polygon specifying the geometry of a region is always closed and therefore there is no need to repeat the first point of the polygon as the ending point of the polygon.

px, py specify the coordinates of the points composing the polygon or the polyline relatively to the reference space. The value (px = 0, py = 0) represents the position of the top-left pixel in the reference space.

mask_coding_method indicates the coding method applied on the mask contained in data. The following values are defined:

0: No mask encoding scheme is applied.

1: Mask is compressed with deflate() as defined in IETF RFC 1951.

Other values are reserved.

mask_coding_parameters indicate additional encoding parameters needed for successfully processing the coded mask data. When mask_coding_method is equal to 1, mask_coding_parameters indicate the number of bytes in the coded mask array data. The value of mask_coding_parameters is reserved when the value of mask_coding_method is greater than 1.

data contains the coded or uncompressed representation of a mask that contains the pixels for an inline mask in raster-scan order. Each pixel is represented using a single bit and 8 pixels are packed in one byte. Byte packing shall be in big-endian order. No padding shall be put at the end of each line if the width of the mask is not a multiple of 8 pixels. Only the last data byte shall be padded with bits set to 0.

6.10.2 Mask item

6.10.2.1 Definition

An image item with an item_type value of 'mski' is a mask item that defines a mask. A mask item can be associated with a region item as defined in 6.10.1 to define a mask for a particular region that is defined by the region item.

A mask item has a defined width and height in pixels that shall be signalled using an ImageSpatialExtentsProperty associated with the mask item. Its data represents a lossless-compressed or uncompressed series of bits that corresponds to pixels for a mask in raster-scan order.

If a mask item's data is lossless-compressed, the compression method shall be signalled using the `content_encoding` parameter in the item's `ItemInfoEntry`.

Mask specific configuration information shall be stored in `MaskConfigurationProperty` associated with the mask item.

`bits_per_pixel` and byte packing are defined as follows:

- when `bits_per_pixel` of `MaskConfigurationProperty` equals to 1, 2 or 4; pixels packed per byte are 8, 4 or 2, respectively. Byte packing shall be in big-endian order. No padding shall be put at the end of each line if the mask width is not a multiple of 8 pixels. Only the last data byte shall be padded.
- when `bits_per_pixel` of `MaskConfigurationProperty` equals to 8, 16 or 24; the mask value of a pixel is represented with 1, 2 or 3 bytes, respectively. Bytes of a pixel shall be serialized starting from the most significant byte.

Pixel values shall be interpreted as defined in 6.10.1.

6.10.2.2 Mask configuration item property

6.10.2.2.1 Definition

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

Each image item of type `'mski'` shall have an associated `MaskConfigurationProperty`.

The `MaskConfigurationProperty` provides information required to generate the mask of the associated mask item.

`essential` shall be equal to 1 for a `MaskConfigurationProperty`.

6.10.2.2.2 Syntax

```
aligned(8) class MaskConfigurationProperty
extends ItemFullProperty('mskC', version = 0, flags = 0){
   unsigned int(8) bits_per_pixel;
}
```

6.10.2.2.3 Semantics

`bits_per_pixel` provides the number of bits per pixel. It shall be 1, 2, 4, 8, 16 or 24. Other values are reserved.

6.10.3 Region annotation

A region annotation consists in metadata or image items associated with one or more regions of an image item.

A region annotation may be associated with one or more regions of an image item by:

- describing in a region item the geometry of these one or more regions;
- associating the region item with the image item it describes using the `'cdsc'` (content describes) item reference from the region item to the image item; and,

- associating any or all of the following with the region item:
  - descriptive image properties, using the ItemPropertyAssociationBox;

    NOTE For instance, a region annotation can use a UserDescriptionProperty to associate a description/tags with a region of an image item.
  - metadata items, using an item reference of type 'cdsc' from the metadata item to the region item.
  - image items or an entity group, using an item reference of type 'eroi' from the region item to the image item or entity group.

The region annotation applies to each region described in the region item individually.

The same region annotation may be associated with several image items by associating the same region item with multiple image items.

6.11 Derived region items

6.11.1 General

An item is a derived region item, when it includes a 'drgn' item reference to one or more other region items, which are inputs to the derivation. The reference space and regions defined by a derived region item are obtained by applying the operation of the derived region item to the reference space and regions of the input region items of the derived region item. The exact operation performed to obtain the reference space and the regions is identified by the item_type of the item.

Transformative item properties associated with a derived region item may apply to the reference space and regions defined by the derived region item before they are applied to the image item referenced by the derived region item with an item reference of type 'cdsc'.

