ISO/IEC 14496-12:20xx/AMD X:20xx(E)

ISO/IEC 23008-12:20xx/AMD 4:20xx(E)

ISO/IEC JTC1/SC 29

Secretariat: JISC

Information technology — Coding of audio-visual objects — Part 12: ISO base media file format — Amendment X: MetaBox version 1

Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format — Amendment 3: Low-overhead Image Format

Preliminary WD stage

Contents

Foreword

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

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 documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives 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 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 the following URL: 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*.

A list of all parts in the ISO/IEC 14496 and 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.

Note

This document contains changes to both ISO/IEC 14496-12 and ISO/IEC 23008-12 to keep them synchronized. They should be split into two documents at some stage of the specification process.

Parts of this document rely on the changes contained in Amd3 of ISO/IEC 23008-12.

Information technology — Coding of audio-visual objects — Part 12: ISO base media file format — Amendment X: MetaBox version 1

1 Allow for MetaBox with version 1

In subclause 4.2.2, replace the following text:

Derived specifications are not permitted to define a new version of a box defined in this document.

with the following text:

Derived specifications are not permitted to declare a new version of a box defined in this document, but may define the syntax of a version of a box when this document explicitly allows derived specifications to do so.

2 Declare MetaBox with version 1

In subclause 8.11.1.1, replace the following text:

NOTE In previous editions of this document, the presence of the `HandlerBox` was mandatory. The `'isod'` brand and any brands derived thereof allow omission of the `HandlerBox` unless specified otherwise below.

with the following text:

NOTE 1 In previous editions of this document, the presence of the `HandlerBox` was mandatory. The `'isod'` brand and any brands derived thereof allow omission of the `HandlerBox` unless specified otherwise below.

A `MetaBox` with version 1 is similar to a `MetaBox` with version 0 as it contains general untimed metadata and data that is not annotating other data, and is present at file level.

A `MetaBox` with version 1 carries the same information as a `MetaBox` with version 0, but in an alternate form.

Derived specifications may define the exact syntax carried in the `MetaBox` with version 1.

A file reader shall respect the mapping defined by the derived specification to reconstruct a `MetaBox` with version 0 from the `MetaBox` with version 1 and shall process the content of the reconstructed `MetaBox` with version 0 as specified in this document and in the document containing the derived specification.

When the version of the `MetaBox` is equal to 1, the following applies:

- Specifications apply as if the inferred child boxes and syntax elements values were present in the `MetaBox`.
- When a file reader does not support parsing of the `MetaBox` with version 1, the file reader may ignore the entire `MetaBox`.
- Writers shall not generate files with brands in the `FileTypeBox` that have conflicting definitions of the syntax of the `MetaBox` with version 1.

In subclause 8.11.1.1, replace the following text:

NOTE The `MetaBox` is unusual in that it is a container box yet extends `FullBox`, not `Box`.

with the following text:

NOTE 2 The `MetaBox` is unusual in that it is a container box yet extends `FullBox`, not `Box`.

In subclause 8.11.1.1, replace the following text:

When the `MetaBox` contains neither a `PrimaryItemBox` nor a `GroupsListBox`, the `MetaBox` is required to contain a `HandlerBox` indicating the structure or format of the `MetaBox` contents.

with the following text:

When the `MetaBox` with `version` equal to 0 contains neither a `PrimaryItemBox` nor a `GroupsListBox`, the `MetaBox` is required to contain a `HandlerBox` indicating the structure or format of the `MetaBox` contents.

In subclause 8.11.1.1, add the following text at the end:

NOTE 3 Other specifications based on the box structure defined in this document derive `MetaBox` from a `Box` instead of a `FullBox`. Readers supporting several of these specifications need to be careful when parsing this box. It is suggested to write a `MetaBox` with `version` equal to 1 only when the `size` of the `MetaBox` is not 0 and less than 0x01000000 + 8, which enables a reader to conclude that the next 32-bit unsigned integer after `type` equal to `'meta'` is not a valid `size` field for the first child box within a `MetaBox` derived from a `Box` and thus the `MetaBox` is derived from a `FullBox`.

In subclause 8.11.1.2, replace everything with the following text:

```
aligned(8) class MetaBox (version, flags)
   extends FullBox('meta', version, flags)
{
   if (version == 0) {
      HandlerBox            theHandler;        // optional
      PrimaryItemBox        primary_resource;  // optional
      DataInformationBox    file_locations;    // optional
      ItemLocationBox       item_locations;    // optional
      ItemProtectionBox     protections;       // optional
      ItemInfoBox           item_infos;        // optional
      IPMPControlBox        IPMP_control;      // optional
      ItemReferenceBox      item_refs;         // optional
      ItemPropertiesBox     item_properties;   // optional
      ItemDataBox           item_data;         // optional
      GroupsListBox         entity_groups;     // optional
      Box                   other_boxes[];     // optional
   } else if (version == 1) {
      // derived specifications may define the syntax within version 1
      bit(8) data[]; // until the end of the box
   }
}
```

3 Annex changes

In clause E.4, add the following text after NOTE 2:

Only the support for version 0 of the `MetaBox` is required; version 1 support is not required.

Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format — Amendment 3: Low-overhead Image Format

1 Allow for MetaBox with version 1

In clause 6.2, replace the following text:

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`.

with the following text:

A `MetaBox` (`'meta'`), as specified in ISO/IEC 14496-12, is required at file level. That `MetaBox` with version 0 shall contain the boxes specified to be mandatorily present by ISO/IEC 14496-12. This document specifies the transformation to reconstruct the `MetaBox` with version 0 from a `MetaBox` with version 1. Additional requirements for the contents of the file-level `MetaBox` are specified in this document. The 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`.

Add the following new subclause after clause 6.2:

6.2.1 Files with MetaBox version 1

When a `MetaBox` (`'meta'`) with version 1 as specified in ISO/IEC 14496-12 is used, it is considered equivalent to the reconstructed `MetaBox` (`'meta'`) with version 0. File readers shall process the file as defined in Clause 6.10 of this document.

The content of `MetaBox` (`'meta'`) with version 1 is specified in subclause 6.10.1 of this document.

The transformation applied by the file reader to reconstruct a `MetaBox` (`'meta'`) with version 0 from a `MetaBox` (`'meta'`) with version 1 is specified in subclause 6.10.2.

2 Add PixelInformationProperty with version 1

Change the following text in clause 6.5.6.1:

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

To:

> The `PixelInformationProperty` descriptive item property indicates the number and bit depth of colour and alpha/depth components, if present, in the reconstructed image of the associated image item.

Change the syntax in clause 6.5.6.2 to:

```
aligned(8) class PixelInformationProperty
extends ItemFullProperty('pixi', version, px_flags){
   if(version == 0 || version == 1) {
     unsigned int(8) num_channels;
     for (i=0; i<num_channels; i++) {
          unsigned int(8) bits_per_channel;
     }
   }
   if(version == 1) {
     unsigned int has_alpha = (px_flags & 1) != 0;
     unsigned int alpha_premultiplied = (px_flags & 2) != 0;
     unsigned int has_subsampling = (px_flags & 4) != 0;
     for (i=0; i<num_channels; i++) {
          unsigned int(3) channel_idc;
          unsigned int(2) channel_data_type;
          unsigned int(1) channel_label_present;
          unsigned int(2) reserved;
          if(has_subsampling) {
               unsigned int(4) subsampling_type;
               unsigned int(4) subsampling_location;
          }
          if(channel_label_present) {
               utf8string channel_label;
          }
     }
   }
}
```

Add the following text to the end of clause 6.5.6.3:

has_alpha: If true, indicates that the coded image contains alpha. If true, at least one of the channels shall have channel_idc set to 0 and one shall have channel_idc set to 1. Only applicable when version is 1.

alpha_premultiplied: If true, the channels with channel_idc set to 0 have been premultiplied by the channel with channel_idc set to 1. If true, has_alpha shall also be true. Only applicable when version is 1.

has_subsampling: If true, indicates that subsampling information is present. Only applicable when version is 1.

channel_idc: This field indicates the contents of the channel. A value of 0 indicates colour/grayscale. A value of 1 indicates alpha. A value of 2 indicates depth. Values 3-8 are reserved for future use. At most one channel shall have a channel_idc of 1.

channel_data_type: This field indicates the data type of the channel. A value of 0 indicates unsigned integers. A value of 1 indicates signed integers. A value of 2 indicates floating point. Value 3 is reserved for future use.

subsampling_type: This field indicates the subsampling type as specified by **GenericSubsamplingType** in Rec. ITU-T H.273 | ISO/IEC 23091-2.

subsampling_location: This field indicates the subsampling sample location as specified by **GenericSubsamplingSampleLocType** in Rec. ITU-T H.273 | ISO/IEC 23091-2.

