INTERNATIONAL STANDARD

ISO/IEC 23008-12

First edition
2017-12

AMENDMENT 1
2020-11

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

Part 12: Image File Format

AMENDMENT 1: Support for predictive image coding, bursts, bracketing and other improvements

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

Partie 12: Format de fichier d'image

AMENDEMENT 1: Support pour le codage prédictif des images, les rafales, le bracketing et autres améliorations

Reference number
ISO/IEC 23008-12:2017/Amd.1:2020(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

Foreword

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

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

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

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

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

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

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

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

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

Part 12: Image File Format

AMENDMENT 1: Support for predictive image coding, bursts, bracketing and other improvements

Clause 3

Add the following terms and definitions at the end of 3.1:

3.1.40
predictively coded image item
image item (3.1.17) that has a decoding dependency to one or more other coded image items (3.1.5)

3.1.41
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.

3.1.42
visual context
visual rendering surface such as a screen buffer, which may already contain visual material, and onto which an image (3.1.15) can be rendered

Clause 4, item a)

Replace the text in item a) with the following:

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;

Clause 4, third paragraph:

Replace the third paragraph with the following:

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.

6.4

Add the following new subclause after subclause 6.4.8:

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'`.

6.5.1

Add the following paragraphs at the end of the subclause 6.5.1, after the NOTE:

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

Add the following new subclauses after subclause 6.5.12:

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 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 IETF 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 IETF 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 subclause 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 subclause 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 colour temperature component of the white balance in Kelvin.

`green_magenta` is a signed integer indicating the colour 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 colour shift, while a positive Duv indicates a green colour 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 subclause 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 subclause 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 subclause 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 organized from left-to-right, then the second row is organised from right-to-left, the third row is organized 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.

6.6.2.3

Replace the text in NOTE 1 with the following:

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 subclause 6.5.13.

6.8

Rename the heading of subclause 6.8 as "Entity and sample groups".

Move the current subclause 6.8 "Relating an untimed item to a timed sequence" and all of its subclauses to subclause 6.8.1. Add the following new subclause 6.8.2 at the end of the subclause:

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 'iaug' entity group associates an image item with an audio track. When displaying an image item mapped to an 'iaug' entity group, a reader should also play the related audio track provided that audio playback is enabled e.g. based on user preferences.

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

If the file contains alternative audio tracks to be played with an image item, the audio tracks should have the same alternate_group value in their TrackHeaderBox and any one of these audio tracks should be included in 'iaug' entity group. A reader processing an 'iaug' 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 'iaug' entity group and the image item in the same 'iaug' entity group is displayed. If the image item is a derived image item, only the audio track that is associated with the final derived image item should be played back, and audio tracks that are associated with image items utilized during the derivation of the derived image item should not be played back.

When an audio track is not meant to be played back without the image item associated with the audio track using the 'iaug' 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 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`
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 subclause 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 subclause 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 colour temperature component of the white balance in Kelvin.

green_magenta is a signed integer indicating the colour 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 colour shift, while a positive Duv indicates a green colour 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 subclause 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 subclause 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 subclause 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 favourites collection entity group (`'favc'`) indicates a set of entities that form a collection of favourites images.

Human readable description may be associated with a favourites 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 favourites 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...) 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 subclause 6.8.8.1.

Clause 6

Add the following new subclause after subclause 6.8:

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 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.

10.2.1.1

Replace the last paragraph at the end of the subclause with the following:

Note particularly that the brand `'mif1'` does not mandate a `MovieBox` (`'moov'`) and therefore no brand from 14496-12:2015, Annex E is mandated.

10.2.1

Renumber the existing subclause 10.2.1 as 10.2.2. Add the following new subclause before 10.2.2:

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.

Add the following new subclause after subclause 10.2.2:

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 subclauses 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 subclauses 6.9.1 and 6.9.2 respectively.

Readers shall recognize the following item properties:

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 obey 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 following item properties:

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

Annexes C, D, F, and G

Replace the existing annexes with the following annexes:

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 here should be used. When such a brand is a compatible brand, this MIME type may also be used.

The registration below is recorded at IANA and this text and the IANA registered text should be kept aligned.

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 IETF RFC 6381 and its successors.

     codecs:                Specified by IETF 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 IETF 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 IETF RFC 4337:2006, section 4 and IETF RFC 6381:2011, section 7. 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 IETF 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 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.