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

The following clauses specify the item_type and the syntax of the item data for some derived region items.

6.11.2 Derived region item types

6.11.2.1 Identity derivation

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

NOTE 1 A derived region item of type 'iden' has an empty item body, and the one or more regions of an image defined by this derived region item is the result of applying the transformative item properties associated with this derived region item.

NOTE 2 A derived region item of type 'iden' can be used, for example, when it is desirable to have both the original version of a region item associated with an image item and a cropped version of the same region item (obtained through the 'clap' transformative item property) associated with a cropped version of the same image item.

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. When a brand specified in 10.3 is among the compatible brands of a file, the requirements specified in Clause 7 shall be applied.

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 14496-12:2022, 6.3.2.
- The following constraints on the value of `matrix` shall be applied:
- either `a` or `c` but not both shall be equal to 0x00010000 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 0x00010000 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 degrees.

  - 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}.

(flags & 1) equal to 1 in the EditListBox indicates that media is repeated, as specified in ISO/IEC 14496-12.

When (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. The last repetition of the edit list can be cut to match the track duration.

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

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 FileTypeBox.

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. 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. Not all samples in a master image sequence track need to a 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 'auxl' 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.

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', 0, 0)
{
   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 'irot' transformative item property. Rotation metadata (e.g. according to Annex A) is ignored in the displaying process.

8.2 Metadata for image items

8.2.1 General

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.

8.2.2 Deductive information

8.2.2.1 Definition

Deductive information is a type of metadata that is related to an image and generated by an image analysis process or manually. Image context analysis, object detection or face detection are examples of such algorithms that generate deductive information.

Deductive information can be stored as an opaque data block in an item with an item type of 'uri '.

item_uri_type of ItemInfoEntry box shall contain the absolute URI which uniquely identifies the specification of deductive information.

Items with item type 'uri ' that contain deductive information are linked to the image item via item reference of type 'cdsc'.

An item with item type 'uri ' which contains deductive information may have associated item properties of type 'uuid' which may further provide information about the stored opaque data block.

For example, such a 'uuid' item property may contain information about the parameter and fragment parts of the URL which is used to invoke a visual processing service that generated the deductive information. Other information such as service version, description and schema location information may also be stored in the 'uuid' item property. The exact syntax of the property data should be further defined by the extended_type field of the 'uuid' item property.

8.2.2.2 Cascaded deductive information relationships

Deductive information which is related to an image item may be a result of a cascaded operation of multiple deductive information algorithms (e.g. detect faces and then recognize faces as a separate process and store their related metadata as yet another process).

Item reference type 'dpnd' shall be used to indicate such cascading dependencies. from_item_ID shall refer to the item with item type 'uri 'which contains the deductive information. to_item_ID values shall list the other items with item type 'uri ' which contain deductive information that from_item_ID is dependent on. The order of the list of to_item_ID values shall indicate the application order of the related deductive information.

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.

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` 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

Enhancements to the ISO base media file format are specified in ISO/IEC 14496-12.

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 General requirements on brands

Under any brand, the primary item (or an alternative if alternative support is required) shall be processable by a reader implementing only the required features of that brand. Specifically, given that each brand has a set of properties that a reader is required to support: the item shall not have properties that are marked as essential and are outside this set.

10.2.2 'mif1' structural brand

10.2.2.1 Requirements on files

Files containing the brand 'mif1' in the compatible brands array of the FileTypeBox shall conform to the constraints defined in this subclause.

When the 'mif1' 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 boxes listed in Table 2 are required in a file under the 'mif1' 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.

Table 2 — Required boxes in a file under the 'mif1' brand

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

Note particularly that the brand 'mif1' does not mandate a MovieBox ('moov') and therefore no brand from Annex E of 14496-12 is mandated.

10.2.2.2 Requirements on readers

Support for the boxes listed in Table 3 is required under the 'mif1' brand. The Version column in the following table specifies the versions of the boxes that shall be supported by the readers of the 'mif1' brand.

Table 3 — Boxes to be supported under the 'mif1' 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

Table 3 *(continued)*

Hierarchy of boxes			Ver-sion	Box description
	iinf		0, 1	item information
		infe	2, 3	item information entry
	iref		0, 1	item reference box
	pitm		0, 1	primary item reference
	idat		-	item data
	iprp		-	item properties
NOTE A '-' in the Version column indicates that the box is a container box.				

The boxes from Table 4 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.