EDITORS NOTE:
Anything other than unsigned integers may need clarifications with regards to colr-nclx as noted here: https://git.mpeg.expert/MPEG/Systems/FileFormat/HEIF/-/issues/95#note_81705

3 Define MetaBox with version 1

Add the following new clause after clause 6.9:

6.10 Processing MetaBox with version 1

6.10.1 Content of MetaBox with version 1

6.10.1.1 Definition

Box type:	`'meta'  (version=1)`
Container:	`file`
Mandatory:	No
Quantity:	At most one

The `MetaBox` with version 1 as defined in subclause 6.10.1.2 provides a more compact way to carry image items in a file than a `MetaBox` with version 0. The content of `MetaBox` with version 1 is defined below.

6.10.1.2 Syntax

```
aligned(8) class MetaBoxV1(flags) extends MetaBox(version = 1, flags)
{
   unsigned int has_alpha                =  (flags & 0x000001) != 0;
   unsigned int alpha_is_premultiplied   =  (flags & 0x000002) != 0;
   unsigned int has_hdr                  =  (flags & 0x000004) != 0;
   unsigned int has_explicit_cicp        =  (flags & 0x000008) != 0;
   unsigned int has_icc                  =  (flags & 0x000010) != 0;
   unsigned int has_exif                 =  (flags & 0x000020) != 0;
   unsigned int has_xmp                  =  (flags & 0x000040) != 0;
   unsigned int full_range               =  (flags & 0x000080) != 0;
   unsigned int pixel_format             =  (flags & 0x000F00) >> 8;   // 4 bits
   unsigned int chroma_subsampling       =  (flags & 0x003000) >> 12; // 2 bits
   unsigned int is_horizontally_centered =  (flags & 0x004000) != 0;
   unsigned int is_vertically_centered   =  (flags & 0x008000) != 0;
   unsigned int orientation              =  (flags & 0x070000) >> 16; // 3 bits
   unsigned int has_explicit_codec_types =  (flags & 0x080000) != 0;
   unsigned int dimension_size           = ((flags & 0x100000) != 0) ?  7 : 15;
   unsigned int codec_config_size_size   = ((flags & 0x200000) != 0) ?  3 : 12;
   unsigned int item_data_size_size      = ((flags & 0x400000) != 0) ? 15 : 28;
   unsigned int metadata_size_size       = ((flags & 0x800000) != 0) ? 10 : 20;

   unsigned int(dimension_size) width_minus_one;
   unsigned int(dimension_size) height_minus_one;

   if (has_icc)
      unsigned int(metadata_size_size) icc_data_size_minus_one;

   if (has_explicit_cicp)
      bit(8) colour_primaries;
```

```
else if (has_icc)
   colour_primaries = 2;
else
   colour_primaries = 1;

if (has_explicit_cicp)
   bit(8) transfer_characteristics;
else if (has_icc)
   transfer_characteristics = 2;
else
   transfer_characteristics = 13;

if (chroma_subsampling == 0)
   matrix_coefficients = 2;
else if (has_explicit_cicp)
   bit(8) matrix_coefficients;
else
   matrix_coefficients = 6;

if (has_explicit_codec_types) {
   bit(32) infe_type;
   bit(32) codec_config_type;
}

unsigned int(codec_config_size_size) main_item_codec_config_size;
unsigned int(item_data_size_size) main_item_data_size_minus_one;

if (has_alpha)
   unsigned int(item_data_size_size) alpha_item_data_size;
if (has_alpha && alpha_item_data_size > 0)
   unsigned int(codec_config_size_size) alpha_item_codec_config_size;

if (has_hdr) {
   bit(1) has_gainmap;

   bit(1) has_clli;
   bit(1) has_mdcv;
   bit(1) has_cclv;
   bit(1) has_amve;
   bit(1) has_reve;
   bit(1) has_ndwt;
   if (has_clli)
      ContentLightLevel clli;
   if (has_mdcv)
      MasteringDisplayColourVolume mdcv;
   if (has_cclv)
      ContentColourVolume cclv;
   if (has_amve)
      AmbientViewingEnvironment amve;
   if (has_reve)
      ReferenceViewingEnvironment reve;
   if (has_ndwt)
      NominalDiffuseWhite ndwt;
}
if (has_hdr && has_gainmap) {
   unsigned int(metadata_size_size) gainmap_metadata_size;
   unsigned int(item_data_size_size) gainmap_item_data_size;
   unsigned int(codec_config_size_size) gainmap_item_codec_config_size;
   unsigned int(dimension_size) gainmap_width_minus_one;
   unsigned int(dimension_size) gainmap_height_minus_one;
   bit(8) gainmap_matrix_coefficients;
   bit(1) gainmap_full_range;
   bit(2) gainmap_chroma_subsampling;
   bit(1) gainmap_is_horizontally_centered;
   bit(1) gainmap_is_vertically_centered;
   bit(4) gainmap_pixel_format;

   bit(1) tmap_has_clli;
   bit(1) tmap_has_mdcv;
   bit(1) tmap_has_cclv;
   bit(1) tmap_has_amve;
   bit(1) tmap_has_reve;
   bit(1) tmap_has_ndwt;
   if (tmap_has_clli)
      ContentLightLevel tmap_clli;
   if (tmap_has_mdcv)
      MasteringDisplayColourVolume tmap_mdcv;
   if (tmap_has_cclv)
      ContentColourVolume tmap_cclv;
   if (tmap_has_amve)
      AmbientViewingEnvironment tmap_amve;
   if (tmap_has_reve)
      ReferenceViewingEnvironment tmap_reve;
   if (tmap_has_ndwt)
      NominalDiffuseWhite tmap_ndwt;

   bit(1) tmap_has_icc;
   if (tmap_has_icc)
      unsigned int(metadata_size_size) tmap_icc_data_size_minus_one;
   bit(1) tmap_has_explicit_cicp;
   if (tmap_has_explicit_cicp) {
      bit(8) tmap_colour_primaries;
      bit(8) tmap_transfer_characteristics;
      bit(8) tmap_matrix_coefficients;
      bit(1) tmap_full_range;
   }
   else {
      tmap_colour_primaries = 1;
      tmap_transfer_characteristics = 13;
      tmap_matrix_coefficients = 6;
      tmap_full_range = 1;
   }
}

if (has_exif)
   unsigned int(metadata_size_size) exif_data_size_minus_one;
if (has_xmp)
```

```
      unsigned int(metadata_size_size) xmp_data_size_minus_one;

   trailing_bits(); // bit padding till byte alignment

   if (has_alpha && alpha_item_data_size > 0 && alpha_item_codec_config_size > 0)
      unsigned int(8) alpha_item_codec_config[alpha_item_codec_config_size];
   if (has_hdr && has_gainmap && gainmap_item_codec_config_size > 0)
      unsigned int(8) gainmap_item_codec_config[gainmap_item_codec_config_size];
   if (main_item_codec_config_size > 0)
      unsigned int(8) main_item_codec_config[main_item_codec_config_size];

   if (has_icc)
      unsigned int(8) icc_data[icc_data_size_minus_one + 1];
   if (has_hdr && has_gainmap && tmap_has_icc)
      unsigned int(8) tmap_icc_data[tmap_icc_data_size_minus_one + 1];
   if (has_hdr && has_gainmap && gainmap_metadata_size > 0) {
      unsigned int(8) gainmap_metadata[gainmap_metadata_size];

   if (has_alpha && alpha_item_data_size > 0)
      unsigned int(8) alpha_item_data[alpha_item_data_size];
   if (has_hdr && has_gainmap && gainmap_item_data_size > 0)
      unsigned int(8) gainmap_item_data[gainmap_item_data_size];
   unsigned int(8) main_item_data[main_item_data_size_minus_one + 1];
   if (has_exif)
      unsigned int(8) exif_data[exif_data_size_minus_one + 1];
   if (has_xmp)
      unsigned int(8) xmp_data[xmp_data_size_minus_one + 1];
}
```

EDITORS NOTE:
HDR-related specifications such as ISO 21496-1 were not finalized at the time of writing this document. The changes in this document should be kept as close to the up-to-date HDR-related specifications as possible.

