**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

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

ISO copyright office

Case postale 56 • CH-1211 Geneva 20

Tel. + 41 22 749 01 11

Fax + 41 22 749 09 47

E-mail copyright@iso.org

Web www.iso.org

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

**Contents**

[Information technology — Coding of audio-visual objects — Part 12: ISO base media file format — Amendment X: MetaBox version 1 1](#__RefHeading___Toc3780_1483204132)

[1 Allow for MetaBox with version 1 1](#__RefHeading___Toc3784_1483204132)

[2 Declare MetaBox with version 1 1](#__RefHeading___Toc3800_1483204132)

[3 Annex changes 2](#__RefHeading___Toc3798_1483204132)

[Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format — Amendment 3: Low-overhead Image Format 1](#__RefHeading___Toc3782_1483204132)

[1 Allow for MetaBox with version 1 1](#__RefHeading___Toc3796_1483204132)

[6.2.1 Files with MetaBox version 1 1](#__RefHeading___Toc3816_1483204132)

[2 Add PixelInformationProperty with version 1 1](#__RefHeading___Toc3794_1483204132)

[3 Define MetaBox with version 1 3](#__RefHeading___Toc3792_1483204132)

[6.10 Processing MetaBox with version 1 3](#__RefHeading___Toc3802_1483204132)

[6.10.1 Content of MetaBox with version 1 3](#__RefHeading___Toc3830_1483204132)

[6.10.1.1 Definition 3](#__RefHeading___Toc3832_1483204132)

[6.10.1.2 Syntax 3](#__RefHeading___Toc3872_1483204132)

[6.10.1.3 Semantics 6](#__RefHeading___Toc3870_1483204132)

[6.10.2 Equivalence with MetaBox with version 0 10](#__RefHeading___Toc3828_1483204132)

[6.10.2.1 HandlerBox 10](#__RefHeading___Toc3868_1483204132)

[6.10.2.2 PrimaryItemBox 10](#__RefHeading___Toc3866_1483204132)

[6.10.2.3 ItemInfoBox 10](#__RefHeading___Toc3864_1483204132)

[6.10.2.4 ItemReferenceBox 11](#__RefHeading___Toc3862_1483204132)

[6.10.2.5 EntityToGroupBox 11](#__RefHeading___Toc3860_1483204132)

[6.10.2.6 ItemPropertiesBox 11](#__RefHeading___Toc3858_1483204132)

[6.10.2.7 PixelInformationProperty 14](#__RefHeading___Toc3856_1483204132)

[6.10.2.7.1 Reconstruction 14](#__RefHeading___Toc3874_1483204132)

[6.10.2.7.2 Main image PixelInformationProperty 15](#__RefHeading___Toc3880_1483204132)

[6.10.2.7.3 Alpha auxiliary image PixelInformationProperty 15](#__RefHeading___Toc3878_1483204132)

[6.10.2.7.4 Gain map image PixelInformationProperty 15](#__RefHeading___Toc3876_1483204132)

[6.10.2.8 ToneMapImage metadata 16](#__RefHeading___Toc3854_1483204132)

[6.10.2.9 ItemLocationBox 16](#__RefHeading___Toc3852_1483204132)

[6.10.2.10 ItemDataBox 16](#__RefHeading___Toc3850_1483204132)

[4 Add the 'mif3' brand 17](#__RefHeading___Toc3790_1483204132)

[10.2.6 'mif3' structural brand 17](#__RefHeading___Toc3826_1483204132)

[10.2.6.1 Requirements on files 17](#__RefHeading___Toc3848_1483204132)

[10.2.6.2 Requirements on readers 17](#__RefHeading___Toc3846_1483204132)

[5 Add groups of regions 17](#__RefHeading___Toc3788_1483204132)

[11.3.4 Groups of regions 18](#__RefHeading___Toc3824_1483204132)

[11.3.4.1 Union of regions entity group 18](#__RefHeading___Toc3844_1483204132)

[11.3.4.2 Compound region entity group 18](#__RefHeading___Toc3842_1483204132)

[6 Annex changes 19](#__RefHeading___Toc3786_1483204132)

[J.7 Single image 19](#__RefHeading___Toc3814_1483204132)

[L.4.1 'mif1'-compliant VVC image and image collection brands 19](#__RefHeading___Toc3812_1483204132)

[L.4.4 'mif3'-compliant VVC image and image collection brands 19](#__RefHeading___Toc3810_1483204132)

[L.4.4.1 Requirements on files 19](#__RefHeading___Toc3822_1483204132)

[L.4.4.2 Requirements on readers 20](#__RefHeading___Toc3820_1483204132)

[L.4.4.3 Compact VVC decoder configuration 20](#__RefHeading___Toc3818_1483204132)

[L.4.4.3.1 Definition 20](#__RefHeading___Toc3840_1483204132)

[L.4.4.3.2 Syntax 21](#__RefHeading___Toc3838_1483204132)

[L.4.4.3.3 Semantics 21](#__RefHeading___Toc3836_1483204132)

[L.4.4.3.4 Equivalence with the VVC decoder configuration 22](#__RefHeading___Toc3834_1483204132)

[P.1 General 23](#__RefHeading___Toc3808_1483204132)

[P.2 Registration 23](#__RefHeading___Toc3806_1483204132)

[P.3 Examples (informative) 24](#__RefHeading___Toc3804_1483204132)

**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](http://www.iso.org/directives) or [www.iec.ch/members\_experts/refdocs](http://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](http://www.iso.org/patents)) or the IEC list of patent declarations received (see [patents.iec.ch](https://patents.iec.ch/)).

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

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

##### 6.10.2.10 ItemDataBox

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\_ids.

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.