Table 4 — Boxes to be supported under the 'mif1' brand when an item is protected

Hierarchy of boxes			Version	Box description
ipro			0	item protection
	sinf		-	protection scheme information box
		frma	-	original format box
		schm	0	scheme type box
		schi	-	scheme information box
NOTE A '-' in the Version column indicates that the box is a container box.				

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 item properties in Table 5 below:

Table 5 — Item properties to be recognized under the 'mif1' brand

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

10.2.3 'mif2' structural brand

10.2.3.1 Requirements on files

Files containing the brand `'mif2'` in the compatible brands array of the `FileTypeBox` shall conform to the constraints defined in this subclause.

The brand `'mif2'` requires support for all file features of the `'mif1'` brand.

When auxiliary images for alpha plane or depth map are present in the file and the `'mif2'` brand is present among the compatible brands array of the `FileTypeBox`, alpha plane or depth map formats as specified respectively in Clauses 6.9.1 and 6.9.2 shall be used instead of alpha plane or depth map formats specified in Annex B.

10.2.3.2 Requirements on readers

Readers shall support all reader features of 'mif1' brand.

Additionally, support for the following is required under this brand:

- within the entity groups, support for `EntityToGroupBox` with `grouping_type` equal to `'altr'`.
- support for `TypeCombinationBox` associated with the `FileTypeBox`.
- support for item reference of type `'prem'`.
- support for alpha plane or depth map formats as specified in Clauses 6.9.1 and 6.9.2 respectively.

Readers shall recognize the item properties in Table 6 below:

Table 6 — Item properties to be recognized under the `'mif2'` brand

Four-character code	Name of the property
`rref`	required reference types
`iscl`	Image scaling

Any reader conforming to the `'mif2'` brand shall support displaying an item that is either the primary item or any item from the alternate group containing 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.

10.2.4 'pred' brand

10.2.4.1 Definition

This brand enables file players to identify and decode HEIF files containing predictively coded image items. When present, this brand shall be among the `compatible_brands` of a `TypeCombinationBox`.

10.2.4.2 Requirements on files

A file having the `'pred'` brand in the compatible_brands of a `TypeCombinationBox` associated with the `FileTypeBox` may contain predictively coded image items and shall conform to following constraints:

- If `'mif1'` brand is among the compatible brands array of the `FileTypeBox` then the primary item shall be independently coded. Additionally, an alternate group containing this primary item and possibly predictively coded image items may exist.

NOTE If the 'pred' brand is in the compatible_brands of a TypeCombinationBox associated with the FileTypeBox, and the 'mif1' brand is not among the compatible brands array of the FileTypeBox then the primary item and possibly all items from the alternate group containing the primary item can be predictively coded image items

— For each predictively coded image item present in the file, the file shall also contain all items that the predictively coded image item depends on (by item references of type 'pred').

10.2.4.3 Requirements on readers

Readers shall support the following:

— support for item reference of type 'pred'.

— within the entity groups, support for EntityToGroupBox with grouping_type equal to 'altr'.

Readers shall recognize the item properties in Table 7 below:

Table 7 — Item properties to be recognized under the 'pred' brand

Four-character code	Name of the property
rref	required reference types

10.2.5 '1pic' brand

10.2.5.1 Definition

This brand enables file players to identify and decode HEIF files containing coded image items that only contain one picture and that picture is intra coded. When present, this brand shall be among the compatible_brands array of the FileTypeBox or of the BrandProperty.

NOTE In case a coded image item is of a multi-layer codec, this brand indicates the presence of only one layer.

10.2.5.2 Requirements on files and items

A file having the '1pic' brand in the compatible_brands array of the FileTypeBox shall contain coded image items that only contain one picture and that picture is intra coded.

An item having a BrandProperty with a '1pic' brand in the compatible_brands array shall be a coded image item that only contain one picture and that picture is intra coded.

10.3 Image sequence brands

10.3.1 'msf1' structural brand

10.3.1.1 Requirements on files

Files containing the brand 'msf1' in the compatible brands array of the FileTypeBox shall conform to the constraints defined in this subclause.

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 B.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 7.3, shall be supported.

Annex A
(normative)

Storage of externally specified metadata

A.1 General

This annex specifies the format to store metadata complying with Exif (JEITA CP-3451E), 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-3451E. 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-3451E 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:

a) for sync samples, the metadata in the `MetaBox` and the metadata that is the metadata sample data;

b) 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' with content_type in ItemInfoEntry set equal to the MIME type specified for MPEG-7 metadata.