6.10.1.3 Semantics

has_alpha: 0 specifies that the image is opaque. Otherwise the image has an alpha layer, whether the codec has native translucency support or an alpha auxiliary image item is used.

alpha_is_premultiplied: when set to 1 specifies that the color channels are pre-multiplied by the alpha channel, otherwise the color channels are not pre-multiplied. Ignored if has_alpha is 0.

has_hdr: 0 specifies that the image is SDR and has no associated HDR-related signaling. Otherwise the image is either SDR with a SDR-to-HDR gain map, or HDR with an optional HDR-to-SDR gain map.

has_explicit_cicp: 0 specifies sRGB on-screen colors as the values of **ColourPrimaries** and **TransferCharacteristics**, as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2, respectively set to 1 and 13 if has_icc is 0, and to 2 and 2 otherwise. 0 specifies sRGB on-screen colors as the value of **MatrixCoefficients**, as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2, set to 2 if chroma_subsampling is 0, and to 6 otherwise. 1 specifies that these values are signaled explicitly.

has_icc: specifies the presence of an **ICC profile** as defined in ISO 15076-1 or ICC.1[23].

has_exif: specifies the presence of **Exif** metadata.

has_xmp: specifies the presence of **XMP** metadata.

full_range: carries a **VideoFullRangeFlag** as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the main image.

pixel_format: specifies the number of bits per channel and the presence of a decimal point for the pixels of the reconstructed image. Values 0, 1, and 2 correspond to half-precision float (**binary16**), single-precision float (**binary32**), and double-precision float (**binary64**) formats as defined by IEEE 754-2008, respectively. Other values correspond to the bit depth minus one per channel of unsigned integer samples.

chroma_subsampling: 0 specifies that there is exactly one channel of coded color samples (monochrome), otherwise there are exactly three channels of coded color samples (one luma channel and two chroma channels). A value of 1 specifies that these chroma channels are subsampled both horizontally and vertically by a factor 2 (i.e. 4:2:0). A value of 2 specifies that these chroma channels are subsampled by a factor 2 horizontally (i.e. 4:2:2). A value of 3 specifies that there is no subsampling of these chroma channels (i.e. 4:4:4).

is_horizontally_centered: 0 specifies that the chroma samples of the main image are co-located horizontally with the luma samples of the main image, otherwise they are horizontally centered between the luma samples of the main image. Ignored unless chroma_subsampling is 1 or 2.

is_vertically_centered: 0 specifies that the chroma samples of the main image are co-located vertically with the luma samples of the main image, otherwise they are vertically centered between the luma samples of the main image. Ignored unless chroma_subsampling is 1.

orientation: carries the **Exif orientation** value as defined in JEITA CP-3451C section 4.6.4.A "Orientation", minus one.

has_explicit_codec_types: 0 specifies that the minor_version of the FileTypeBox carries a brand defining a single coded image item type and a single codec configuration property box type. Shall be set to 1 otherwise.

dimension_size: 0 specifies that the length of the fields width_minus_one and height_minus_one is 7 bits, otherwise 15 bits.

codec_config_size_size: 0 specifies that the length of the fields main_item_codec_config_size, alpha_item_codec_config_size, and gainmap_item_codec_config_size is 3 bits, otherwise 12 bits.

item_data_size_size: 0 specifies that the length of the fields main_item_data_size_minus_one, alpha_item_data_size, and gainmap_item_data_size is 15 bits, otherwise 28 bits.

metadata_size_size: 0 specifies that the length of the fields icc_data_size_minus_one, exif_data_size_minus_one, and xmp_data_size_minus_one is 10 bits, otherwise 20 bits.

width_minus_one: carries the width minus one of the reconstructed image in pixels.

height_minus_one: carries the height minus one of the reconstructed image in pixels.

icc_data_size_minus_one: carries the size minus one in bytes of the **ICC profile** as defined in ISO 15076-1 or ICC.1[23]. -1 if has_icc is 0.

colour_primaries: carries a **ColourPrimaries** value as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the main image.

transfer_characteristics: carries a **TransferCharacteristics** value as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the main image.

matrix_coefficients: carries a **MatrixCoefficients** value as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the main image.

infe_type: carries the coded image item type. Corresponds to the item_type field of the version 2 of the ItemInfoEntry box. Defined by the brand carried by the minor_version of the FileTypeBox if has_explicit_codec_types is 0.

codec_config_type: carries the codec configuration property box type. Defined by the brand carried by the minor_version of the FileTypeBox if has_explicit_codec_types is 0.

main_item_codec_config_size: carries the size of the codec configuration for the main image item in bytes.

main_item_data_size_minus_one: carries the size minus one of the coded sample data for the main image item in bytes.

alpha_item_data_size: carries the size of the coded sample data for the alpha auxiliary image item in bytes. If has_alpha is set to 1, the value 0 specifies that the codec has native translucency support and that the alpha samples are coded alongside the color samples in the main_item_data chunk. 0 if has_alpha is 0.

alpha_item_codec_config_size: carries the size of the codec configuration for the alpha auxiliary image item in bytes. The value 0 specifies that the codec does not need any configuration data for alpha. 0 if alpha_item_data_size is 0.

has_gainmap: 0 specifies that the file has no tone-mapped image and no associated HDR-related ISO 21496-1 gain map. Otherwise the file contains a tone-mapped image and is associated with a gain map, whether the codec has native gain map support or a separate **gain map image item** is used. 0 if has_hdr is 0.

has_clli: 1 specifies that there is signaling for ContentLightLevel attached to the main image. Otherwise no such signaling is present. 0 if has_hdr is 0.

has_mdcv: 1 specifies that there is signaling for MasteringDisplayColourVolume attached to the main image. Otherwise no such signaling is present. 0 if has_hdr is 0.

has_cclv: 1 specifies that there is signaling for ContentColourVolume attached to the main image. Otherwise no such signaling is present. 0 if has_hdr is 0.

has_amve: 1 specifies that there is signaling for AmbientViewingEnvironment attached to the main image. Otherwise no such signaling is present. 0 if has_hdr is 0.

has_reve: 1 specifies that there is signaling for ReferenceViewingEnvironment attached to the main image. Otherwise no such signaling is present. 0 if has_hdr is 0.

has_ndwt: 1 specifies that there is signaling for NominalDiffuseWhite attached to the main image. Otherwise no such signaling is present. 0 if has_hdr is 0.

clli: The box body of the **ContentLightLevelBox** as defined in ISO/IEC 14496-12 attached to the main image. Only present if has_clli is 1.

mdcv: The box body of the **MasteringDisplayColourVolumeBox** as defined in ISO/IEC 14496-12 attached to the main image. Only present if has_mdcv is 1.

cclv: The box body of the **ContentColourVolumeBox** as defined in ISO/IEC 14496-12 attached to the main image. Only present if has_cclv is 1.

amve: The box body of the **AmbientViewingEnvironmentBox** as defined in ISO/IEC 14496-12 attached to the main image. Only present if has_amve is 1.

reve: The box body of the **ReferenceViewingEnvironmentBox** attached to the main image. Only present if has_reve is 1.

ndwt: The box body of the **NominalDiffuseWhiteBox** attached to the main image. Only present if has_ndwt is 1.

gainmap_metadata_size: carries the size of the gain map metadata. 0 if has_gainmap is 0.

gainmap_item_data_size: carries the size of the coded sample data for the HDR-related gain map image item in bytes. If has_gainmap is set to 1, a size of 0 is reserved for future use. 0 if has_gainmap is 0.

gainmap_item_codec_config_size: carries the size of the codec configuration for the gain map auxiliary image item in bytes. The value 0 specifies that the codec does not need any configuration data for the gain map. 0 if gainmap_item_data_size is 0.

gainmap_width_minus_one: carries the width minus one of the gain map image in pixels.

gainmap_height_minus_one: carries the height minus one of the gain map image in pixels.

gainmap_matrix_coefficients: carries a **MatrixCoefficients** value as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the gain map image.

gainmap_full_range: carries a **VideoFullRangeFlag** as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the gain map image.

gainmap_chroma_subsampling: 0 specifies that there is exactly one channel of coded gain map samples (monochrome), otherwise there are exactly three channels of coded gain map samples (one luma channel and two chroma channels). A value of 1 specifies that these chroma channels are subsampled both horizontally and vertically by a factor 2 (i.e. 4:2:0). A value of 2 specifies that these chroma channels are subsampled by a factor 2 horizontally (i.e. 4:2:2). A value of 3 specifies that there is no subsampling of these chroma channels (i.e. 4:4:4).

gainmap_is_horizontally_centered: 0 specifies that the chroma samples of the gain map image are co-located horizontally with the luma samples of the gain map image, otherwise they are horizontally centered between the luma samples of the gain map image. Ignored unless gainmap_chroma_subsampling is 1 or 2.

gainmap_is_vertically_centered: 0 specifies that the chroma samples of the gain map image are co-located vertically with the luma samples of the gain map image, otherwise they are vertically centered between the luma samples of the gain map image. Ignored unless gainmap_chroma_subsampling is 1.

gainmap_pixel_format: specifies the number of bits per channel and the presence of a decimal point for the pixels of the gain map image. Values 0, 1, and 2 correspond to half-precision float (**binary16**), single-precision float (**binary32**), and double-precision float (**binary64**) formats as defined by IEEE 754-2008, respectively. Other values correspond to the bit depth minus one per channel of unsigned integer samples.

tmap_has_clli: 1 specifies that there is signaling for ContentLightLevel attached to the tone-mapped image. Otherwise no such signaling is present. 0 if has_gainmap is 0.

tmap_has_mdcv: 1 specifies that there is signaling for MasteringDisplayColourVolume attached to the tone-mapped image. Otherwise no such signaling is present. 0 if has_gainmap is 0.

tmap_has_cclv: 1 specifies that there is signaling for ContentColourVolume attached to the tone-mapped image. Otherwise no such signaling is present. 0 if has_gainmap is 0.

tmap_has_amve: 1 specifies that there is signaling for AmbientViewingEnvironment attached to the tone-mapped image. Otherwise no such signaling is present. 0 if has_gainmap is 0.

tmap_has_reve: 1 specifies that there is signaling for ReferenceViewingEnvironment attached to the tone-mapped image. Otherwise no such signaling is present. 0 if has_gainmap is 0.

tmap_has_ndwt: 1 specifies that there is signaling for NominalDiffuseWhite attached to the tone-mapped image. Otherwise no such signaling is present. 0 if tmap_has_gainmap is 0.

tmap_clli: The box body of the **ContentLightLevelBox** as defined in ISO/IEC 14496-12 attached to the tone-mapped image. Only present if tmap_has_clli is 1.

tmap_mdcv: The box body of the **MasteringDisplayColourVolumeBox** as defined in ISO/IEC 14496-12 attached to the tone-mapped image. Only present if tmap_has_mdcv is 1.

tmap_cclv: The box body of the **ContentColourVolumeBox** as defined in ISO/IEC 14496-12 attached to the tone-mapped image. Only present if tmap_has_cclv is 1.

tmap_amve: The box body of the **AmbientViewingEnvironmentBox** as defined in ISO/IEC 14496-12 attached to the tone-mapped image. Only present if tmap_has_amve is 1.

tmap_reve: The box body of the **ReferenceViewingEnvironmentBox** attached to the tone-mapped image. Only present if tmap_has_reve is 1.

tmap_ndwt: The box body of the **NominalDiffuseWhiteBox** attached to the tone-mapped image. Only present if tmap_has_ndwt is 1.

tmap_has_icc: if 1, specifies that the tone-mapped image is associated with an ICC profile. 0 if has_gainmap is 0.

tmap_icc_data_size_minus_one: carries the size minus one in bytes of the **ICC profile**, as defined in ISO 15076-1 or ICC.1[23], associated with the tone-mapped image. Not signaled if tmap_has_icc is 0.

tmap_has_explicit_cicp: 0 specifies sRGB on-screen colors as the values of **ColourPrimaries**, **TransferCharacteristics** and **MatrixCoefficients**, as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2, associated with the tone-mapped image, set to 1, 13 and 6, respectively. Otherwise these values are signaled explicitly.

tmap_colour_primaries: carries a **ColourPrimaries** value as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the tone-mapped image.
tmap_transfer_characteristics: carries a **TransferCharacteristics** value as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the tone-mapped image.
tmap_matrix_coefficients: carries a **MatrixCoefficients** value as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the tone-mapped image.
tmap_full_range: carries a **VideoFullRangeFlag** as defined in Rec. ITU-T H.273 | ISO/IEC 23091-2 for the tone-mapped image. Set to 1 if tmap_has_explicit_cicp is 0.

exif_data_size_minus_one: specifies the size minus one of the **Exif** metadata in bytes. -1 if has_exif is 0.
xmp_data_size_minus_one: specifies the size minus one of the **XMP** metadata in bytes. -1 if has_xmp is 0.
trailing_bits: padding bits to ensure payloads are 8-bit aligned. Shall be 0.

alpha_item_codec_config: carries the optional alpha image codec configuration data. When alpha_item_codec_config_size is 0, alpha_item_codec_config is not present.
gainmap_item_codec_config: carries the HDR-related gain map image item codec configuration data. When gainmap_item_codec_config_size is 0, gainmap_item_codec_config is not present.
main_item_codec_config: carries the main image item codec configuration data. When main_item_codec_config_size is 0, main_item_codec_config is not present.
icc_data: carries the **ICC profile** data of the main image as defined in ISO 15076-1 or ICC.1[23]. When has_icc is 0, icc_data is not present.
tmap_icc_data: carries the **ICC profile** data of the optional HDR-related tone-mapped image as defined in ISO 15076-1 or ICC.1[23]. When tmap_has_icc is 0, tmap_icc_data is not present.
gainmap_metadata: Gain map metadata as defined by the **GainmapMetadata** struct in ISO 21496-1. Not present if gainmap_metadata_size is 0.
alpha_item_data: carries the coded sample data of the optional alpha image. When alpha_item_data_size is 0, alpha_item_data is not present.
gainmap_item_data: carries the coded sample data of the optional gain map image. When gainmap_item_data_size is 0, gainmap_item_data is not present.
main_item_data: carries the coded sample data of the main image.
exif_data: specifies the optional **Exif** metadata. When has_exif is set to 0, exif_data is not present.
xmp_data: specifies the optional **XMP** metadata. When has_xmp is set to 0, xmp_data is not present.

6.10.2 Equivalence with MetaBox with version 0

A MetaBox with version 1 has a one-to-one mapping to a MetaBox with version 0. Readers shall treat MetaBox with version 1 as if it were the equivalent MetaBox with version 0 that is transformed from MetaBox with version 1 as specified in this subclause. The equivalent MetaBox with version 0 has its flags set to 0 and the sub-boxes described by the following subclauses.

6.10.2.1 HandlerBox

The equivalent MetaBox with version 0 shall have a HandlerBox with handler_type equal to 'pict'.

6.10.2.2 PrimaryItemBox

The equivalent MetaBox with version 0 shall have a PrimaryItemBox with item_ID set to 1.