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

Clause B.2 specifies requirements for files containing HEVC-coded image items. When a brand specified in Clause B.4.1 is among the compatible brands of a file, the requirements specified in Clause B.2 shall be applied.

The specifications of Clause 6 apply.

B.2.2 Image data

B.2.2.1 Definition

B.2.2.1.1 General

An HEVC image item contains independently coded HEVC image (for the case of external base layer see below). HEVC coded image sequences shall be stored according to Clause B.3.

NOTE 1 When conformance to the 'pred' brand is indicated in addition to any of the HEVC-specific image and image collection brands, an HEVC image item can contain independently coded HEVC image or predictively coded HEVC image which has decoding dependencies to one or more other coded image items. A predictively coded HEVC image item contains an item reference of type 'pred'.

HEVCItemData 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.

Sub-sample information for HEVC image items may be given using an associated sub-sample information item property.

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' can 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 1 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.c

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. The TargetOlsProperty contains the target_ols_idx, which provides the output layer set index to be used as input for the decoding process of L-HEVC coded image item. The target_ols_idx is used as the value of TargetOlsIdx variable, specified in Clause F.8 of HEVC.

NOTE 2 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.

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 BlIrapPicFlag, 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 NAL unit payload but does not include the length field itself.

`NALUnit` contains a single NAL unit. The syntax of a NAL unit shall conform to the syntax specified in ISO/IEC 23008-2 and includes both the two-byte NAL unit header and the variable-length NAL unit 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 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.2.3.3 Operating points information property

B.2.3.3.1 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.2.3.3.2 Syntax

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

B.2.3.3.3 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.2.4 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 23008-2:2020, 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 23008-2:2020, Table F.2.

NOTE ISO/IEC 23008-2:2020, Table F.2 specifies auxiliary picture types for auxiliary pictures within auxiliary picture layers. urn:mpeg:hevc:2015:auxid:xxx can be used for auxiliary pictures stored as items with an item reference of type 'auxl' 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.

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.

nalu_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 NAL unit 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 NAL unit 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 can 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 6.5.28 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 2x2 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 will 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 applied.

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 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.2.3.2, 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 10.2.2.1. 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 the 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 the 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 10.2.2.2 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 the 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 the 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 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 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 [general_profile_compatibility_flags & $2^{(32 - 1)}$] is greater than 0.

Readers for the 'hevx' 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 [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.

Readers shall support all values allowed by 7.2.1 for the matrix syntax element of the TrackHeaderBox and shall conform to the CleanApertureBox of the visual sample entry when displaying an image sequence track with 'hvc1' sample entry.

NOTE This clause and 7.2.1 establish that readers support rotation by 0, 90, 180, and 270 degrees 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 10.2.2.1. 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 for which all of the following conditions are true:

- The item is present in the file.
- One of the following conditions is true:
  - The item is the primary item.
  - The item is any item from the alternate group containing the primary item.
  - The item is included in a 'ster' entity group together with the primary item or any item from the alternate group containing the primary item.
- The item 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 the 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 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 the 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.

NOTE The use of an external base layer is allowed in the 'heim' and 'heis' brands. The coded data for an external base layer picture is not required to be within a HEIF file. The DataReferenceBox allows referring to coded data that is present in an external file. For example, a JPEG file can be used as an external base layer picture for a HEIF file conforming to the 'heim' or 'heis' brand.

B.4.3.3 Requirements on readers

The requirements on readers specified in 10.2.2.2 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 the 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 the 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 conform to the CleanApertureBox of the visual sample entry when displaying an image sequence track with 'hvc1' or 'hvc2' sample entry.

NOTE This clause and 7.2.1 establish that readers support rotation by 0, 90, 180, and 270 degrees 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 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 B.2 (single image and image collection), the MIME type defined in this annex should be used. When such a brand is a compatible brand, this MIME type may also be used.

The registration below is the MIME type registration as recorded at IANA.

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). (A brand in the file header identifies a specific profile of a more general
        format.)

    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.  The MIME subtype name may be 'heif' only if the file conforms to the
    requirements of the 'mif1' brand, and contains that brand as a compatible brand.