6.10.2.3 ItemInfoBox

The equivalent MetaBox with version 0 shall have an ItemInfoBox containing the following entries:

- ItemInfoEntry of version 2 with flags set to 0, item_ID set to 1 and item_type set to infe_type. All other fields are set to null or 0 as appropriate.
- If alpha_item_data_size is not 0, ItemInfoEntry of version 2 with flags set to 1, item_ID set to 2 and item_type set to infe_type. All other fields are set to null or 0 as appropriate.
- If has_gainmap is 1, ItemInfoEntry of version 2 with flags set to 0, item_ID set to 3 and item_type set to 'tmap'. All other fields are set to null or 0 as appropriate.
- If gainmap_item_data_size is not 0, ItemInfoEntry of version 2 with flags set to 1, item_ID set to 4 and item_type set to infe_type. All other fields are set to null or 0 as appropriate.
- If has_exif is 1, ItemInfoEntry of version 2 with flags set to 1, item_ID set to 6 and item_type set to 'Exif'. All other fields are set to null or 0 as appropriate.
- If has_xmp is 1, ItemInfoEntry of version 2 with flags set to 1, item_ID set to 7 and item_type set to 'mime' and content_type set to 'application/rdf+xml'. All other fields are set to null or 0 as appropriate.

6.10.2.4 ItemReferenceBox

The ItemReferenceBox is populated with the following entries:

- If ItemInfoBox has an entry for item_ID 2: Item type reference with referenceType set to 'auxl', from_item_ID set to 2, reference_count set to 1 and to_item_ID set to 1.
- If ItemInfoBox has an entry for item_ID 2 and alpha_is_premultiplied is set to 1: Item type reference with referenceType set to 'prem', from_item_ID set to 1, reference_count set to 1 and to_item_ID set to 2.
- If has_gainmap is 1 and ItemInfoBox has an entry for item_ID 4: item type reference with referenceType set to 'dimg', from_item_ID set to 3, reference_count set to 2 and the following to_item_ID entries: 1, 4.
- If has_gainmap is 1 and ItemInfoBox has zero entry for item_ID 4: item type reference with referenceType set to 'dimg', from_item_ID set to 3, reference_count set to 1 and to_item_ID set to 1.

  EDITORS NOTE:
  ‘tmap’ items currently require 2 inputs. In a future where a codec supports coding the gain map interleaved with the base image in the same coded payload, a ‘tmap’ item would presumably only have one input. We’re leaving the option open here for now for that future.

- If ItemInfoBox has an entry for item_ID 6: item type reference with referenceType set to 'cdsc', from_item_ID set to 6, reference_count set to 1 and to_item_ID set to 1.
- If ItemInfoBox has an entry for item_ID 7: item type reference with referenceType set to 'cdsc', from_item_ID set to 7, reference_count set to 1 and to_item_ID set to 1.

If the resulting ItemReferenceBox contains at least one entry, it shall be added to the equivalent MetaBox with version 0. An empty ItemReferenceBox shall be ignored.

6.10.2.5 EntityToGroupBox

If has_gainmap is 1, the equivalent MetaBox with version 0 shall have a GroupsListBox containing a single sub-box. That sub-box is an EntityToGroupBox with grouping_type set to 'altr', version set 0, flags set to 0, group_id set to 5, num_entities_in_group set to 2 and the following entity_id entries: 3, 1.

6.10.2.6 ItemPropertiesBox

The equivalent MetaBox with version 0 shall have an ItemPropertiesBox containing an ItemPropertyContainerBox and an ItemPropertyAssociationBox.

The ItemPropertyContainerBox shall have 32 entries as listed below. Any entry for which the condition is not true is replaced with a FreeSpaceBox.

Entry	Condition	Contents
1	main_item_codec_config_size is not 0	Item property with the type set to codec_config_type and with contents from main_item_codec_config.
2	true	ImageSpatialExtentsProperty with image_width set to width_minus_one + 1 and image_height set to height_minus_one + 1.
3	true	PixelInformationProperty version 1 with field values defined in section 6.10.2.7 of this document.
4	true	ColourInformationBox with colour_type set to 'nclx' and colour_primaries, transfer_characteristics, matrix_coefficients and full_range set to the values from the MetaBox version 1.
5	has_icc is 1	ColourInformationBox with the colour_type set to 'prof' and with ICC_profile contents being icc_data.
6	alpha_item_codec_config_size is not 0	Item property with the type set to codec_config_type and with contents from alpha_item_codec_config.
7	alpha_item_data_size is not 0	AuxiliaryTypeProperty with aux_type set to 'urn:mpeg:mpegB:cicp:systems:auxiliary:alpha'.
8	alpha_item_data_size is not 0	PixelInformationProperty version 1 with field values defined in section 6.10.2.7 of this document.
9	orientation is 2, 4, 5, 6 or 7	ImageRotation property with angle set to 2, 1, 3, 3, 1, respectively
10	orientation is 1, 3, 4 or 6	ImageMirror property with axis set to 1, 0, 0, 0, respectively
11	has_clli is 1	ContentLightLevelBox with body set to clli
12	has_mdcv is 1	MasteringDisplayColourVolumeBox with body set to mdcv
13	has_cclv is 1	ContentColourVolumeBox with body set to cclv
14	has_amve is 1	AmbientViewingEnvironmentBox with body set to amve

15	`has_reve` is 1	`ReferenceViewingEnvironmentBox` with body set to `reve` and `version` and `flags` set to 0
16	`has_ndwt` is 1	`NominalDiffuseWhiteBox` with body set to `ndwt` and `version` and `flags` set to 0
17	`gainmap_item_codec_config` is not 0	Item property with the type set to `codec_config_type` and with contents from `gainmap_item_codec_config`.
18	`gainmap_item_data_size` is not 0	`ImageSpatialExtentsProperty` with `image_width` set to `gainmap_width_minus_one` + 1 and `image_height` set to `gainmap_height_minus_one` + 1.
19	`gainmap_item_data_size` is not 0	`PixelInformationProperty` version 1 with field values defined in section 6.10.2.7 of this document.
20	`gainmap_item_data_size` is not 0	`ColourInformationBox` with `colour_type` set to `'nclx'` and `colour_primaries` and `transfer_characteristics` set to 2, `matrix_coefficients` set to `gainmap_matrix_coefficients` and `full_range` set to `gainmap_full_range`.
21	`has_gainmap` is 1	`ImageSpatialExtentsProperty` equal to entry 2 if `orientation` is 0, 1, 2, 3 and with `image_width` set to `height_minus_one + 1` and `image_height` set to `width_minus_one + 1` for any other orientation.
22	`has_gainmap` is 1 and (`tmap_has_explicit_cicp` is 1 or `tmap_has_icc` is 0)	`ColourInformationBox` with `colour_type` set to `'nclx'` and `colour_primaries`, `transfer_characteristics`, `matrix_coefficients` and `full_range` set to `tmap_colour_primaries`, `tmap_transfer_characteristics`, `tmap_matrix_coefficients` and `tmap_full_range`, respectively.
23	`has_gainmap` is 1 and `tmap_has_icc` is 1	`ColourInformationBox` with the `colour_type` set to `'prof'` and with `ICC_profile` contents being `tmap_icc_data`.
24	`tmap_has_clli` is 1	`ContentLightLevelBox` with body set to `tmap_clli`
25	`tmap_has_mdcv` is 1	`MasteringDisplayColourVolumeBox` with body set to `tmap_mdcv`
26	`tmap_has_cclv` is 1	`ContentColourVolumeBox` with body set to `tmap_cclv`
27	`tmap_has_amve` is 1	`AmbientViewingEnvironmentBox` with body set to `tmap_amve`
28	`tmap_has_reve` is 1	`ReferenceViewingEnvironmentBox` with body set to `tmap_reve` and `version` and `flags` set to 0
29	`tmap_has_ndwt` is 1	`NominalDiffuseWhiteBox` with body set to `tmap_ndwt` and `version` and `flags` set to 0

30	`false`	Reserved
31	`false`	Reserved
32	`false`	Reserved

The `ItemPropertyAssociationBox` shall have the entries below. Any association to a `FreeSpaceBox` shall be dropped.

- Item 1 shall be associated with `ItemPropertyContainerBox` entries:
  - 1, essential
  - 2, non-essential
  - 3, non-essential
  - 4, essential
  - 5, essential
  - 9, essential
  - 10, essential
- If `has_hdr` is 1, item 1 shall be associated with `ItemPropertyContainerBox` entries:
  - 11, non-essential
  - 12, non-essential
  - 13, non-essential
  - 14, non-essential
  - 15, non-essential
  - 16, non-essential
- If `alpha_item_data_size` is not 0, item 2 shall be associated with `ItemPropertyContainerBox` entries:
  - 6, essential
  - 2, non-essential
  - 7, essential
  - 8, non-essential
  - 9, essential
  - 10, essential
- If `has_gainmap` is 1, item 3 shall be associated with `ItemPropertyContainerBox` entries:
  - 21, non-essential
  - 22, essential
  - 23, essential
- If `has_hdr` is 1 and `has_gainmap` is 1, item 3 shall be associated with `ItemPropertyContainerBox` entries:
  - 24, non-essential
  - 25, non-essential
  - 26, non-essential
  - 27, non-essential
  - 28, non-essential
  - 29, non-essential
- If `gainmap_item_data_size` is not 0, item 4 shall be associated with `ItemPropertyContainerBox` entries:
  - 17, essential
  - 18, non-essential
  - 9, essential
  - 10, essential
  - 19, non-essential
  - 20, non-essential

6.10.2.7 PixelInformationProperty

6.10.2.7.1 Reconstruction

The various `PixelInformationProperty` boxes associated with the image items in the file are reconstructed given the arguments

- main_components
- has_alpha
- alpha_is_premultiplied
- subsampling
- is_horizontally_centered
- is_vertically_centered
- pixel_format

Reconstruction happens as follows:

- version set to 1.
- px_flags set to
  - 7 if has_alpha and_alpha_is_premultiplied are both true
  - 5 if has_alpha is true and alpha_is_premultiplied is false
  - 4 otherwise
- `num_channels` set to
  - main_components if has_alpha is false
  - main_components + 1 if has_alpha is true
- each `bits_per_channel` entry set to 16 if `pixel_format` is 0, to 32 if `pixel_format` is 1, to 64 if `pixel_format` is 2, or to `pixel_format`+1 otherwise.
- `channel_label_present` is set to 0 for all entries
- the following entries for channel subsampling and formats
  - the following entry if has_alpha is true (skipped if false)
    - `channel_idc` set to 1
    - `channel_data_type` set to 2 if `pixel_format` is 0, 1 or 2, or to 0 otherwise,
    - `subsampling_type` set to 0
    - `subsampling_location` set to 0
  - the following entry if main_components > 0 (skipped otherwise)
    - `channel_idc` set to 0
    - `channel_data_type` set to 2 if `pixel_format` is 0, 1 or 2, or to 0 otherwise,
    - `subsampling_type` set to 0
    - `subsampling_location` set to 0
  - the following entry, repeated twice, if main_components > 1 (skipped otherwise)
    - `channel_idc` set to 0
    - `channel_data_type` set to 2 if `pixel_format` is 0, 1 or 2, or to 0 otherwise,
    - `subsampling_type` set to
      - 2 if subsampling is 1
      - 1 if subsampling is 2
      - 0 if subsampling is 3
    - `subsampling_location` set to
      - 1 if is_horizontally_centered and is_vertically_centered are both true
      - 3 if is_horizontally_centered is true and is_vertically_centered is false
      - 0 if is_horizontally_centered is false and is_vertically_centered is true
      - 2 if is_horizontally_centered and is_vertically_centered are both false

6.10.2.7.2 Main image PixelInformationProperty

The PixelInformationProperty associated with the main image item is reconstructed as described in section 6.10.2.7.1 with the following arguments:

- main_components = 1 if chroma_subsampling is 0, else 3
- has_alpha = has_alpha is 1 and alpha_item_data_size is 0
- alpha_premultiplied = alpha_is_premultiplied
- subsampling = chroma_subsampling
- is_horizontally_centered = is_horizontally_centered
- is_vertically_centered = is_vertically_centered
- pixel_format = pixel_format

6.10.2.7.3 Alpha auxiliary image PixelInformationProperty

If alpha_item_data_size is not 0, the PixelInformationProperty associated with the alpha auxiliary image item is reconstructed as described in section 6.10.2.7.1 with the following arguments:

- main_components = 0
- has_alpha = 1
- alpha_premultiplied = 0
- subsampling = 0
- is_horizontally_centered = 0
- is_vertically_centered = 0
- pixel_format = pixel_format

6.10.2.7.4 Gain map image PixelInformationProperty

If has_gainmap is not 0, the PixelInformationProperty associated with the gain map image item is reconstructed as described in section 6.10.2.7.1 with the following arguments:

- main_components = 1 if gainmap_chroma_subsampling is 0, else 3,
- has_alpha = 0
- alpha_premultiplied set to 0,
- subsampling = gainmap_chroma_subsampling
- is_horizontally_centered = gainmap_is_horizontally_centered
- is_vertically_centered = gainmap_is_vertically_centered
- pixel_format = gainmap_pixel_format

EDITORS NOTE:
The text for the ‘tmap’ item recommends it being associated with ‘pixi’ property that hints at the approximate bit-depth achieved after combining the base and gain map. We have these choices:

1. Simply ignore this recommendation
2. Code the number of extra bits achieved after reconstruction (i.e. if base is 8 bits, code value “2” to say you get 10 bits out). How does this work with floating point though?
3. Code the full set of fields required to create a v1 ‘pixi’.

6.10.2.8 ToneMapImage metadata

If has_gainmap is 1, tmap_item_data is defined as a data chunk of tmap_item_data_size bytes, being gainmap_metadata_size+1 bytes, containing the ToneMapImage metadata as defined in section 6.6.2.4.2, with the following:

- version set to 0
- GainMapMetadata set to gainmap_metadata

If has_gainmap is 0, tmap_item_data is defined as an empty chunk and tmap_item_data_size as 0 byte.

6.10.2.9 ItemLocationBox

The equivalent MetaBox with version 0 shall have an ItemLocationBox of version 1 or version 2 containing the following entries:

- item_ID 1, with construction_method set to 1, offset set to alpha_item_data_size+tmap_item_data_size+gainmap_item_data_size and length set to main_item_data_size_minus_one+1,
- Optional item_ID 2, with construction_method set to 1, offset set to 0 and length set to alpha_item_data_size,
- Optional item_ID 3, with construction_method set to 1, offset set to alpha_item_data_size, and length set to tmap_item_data_size,
- Optional item_ID 4, with construction_method set to 1, offset set to alpha_item_data_size+tmap_item_data_size, and length set to gainmap_item_data_size,
- Optional item_ID 6, with construction_method set to 1, offset set to main_item_data_size_minus_one+1+alpha_item_data_size+tmap_item_data_size+gainmap_item_data_size, and length set to exif_data_size_minus_one+1,
- Optional item_ID 7, with construction_method set to 1, offset set to main_item_data_size_minus_one+1+alpha_item_data_size+tmap_item_data_size+gainmap_item_data_size+exif_data_size_minus_one+1, and length set to xmp_data_size_minus_one+1,

6.10.2.10ItemDataBox

The equivalent MetaBox with version 0 shall have an ItemDataBox containing the non-empty chunks among alpha_item_data, tmap_item_data, gainmap_item_data, main_item_data, exif_data and xmp_data, concatenated in that order.

4 Add the 'mif3' brand

Add the following new subclause after subclause 10.2.5:

10.2.6 'mif3' structural brand

10.2.6.1 Requirements on files

Files containing the brand 'mif3' as the major brand or in the compatible brands array of the FileTypeBox shall conform to the constraints defined in this subclause.