Required parameters: none
Optional parameters:
profiles:     Specified by RFC 6381 and its successors.

codecs:       Specified by RFC 6381 and its successors for files conforming to specifications
    derived from ISO/IEC 14496-12.  Note that 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 the type of one or more image items included in the
    file, in any order.  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. The field
    item_type is defined in ISO/IEC 14496-12.
```

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 must 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 must indicate the entire set of item properties that are marked as essential. The item property strings must appear in the order they are associated with the image item in the file.

lhevcptl: For multi-layer image items, specifies the decoding capabilities required for different combinations of layers and temporal sub-layers; the syntax and semantics are specific to HEVC and are formally specified by the lhevcptl optional MIME parameter in ISO/IEC 14496-15 for the L-HEVC sample entry types. Informatively, the syntax is *BLInternal*, *ListItem1*(, *ListItemN*)*, where

BLInternal is 0 or 1;

each Listitem has the structure *OlsIdx.MaxTid.ProfileTierLevel1*(.*ProfileTierLevelN*)*

and

OlsIdx and MaxTid are integers

ProfileTierLevelX is the substring specified for any HEVC sample-entry for the codecs parameter

dependencies:

a list of comma-separated URLs (see RFC 3986) 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. DataReferenceBox and MetaBox are specified in ISO/IEC 14496-12.

Encoding considerations: as for video/mp4

Security considerations: See section 4 of RFC 4337 and section 7 of RFC 6381. This format does not supply integrity or confidentiality protection and so they are applied externally when needed. The security considerations of URLs are discussed in RFC 3986.

Interoperability considerations: Interoperably deployed in reference code available from ISO, Javascript code from Nokia Technologies, in open-source in MP4Box and various other implementations.

Published specification: ISO/IEC 23008-12, available as a Publicly Available Standard at https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Applications: Multimedia, Imaging, Pictures

Fragment identifier considerations: Fragment identifiers are specified in Annex L of ISO/IEC 14496-12, available as a Publicly Available Standard at https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Additional information:

Magic number(s): none

File extension(s): heif (for subtype heif), heic (for subtype heic), hif (for subtypes heif and heic)

```
    Macintosh File Type Code(s): None

Intended usage: Common
```

C.3 Examples

`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 B.3 (image sequences), the MIME type defined in this annex should be used. When such a brand is a compatible brand, this MIME type may also be used.

The registration below is the MIME type registration as recorded at IANA.

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', 'hevx', 'hevm', or 'hevs' brand, and contains that brand as a compatible brand. The MIME subtype name may be 'heif-sequence' only if the file conforms to the requirements of the 'msf1' brand, and contains that brand as a compatible brand. (A brand in the file header identifies a specific profile of a more general format.)

Required parameters: none
Optional parameters:

profiles: Specified by RFC 6381 and its successors.

codecs: Specified by RFC 6381 and its successors for files conforming to specifications derived from ISO/IEC 14496-12. Note that 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.

lhevcptl: For multi-layer image items, specifies the decoding capabilities required for different combinations of layers and temporal sub-layers; the syntax and semantics are specific to HEVC and are formally specified by the lhevcptl optional MIME parameter in ISO/IEC 14496-15 for the L-HEVC sample entry types. Informatively, the syntax is *BLInternal*, *ListItem1*(, *ListItemN*)*, where

BLInternal is 0 or 1;

each Listitem has the structure *OlsIdx.MaxTid.ProfileTierLevel1*(*.ProfileTierLevelN*)*

and

OlsIdx and MaxTid are integers

```
    ProfileTierLevelX is the substring specified for any HEVC sample-entry for the codecs
    parameter

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 4 of RFC 4337 and section 7 of RFC 6381. This
    format does not supply integrity or confidentiality protection and so they are applied
    externally when needed. The security considerations of URLs are discussed in RFC 3986.

Interoperability considerations: Interoperably deployed in reference code available from
    ISO, Javascript code from Nokia Technologies, in open-source in MP4Box and various
    other implementations.

Published specification: ISO/IEC 23008-12, available as a Publicly Available Standard at
    https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Applications: Multimedia, Imaging, Pictures

Fragment identifier considerations: Fragment identifiers are specified in Annex L of
    ISO/IEC 14496-12, available as a Publicly Available Standard at https://standards.iso.
    org/ittf/PubliclyAvailableStandards/index.html

Additional information:

    Magic number(s): none

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

    Macintosh File Type Code(s): None

Intended usage: Common
```

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

Clause E.3 specifies requirements for files containing AVC-coded image items. When a brand specified in subclause E.4.1 is among the compatible brands of a file, the requirements specified in Clause E.3 and its subclauses shall be applied.

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 containing independently coded pictures are normally IDR pictures as defined ISO/IEC 14496-10.

An AVC image item contains independently coded AVC image. AVC coded image sequences shall be stored according to Clause E.3.

NOTE 2 When conformance to the `'pred'` brand is indicated in addition to the AVC-specific image and image collection brand, an AVC image item can contain independently coded AVC image or predictively coded AVC image which has decoding dependencies to one or more other coded image items. A predictively coded AVC image item contains an item reference of type `'pred'`.

`AVCItemData` shall not contain any extractors or aggregators defined in ISO/IEC 14496-15.

NOTE 3 Functionality similar to sharing NAL units through extractors between samples of different tracks can be achieved in image items through the use of extents.

Sub-sample information for AVC image items may be given using an associated sub-sample information item property.

E.2.2.2 Syntax

```
aligned(8) class AVCItemData
{
   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 NAL unit header and the NAL unit payload but does not include the length field itself.

NALUnit contains a single NAL unit. The syntax of a NAL unit shall conform to the syntax specified in ISO/IEC 14496-10 and includes both the NAL unit header and the variable-length NAL unit payload.

E.2.3 AVC configuration item property

Box type:	'avcC'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
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 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'.

There is currently no defined value for AVC auxiliary image for the aux_subtype byte array in AuxiliaryTypeProperty.

E.3 AVC image sequences

E.3.1 General

Clause E.3 specifies requirements for all files containing one or more AVC-coded image sequence tracks. When a brand specified in subclause E.4.2 is among the compatible brands of a file, the requirements specified in Clause E.3 and its subclauses shall be applied.

The specifications of Clause 7 and its subclauses apply.

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

The sample entry of type 'avc1' shall be used for an image sequence track coded with AVC.

The AVCSampleEntry shall be used as specified in ISO/IEC 14496-15.

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

For a track containing an AVC 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.

E.3.3 Auxiliary AVC image sequence tracks

The SEI messages for the auxiliary channel follow the same principle as any other SEI message for an 'avc1' sample entry; i.e. they may be included in the decoder configuration record of the 'avc1' sample entry.

E.4 AVC-specific brands

E.4.1 AVC image and image collection brands

E.4.1.1 General

The brand 'avci' is specified in the following subclauses.

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

— The item has type 'avc1' and conforms to the specifications in E.2.

— The item is not associated with any other types of essential item properties than 'avcC', 'irot', 'clap', and 'imir'.

— The content of the item conforms to the Constrained High Profile of AVC.

E.4.1.2 Requirements on files

Files shall include 'mif1' among the compatible brands and hence conform to the specifications in 10.2.2.1. Additionally, files shall comply with the specifications in E.2.

The files conforming to the 'avci' brand shall additionally be constrained as follows:

Files including 'avci' 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 'avci' brand as specified in E.4.1.1.

— The item is a crop-rotate-mirror derived image item, and the source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'avci' brand as specified in E.4.1.1.

E.4.1.3 Requirements on readers

The requirements on readers specified in 10.2.2.2 shall be supported.

Readers conforming to the `'avci'` 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 `'avci'` brand as specified in E.4.1.1.

— The item is a crop-rotate-mirror derived image item, and the source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the `'avci'` brand as specified in E.4.1.1.

Readers conforming to the `'avci'` brand are recommended but not required to decode all levels as specified by ISO/IEC 14496-10.

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.

E.4.2 AVC image sequence brands

E.4.2.1 General

The brand `'avcs'` is specified in the following subclauses.

E.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 E.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 E.3.

When the `'avcs'` brand is among the compatible brands, there shall be an image sequence track with `'avc1'` 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 `AVCProfileIndication` is equal to 100 and `constraint_set4_flag` is equal to 1.

NOTE When the `'avcs'` brand is among the compatible brands, at least one viewable image sequence track is an AVC image sequence track and contains a bitstream conforming to the Progressive High profile of AVC.

E.4.2.3 Requirements on readers

The requirements on readers specified in 10.3.1.2 shall be supported.

Readers for the `'avcs'` brand shall be able to display an image sequence track with `'avc1'` sample entry type, `track_enabled` equal to 1 and `track_in_movie` equal to 1, for which `AVCProfileIndication` is equal to 100 and `constraint_set4_flag` is equal to 1.

Readers shall support all values allowed by 7.2.1 for the `matrix` syntax element of the `TrackHeaderBox` and shall conform to the `CleanApertureBox` of the visual sample entry when displaying an image sequence track with `'avc1'` sample entry.

NOTE This clause and 7.2.1 establish that readers support rotation by 0, 90, 180, and 270 degrees 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 E.3.3, should be supported.