When the 'mif3' brand is the major brand or present among the compatible brands of the FileTypeBox, the file may be identified by the MIME type defined in Annex P. When the 'mif3' brand is the major brand, the defined file extension and MIME type defined in Annex P should be used.

The boxes listed in Table XX1 are required in a file under the 'mif3' 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. Other file-level boxes shall not be present.

Table XX1 — Required boxes in a file under the `'mif3'` brand

Hierarchy of boxes			Version	Box description
`ftyp`			-	file type and compatibility
`meta`			1	metadata

When the `'mif3'` brand is present as the major brand of the `FileTypeBox`, the minor version of the FileTypeBox shall be 0 or a brand.

10.2.6.2 Requirements on readers

When the `'mif3'` brand is present as the `major_brand` of the `FileTypeBox`, the `minor_version` of the `FileTypeBox` shall be 0 or a brand that is either structurally compatible with the `'mif3'` brand, such as a codec brand complying with the `'mif3'` structural brand, or a brand to which the file conforms after the equivalent `MetaBox` version 0 has been transformed from `MetaBox` version 1 as specified in subclause 6.10.2.

Table XX2 — Boxes to be supported under the `'mif3'` brand in addition to those required for readers of `'mif1'` brand

Hierarchy of boxes			Version	Box description
`meta`			1	metadata

A file containing a `MetaBox` with version 1 shall be treated as if:

- it contained the `'mif1'` brand in the `compatible_brands` array in the `FileTypeBox`,
- it contained the equivalent `MetaBox` with version 0 as described in subclause 6.10.2,
- the `major_brand` of the `FileTypeBox` was the equivalent brand specified by the brand provided in the `minor_version`, if the `major_brand` is `'mif3'` and the `minor_version` is not 0.

5 Add groups of regions

Add the following new subclauses after subclause 11.3.3:

11.3.4 Groups of regions

11.3.4.1 Union of regions entity group

A union of regions entity group (`'unrg'`) indicates the union of all the regions represented by one or more entities describing regions of an image.

Each `entity_id` value in the entity group shall refer to a region item.

All the region items in the union of regions entity group shall be associated with the same image item, inside which the regions are defined, using an item reference of type `'cdsc'` from each region item to the same image item.

If unique IDs are used:

- the union of regions entity group may also be associated with the image item inside which the regions are defined using an item reference of type `'cdsc'` from the union of regions entity group to the image item,
- an annotation may be associated with the union of regions entity group by associating:
  - ○ an item property, using the `ItemPropertyAssociationBox`;

  - ○ a metadata item, using an item reference of type `'cdsc'` from the metadata item to the union of regions entity group;
  - ○ an image item or another entity group, using an item reference of type `'eroi'` from the union of regions entity group to the image item or the other entity group.

11.3.4.2 Compound region entity group

A compound region entity group (`'corg'`) associates one main region item with one or more region items. It indicates an inclusion relationship between a main object covered by regions of a main entity and other objects covered by regions described by one or more other entities, the main object logically including the other objects.

> NOTE 1 For example, a compound region entity group can be used to associate a main region corresponding to a body with regions corresponding to body parts (e.g., the head, legs or arms of the body) to indicate that the body is logically including the body parts.

The entities in a compound region entity group shall be region items.

The number of entities in a compound region entity group shall be at least 2. The first `entity_id` value shall indicate the main region item. It indicates the region covering the main object that is logically including the objects covered by the regions described by the second and following `entity_id`s.

This inclusion relationship does not convey information at the geometry level. A main region signalled as including others regions by a compound region entity group may or may not geometrically include the other regions.

> NOTE 2 For example, the main region item corresponding to the first `entity_id` value can represent a bounding box or a region encompassing partially the regions described by the region items corresponding to the second and following `entity_id` values.

All the region items in the compound region entity group shall be associated with the same image item, inside which the regions are defined, using an item reference of type `'cdsc'` from each region item to the same image item.

If unique IDs are used:

- the compound region entity group may also be associated with the image item, inside which the regions are defined, using an item reference of type `'cdsc'` from the compound region entity group to the image item,
- an annotation may be associated with the compound region entity group by associating:
  - ○ an item property, using the `ItemPropertyAssociationBox`;
  - ○ a metadata item, using an item reference of type `'cdsc'` from the metadata item to the compound region entity group;
  - ○ an image item or another entity group, using an item reference of type `'eroi'` from the union of regions entity group to the image item or the other entity group.

6 Annex changes

Replace all instances of the following text in Annex J:

```
MetaBox 'meta': (container)
```

and all instances of the following text in Annex J:

```
MetaBox: (container)
```

with the following text:

```
MetaBox 'meta' version 0 flags 0x000000:   (container)
```

Add the following new clause after clause J.6:

J.7 Single image

A file with a single coded image item could be structured as follows:

```
FileTypeBox 'ftyp': major_brand='mif3', minor_version='heic'
MetaBox 'meta' version 1 flags 0x001780:   (container)
   width_minus_one=W
   height_minus_one=H
   main_item_codec_config_size=C,
   main_item_data_size_minus_one=D,
   hvcC (with length C)
   HEVC Image (with length D+1)
```

Rename L.4.1 as follows:

L.4.1 'mif1'-compliant VVC image and image collection brands

Add subclause L.4.4 as follows:

L.4.4 'mif3'-compliant VVC image and image collection brands

L.4.4.1 Requirements on files

Files shall include 'mif3' as the major_brand and the brand 'vvi3' as the minor_version in the FileTypeBox and conform to the specifications in 10.2.6.1 of this document ('mif3' structural brand).

The 'vvi3' brand defines the coded image item type to be 'vvc1' as defined in L.2.2.1.2 and the alpha_item_codec_config (if present), gainmap_item_codec_config (if present) and main_item_codec_config to have the format defined in L.4.4.3 of this document.

Files that include 'mif3' as the major_brand and the brand 'vvi3' as the minor_version in the FileTypeBox shall comply with the specifications in L.2 and shall additionally be constrained as follows:

- has_explicit_codec_types in the MetaBox with version 1 shall be equal to 0,
- pixel_format in the MetaBox with version 1 shall be between 7 and 15 inclusive,
- If gainmap_item_data_size in the MetaBox with version 1 is not 0, gainmap_pixel_format in the MetaBox with version 1 shall be between 7 and 15 inclusive,
- The data of each codec configuration property shall use the syntax defined in subclause L.4.2.3.2 of this document,
- Each coded image item shall contain a single IDR picture,
- The data of each coded image item shall consist of only one NAL unit excluding the length and NAL unit header fields, which are inferred by the reader as specified in subclause L.4.2.2 of this document,
- The equivalent file as specified in L.4.4.2 shall conform to the 'vvic' brand.

L.4.4.2 Requirements on readers

The requirements on readers specified in L.4.1.3 and 10.2.6.1 of this document (`'mif3'` structural brand) shall be supported.

As a response to `'mif3'` as the `major_brand` and the brand `'vvi3'` as the `minor_version` in the `FileTypeBox`, the readers shall treat the file as if an equivalent file were created containing `FileTypeBox` with `'vvic'` as the equivalent `major_brand` and the equivalent `MetaBox` version 0 derived with the following additional operations in addition to those specified in subclause 6.10.2:

- infer `infe_type` to be equal to `'vvc1'`,
- infer `codec_config_type` to be equal to `'vvcC'`,
- expand the data of each codec configuration property from the compact syntax to the full structure as defined in section L.4.4.3.4 of this document,
- expand the data of each image item so that the NAL unit length is inferred from the item length, and so that the NAL unit header fields `nal_unit_type`, `nuh_temporal_id_plus1` and `nuh_layer_id` are set to 8 (`IDR_N_LP`), 1 and 0, respectively.

L.4.4.3 Compact VVC decoder configuration

L.4.4.3.1 Definition

This subclause specifies the compact decoder configuration information for ISO/IEC 23090-3 video content. The compact decoder configuration provides essential parameters that are relevant for still images.