Annex F
(normative)

Advanced coding image MIME type registration

F.1 Overview

The file extension and MIME type of a file in this family usually reflect the major brand in the `FileTypeBox`. When the major brand indicates a brand related to Clause E.4.1 (single image and image collection), the MIME type defined in this annex should be used. When such a brand is a compatible brand, this MIME type may also be used.

The registration below is the MIME type registration as recorded at IANA.

F.2 Registration

MIME media type name: image
MIME subtype name: avci

This MIME type may be used when the requirements of the 'avci' brand specified in section E.4.1 of ISO/IEC 23008-12 apply (notably when the primary image satisfies the requirements of E.4.1.1). (A brand in the file header identifies a specific profile of a more general format.)

Required parameters: none
Optional parameters:

profiles: Specified by RFC 6381 and its successors.

codecs: Specified by RFC 6381 and its successors for files conforming to specifications derived from ISO/IEC 14496-12. Note that for AVC, the format of a list item included in the value of the codecs parameter is specified in ISO/IEC 14496-15.

itemtypes: As for the MIME type image/heic

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 'avc1', the value after the '.' is the 'avcoti' value as specified in ISO/IEC 14496-15.

Encoding considerations: as for video/mp4

Security considerations: See section 4 of RFC 4337 and section 7 of RFC 6381. This format does not supply integrity or confidentiality protection and so they are applied externally when needed.

Interoperability considerations: Interoperably deployed in reference code available from ISO, Javascript code from Nokia Technologies, in open-source in MP4Box and various other implementations.

Published specification: ISO/IEC 23008-12, available as a Publicly Available Standard at https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Applications: Multimedia, Imaging, Pictures

Fragment identifier considerations: Fragment identifiers are specified in Annex L of ISO/IEC 14496-12, available as a Publicly Available Standard at https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Additional information:
Magic number(s): none

```
    File extension(s): avci

    Macintosh File Type Code(s): None

Intended usage: Common
```

Annex G
(normative)

Advanced coding sequence MIME type registration

G.1 Overview

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 E.4.2 (image sequence), the MIME type defined in this annex should be used. When such a brand is a compatible brand, this MIME type may also be used.

The registration below is the MIME type registration as recorded at IANA.

G.2 Registration

```
MIME media type name: image

MIME subtype name: avcs
    This MIME type may be used when the requirements of the 'avcs' brand specified in
    section E.4.2 of ISO/IEC 23008-12 apply. (A brand in the file header identifies a
    specific profile of a more general format.)

Required parameters: none

Optional parameters:
profiles:     Specified by RFC 6381 and its successors.

codecs:       Specified by RFC 6381 and its successors for files conforming to specifications
    derived from ISO/IEC 14496-12 and derived specifications.  Note that for AVC, 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.

Encoding considerations: as for video/mp4

Security considerations: See section 4 of RFC 4337 and section 7 of RFC 6381. This
    format does not supply integrity or confidentiality protection and so they are applied
    externally when needed.

Interoperability considerations: Interoperably deployed in reference code available from
    ISO, Javascript code from Nokia Technologies, in open-source in MP4Box and various
    other implementations.

Published specification: ISO/IEC 23008-12, available as a Publicly Available Standard at
    https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Applications: Multimedia, Imaging, Pictures

Fragment identifier considerations: Fragment identifiers are specified in Annex L of
    ISO/IEC 14496-12, available as a Publicly Available Standard at https://standards.iso.
    org/ittf/PubliclyAvailableStandards/index.html

Additional information:
    Magic number(s): none

    File extension(s): avcs
```

```
   Macintosh File Type Code(s): None
Intended usage: Common
```

Annex H
(normative)

JPEG in the Image File Format

H.1 Overview

This annex derives a format to encapsulate JPEG-coded images, image collections, and image sequences in the Image File Format specified above. JPEG-specific brands for a single image and an image collection are specified in H.4 and for image sequences in H.5.

H.2 JPEG images and image collections

H.2.1 Definition

Each JPEG image is stored:

a) either as an item of type `'jpeg'` containing a single compressed image item conforming to the ISO/IEC 10918-1.

b) or as an item labelled as and conforming to the MIME type `'image/jpeg'`.

Readers that support items of type `'jpeg'` shall also support items of MIME type `'image/jpeg'`.

NOTE 1 The storage as image items of type `'jpeg'` is preferred, as meta-data is separated, and JPEG header information can be shared. If image items of MIME type `'image/jpeg'` are used, it is possible to have the same coded image content in an image item of type `'jpeg'` present in the file, sharing the storage of both image and metadata by referring to the same data using appropriate item constructors in the item location box.