L.4.4.3.2 Syntax

```
aligned(8) class CompactVvcDecoderConfigurationRecord {
   unsigned int(2) lengthSizeMinusOne;
   unsigned int(1) ptl_present_flag;
   if (ptl_present_flag) {
      VvcPTLRecord(1) native_ptl;
      if (native_ptl.ptl_multilayer_enabled_flag == 1)
         unsigned int(3) ols_idx;
   }
   unsigned int(1) nal_units_present_flag;
   if (nal_units_present_flag) {
      unsigned int(1) array_completeness;
      unsigned int(8)            vps_nal_unit_length;
      bit(8*vps_nal_unit_length) vps_nal_unit;
      unsigned int(8)            sps_nal_unit_length;
      bit(8*sps_nal_unit_length) sps_nal_unit;
      unsigned int(8)            pps_nal_unit_length;
      bit(8*pps_nal_unit_length) pps_nal_unit;
      unsigned int(1) additional_nal_unit_flag;
      if (additional_nal_unit_flag) {
         unsigned int(3) num_aps_nal_unit;
         unsigned int(3) num_sei_nal_unit;
         for (i=0; i< num_aps_nalus; i++) {
            unsigned int(8)            aps_nal_unit_length;
            bit(8*aps_nal_unit_length) aps_nal_unit;
         }
         for (i=0; i< num_sei_nalus; i++) {
            unsigned int(8)            sei_nal_unit_length;
            bit(8*sei_nal_unit_length) sei_nal_unit;
         }
      }
   }
   trailing_bits();
}
```

L.4.4.3.3 Semantics

nal_units_present_flag: indicates that NAL units are present in the decoder configuration.

vps_nal_unit_length: indicates the length in bytes of the NAL unit. When equal to 0, the VPS NAL unit is not present.

vps_nal_unit: contains the VPS NAL unit as specified in ISO/IEC 23090-3.

sps_nal_unit_length: indicates the length in bytes of the NAL unit. When equal to 0, the SPS NAL unit is not present.

sps_nal_unit: contains the SPS NAL unit as specified in ISO/IEC 23090-3.

pps_nal_unit_length: indicates the length in bytes of the NAL unit. When equal to 0, the PPS NAL unit is not present.

pps_nal_unit: contains the PPS NAL unit as specified in ISO/IEC 23090-3.

additional_nal_unit_flag: equal to 1 indicates the presence of additional NAL units in the decoder configuration record. additional_nal_unit_flag equal to 0 indicates the absence of additional NAL units in the decoder configuration record.

num_aps_nal_unit: indicates the number of APS NAL units included in the configuration record for the referenced CVS.

num_sei_nal_unit: indicates the number of SEI NAL units included in the configuration record for the referenced CVS.

aps_nal_unit_length: indicates the length in bytes of the APS NAL unit.
aps_nal_unit: contains the APS NAL unit as specified in ISO/IEC 23090-3.
sei_nal_unit_length: indicates the length in bytes of the SEI NAL unit.
sei_nal_unit: contains the SEI NAL unit as specified in ISO/IEC 23090-3.
trailing_bits: padding bits to ensure payloads are 8-bit aligned. Shall be 0.

The semantics of the other parameters are the same as for VvcDecoderConfigurationRecord as defined in ISO/IEC 14496-15.

L.4.4.3.4 Equivalence with the VVC decoder configuration

CompactVvcDecoderConfigurationRecord shall be considered equivalent to VvcDecoderConfigurationRecord as defined in ISO/IEC 14496-15 with the following fields:

- if ptl_present_flag is set to 1:
    - num_sublayers is set to 1,
    - constant_frame_rate is set to 1,
    - if the codec configuration property is associated with the main image item:
        - chroma_format_idc is set to the value of the chroma_subsampling field from the MetaBox with version 1,
    - if the codec configuration property is associated with the alpha auxiliary image item:
        - chroma_format_idc is set to 0,
    - if the codec configuration property is associated with the main image item or with the alpha auxiliary image item:
        - bit_depth_minus8 is set to the value minus 7 of the pixel_format field from the MetaBox with version 1,
        - max_picture_width is set to the value plus 1 of the width_minus_one field from the MetaBox with version 1,
        - max_picture_height is set to the value plus 1 of the height_minus_one field from the MetaBox with version 1,
    - if the codec configuration property is associated with the gain map image item:
        - bit_depth_minus8 is set to the value minus 7 of the gainmap_pixel_format field from the MetaBox with version 1,
        - max_picture_width is set to the value plus 1 of the gainmap_width_minus_one field from the MetaBox with version 1,
        - max_picture_height is set to the value plus 1 of the gainmap_height_minus_one field from the MetaBox with version 1,
    - avg_frame_rate is set to 0,
- if nal_units_present_flag is set to 1:
    - num_of_arrays is set to the number of entries below:
        - if vps_nal_unit_length is not 0, there is a VPS NAL unit array with:
            - NAL_unit_type set to 14 (VPS_NUT as defined in ISO/IEC 23090-3),
            - num_nalus set to 1,
            - nal_unit_length set to vps_nal_unit_length,
            - nal_unit set to vps_nal_unit.
        - if sps_nal_unit_length is not 0, there is a SPS NAL unit array with:
            - NAL_unit_type set to 15 (SPS_NUT as defined in ISO/IEC 23090-3),
            - num_nalus set to 1,
            - nal_unit_length set to sps_nal_unit_length,
            - nal_unit set to sps_nal_unit.
        - if pps_nal_unit_length is not 0, there is a PPS NAL unit array with:

        - `NAL_unit_type` set to 16 (`PPS_NUT` as defined in ISO/IEC 23090-3),
        - `num_nalus` set to 1,
        - `nal_unit_length` set to `pps_nal_unit_length`,
        - `nal_unit` set to `pps_nal_unit`.
    - ■ if `additional_nal_unit_flag` is set to 1 and `num_aps_nal_unit` is not 0, there is a prefix APS NAL unit array with:
        - `NAL_unit_type` set to 17 (`PREFIX_APS_NUT` as defined in ISO/IEC 23008-2),
        - `num_nalus` set to `num_aps_nal_unit`, and for each prefix APS NAL unit:
            - ○ `nal_unit_length` is set to `aps_nal_unit_length`,
            - ○ `nal_unit` is set to `aps_nal_unit`.
    - ■ if `additional_nal_unit_flag` is set to 1 and `num_sei_nal_unit` is not 0, there is a prefix SEI NAL unit array with:
        - `NAL_unit_type` set to 23 (`PREFIX_SEI_NUT` as defined in ISO/IEC 23008-2),
        - `num_nalus` set to `num_sei_nal_unit`, and for each prefix SEI NAL unit:
            - ○ `nal_unit_length` is set to `sei_nal_unit_length`,
            - ○ `nal_unit` is set to `sei_nal_unit`.
- the other parameters are carried over as is, and repeated if needed.

Add the following new annex after annex O:

Annex P
(normative)
MetaBox with version 1 MIME type registration

P.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 6.2.1 (`MetaBox` with version 1), the MIME type defined in this annex should be used. When such a brand is a major or compatible brand, this MIME type may also be used.

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

P.2 Registration

```
MIME media type name: image

MIME subtype name: hif2

       The semantics of the subtype are as follows:

        hif2: High efficiency image file containing one image item
        possibly with alpha plane in any coding format. This subtype
        name may be ‘hif2’ only if the file conforms to the
        requirements of the ‘mif3’ 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 ISO-defined (MPEG) video
                codecs, the format of a list item included in the
                value of the codecs parameter is 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. 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
http://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
http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Additional information:

Magic number(s): none

File extension(s): himg, hmg (or hig)

Intended usage: Common
```

P.3 Examples (informative)

```
Content-Type:

       image/hif2; codecs="vvc1.1.L51.CQA"; profiles=”mif3”
```

An image file with low-overhead HEIF container containing one VVC-coded image using Main 10 profile, Main Tier, Level 3.1.

```
Content-Type:

       image/hif2; codecs="hvc1.A1.80.L93.B0"; profiles=”mif3”
```

An image file with low-overhead HEIF container containing one HEVC-coded image.