NOTE 2 This specification does not define a JPEG-specific MIME type or file extension for HEIF files in which the primary image item is a JPEG image item, since storing of JPEG images within HEIF files is encouraged only when the Image File Format provides functionality that is not otherwise available.

The concatenation of the contents of the optional JPEG configuration box (the `JPEGprefix` bytes) with the extents of the JPEG image item shall conform to the specification for a JPEG compressed image as defined in ISO/IEC 10918-1, starting with the SOI (start of image) marker and ending with the EOI (end of image) marker.

NOTE 3 The optional `JPEGPrefix` bytes can be used to share quantization and other tables across several images.

The reconstructed image of a JPEG image item is the image that results when the JPEG compressed data that is not enclosed within any APP marker is decoded.

When an APP1 marker is present in a JPEG image item, it is interpreted as specified in JEITA CP-3451E.

A JPEG image may contain a thumbnail image included within the APP1 marker. Such a thumbnail image may itself be a valid JPEG image. When a JPEG image item contains a JPEG-coded thumbnail image included within the APP1 marker, a separate image item should be present in a HEIF file referring only to the byte range of the thumbnail image included within the APP1 marker, and that separate image item should be indicated, as specified in Clause 6.4.4, to be a thumbnail image for the JPEG image item.

NOTE 4 When a JPEG image item contains a thumbnail image included within the APP1 marker and the thumbnail image is not JPEG-coded, the byte range included within the APP1 marker does not qualify for a thumbnail image item that conforms to the `'jpeg'` brand.

H.2.2 JPEG configuration item property

Box type:	'jpgC'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per item):	No
Quantity (per item):	Zero or one for an image item of type 'jpeg'

Each JPEG image item may have an associated configuration property.

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

H.2.3 Syntax

```
class JPEGConfigurationBox extends Box('jpgC') {
   unsigned int(8) JPEGprefix[];
}
```

H.3 JPEG image sequences

H.3.1 General

Clause H.3 and its subclauses specify requirements for all files containing one or more JPEG-coded image sequence tracks. When a brand specified in Clause H.5 is among the compatible brands of a file, the requirements specified in Clause H.3 and its subclauses shall be applied.

The specifications of Clause 7 and its subclauses apply.

H.3.2 Derivation from ISO/IEC 14496-12

The sample entry of type 'mjpg' shall be used for an image sequence track coded with JPEG.

The JPEGSampleEntry may include a JPEGConfigurationBox as defined above.

The concatenation of the contents of the optional JPEGConfigurationBox (the JPEGprefix bytes) with the sample data of any one sample in the track shall conform to the specification for a JPEG compressed image as defined in ISO/IEC 10918-1, starting with the SOI (start of image) marker and ending with the EOI (end of image) marker.

H.4 JPEG-specific still image brand

H.4.1 General

The brand 'jpeg' is specified in the following subclauses.

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

- The item has type 'jpeg' and conforms to the specifications in H.2, or is coded with MIME type 'image/jpeg' and conforms to that MIME type specification.
- The item is not associated with any other types of essential item properties than 'jpgC', 'irot', 'clap', and 'imir'.

H.4.2 Requirements on files

Files shall include 'mif1' among the compatible brands and hence conform to the specifications in 10.2.2.1. Additionally, files shall comply with the specifications in H.2.

The files conforming to the 'jpeg' brand shall additionally be constrained as follows:

Files including 'jpeg' 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 'jpeg' brand as specified in H.4.1, or is coded with MIME type 'image/jpeg' and conforms to that MIME type specification.

— The item is a crop-rotate-mirror derived image item, and the source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'jpeg' brand as specified in H.4.1.

H.4.3 Requirements on readers

The requirements on readers specified in 10.2.2.2 shall be supported.

Readers conforming to the 'jpeg' 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 'jpeg' brand as specified in H.4.1, or is coded with MIME type 'image/jpeg' and conforms to that MIME type specification.

— The item is a crop-rotate-mirror derived image item, and the source image item of the item is either a crop-rotate-mirror derived image item or a coded image item conforming to the 'jpeg' brand as specified in H.4.1.

H.5 JPEG image sequence brands

H.5.1 General

The brand 'jpgs' is specified in the following subclauses.

H.5.2 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 H.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 H.3.

When the 'jpgs' brand is among the compatible brands, there shall be an image sequence track with 'mjpg' 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.

H.5.3 Requirements on readers

The requirements on readers specified in 10.3.1.2 shall be supported.