ISO/IEC JTC 1/SC 29 N 00381

Date: 2022-01-31

ISO/IEC 23001‑17:202X(E)

ISO/IEC JTC 1/SC 29/WG 3

Secretariat: JISC

Information technology — MPEG Systems technologies — Part17: Uncompressed video in ISO Base Media File Format

Document type: Draft International Standard

Document subtype:

Document language: E

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

**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:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manager of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

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

Violators may be prosecuted.

Contents Page

[Foreword 5](#_Toc87519721)

[1 Scope 1](#_Toc87519722)

[2 Normative references 1](#_Toc87519723)

[3 Terms and definitions 1](#_Toc87519724)

[4 Technical overview 2](#_Toc87519725)

[5 Uncompressed video formats 3](#_Toc87519726)

[5.1 Overview 3](#_Toc87519727)

[5.2 Generic uncompressed video 3](#_Toc87519728)

[5.2.1 Generic uncompressed video 3](#_Toc87519729)

[5.2.2 Generic uncompressed video configuration 3](#_Toc87519730)

[5.3 Predefined uncompressed video configurations 8](#_Toc87519731)

[5.3.1 Overview 8](#_Toc87519732)

[5.3.2 Predefined configurations 9](#_Toc87519734)

[5.3.3 Other common configurations (informative) 10](#_Toc87519735)

[6 Video sample entry extensions 10](#_Toc87519736)

[6.1.1 Overview 10](#_Toc87519737)

[6.1.2 Chroma Location 11](#_Toc87519738)

[6.1.3 Interlaced Content Type 11](#_Toc87519739)

[6.1.4 Frame Packing Information 12](#_Toc87519741)

[6.1.5 Disparity Information 12](#_Toc87519743)

[6.1.6 Depth Mapping Information 13](#_Toc87519744)

[7 Multiple track storage 14](#_Toc87519745)

[7.1 Overview 14](#_Toc87519746)

[7.2 Uncompressed video track group 14](#_Toc87519747)

Foreword

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

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

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](https://www.iso.org/iso-standards-and-patents.html)) or the IEC list of patent declarations received (see [http://patents.iec.ch](http://patents.iec.ch/)).

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

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

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

A list of all parts in the ISO/IEC 23001 series can be found on the ISO website.

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

Information technology — MPEG Systems technologies — Part 17: Uncompressed video in ISO Base Media File Format

# Scope

This document defines how uncompressed 2D image or video data is carried in files in the family of standards based on the ISO base media file format. This includes monochromatic data, colour data, transparency (alpha) information, depth information.

The primary goal of this specification is to allow exchange of uncompressed video or image data while relying on the information set provided by the ISO base media file format, such as timing, color space, sample aspect ratio...

# Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

ISO/IEC 23008-12, Information technology – *High efficiency coding and media delivery in heterogeneous environments – Part 12: Image File Format* (HEIF)

ISO/IEC 23091-2, *Information technology —Coding-independent code points — Part 2: Video*

ISO/IEC 23002-3*, Information technology — MPEG video technologies — Part 3: Representation of auxiliary video and supplemental information*

IEEE 754*, IEEE Std 754™-2008, IEEE Standard for Floating-Point Arithmetic*

# Terms and definitions

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

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

— ISO Online browsing platform: available at [https://www.iso.org/obp](https://www.iso.org/obp/ui)

— IEC Electropedia: available at <http://www.electropedia.org/>

sample

media sample when the uncompressed data is described by a media track, or the payload of an item when the uncompressed data is described by an image item

Note 1 to entry: Media sample as defined in 14496-12.

Note 2 to entry: Payload of an item as defined in 14496-12.

component

one-dimensional information

Note to entry: In this specification, a component may describe visible information such as luminance or chroma, or other information usually not intended for direct display such as depth or transparency.

uncompressed video

One or more frame for which per-component values are coded independently from any component in this or any other frame

Note to entry: In this specification, the uncompressed term is used with some video formats applying some compression on one or more components, such as sub-sampling; however, data access to components for such formats is still independent from other components or frames.

frame

two-dimensional array of pixels contained within a sample

pixel

point associated with one or more components

packing

byte positioning within the sample data of components values

# Technical overview

ISO/IEC 14496-12 shall be used as framework for the video extensions described in this document.

ISO/IEC 23008-12 shall be used as framework for the image extensions described in this document.

The extensions included in this document provide the necessary specifications to carry uncompressed video and images in these frameworks.

Media tracks and media samples may be associated with meta-data information using the various tools defined in ISO/IEC 14496-12, such as sample group descriptions, MetaBox, metadata tracks and sample auxiliary information.

Media items may be associated with meta-data information using the various tools defined in ISO/IEC 23008-12, such as descriptive item properties.

# Uncompressed video formats

## Overview

Uncompressed video formats may be stored in ISO base media formats using a generic uncompressed video description defined hereafter.

For media tracks, uncompressed video formats shall be described by a sample entry extending the VisualSampleEntry. The handler type associated with the track is usually 'vide', 'auxv' or 'pict' but derived specifications may introduce new handler types. The width and height fields of the sample entry shall document the exact frame dimension, in pixels, of any sample of the video stream that is described by this sample entry.

For image items, uncompressed images shall be associated with a ImageSpatialExtentsProperty associated whose image\_width and image\_height fields shall document the exact frame dimension, in pixels, of the image.

For video or image data intended to be displayed, the tools defined in ISO/IEC 14496-12 and ISO/IEC 23008-12 should be used whenever applicable, namely to specify pixel aspect ratio, colour information, display aperture, content light level, etc.

An uncompressed video media sample or item consists in one frame, the first byte of the sample data being the first byte of the first component stored for that frame.

The size in bytes of the associated media sample or item shall be the size in bytes of all the components values as documented by the uncompressed video configuration.

For media tracks, each uncompressed video sample is a sync sample. Consequently, the SyncSampleBox should be absent from the track.

## Generic uncompressed video

### Generic uncompressed video

A generic uncompressed video media shall use a video sample entry with codingname 'uncv' containing one UncompressedVideoConfigBox.

A generic uncompressed image item shall use the item\_type 'uncv', an associated UncompressedVideoConfigBox essential item property, i.e. essential shall be equal to 1 for an UncompressedVideoConfigBox item property associated with an image item of type 'uncv'.

### Generic uncompressed video configuration

#### Definition

Box Type: 'uncC'   
Container: Video sample entry, ItemPropertyContainerBox  
Mandatory: Yes, if codingname of the sample entry is 'uncv' or if the item\_type of the item is 'uncv'  
Quantity: Zero or one per video sample entry or per item

The UncompressedVideoConfigBox is used to document the format of the image data. This box appears in a video sample entry of type 'uncv', or is associated with an item of type 'uncv'.

The UncompressedVideoConfigBox allows describing uncompressed video frames that are composed of one or more components. RGB or YUV formats are typical examples of such uncompressed video samples for which each colour component is a component of the uncompressed video sample. Other component types that associate a value to one or more pixels of the frame can also be described. For example, disparity, depth and transparency are other types of components.

The UncompressedVideoConfigBox may indicate a profile for this description, allowing faster identification of the class of video data used. Profiles are defined in subclause 5.3.

Each frame is made of one or more components, where each component is assigned its own type (e.g., Y, U, V), as detailed in Table 1.

NOTE In some cases, two or more components in a UncompressedVideoConfigBox may have the same type, and file readers may require additional information to process the pixel data. How such information is provided to the file reader is out of scope of this specification.

The order in which components are listed indicates the order in which components are stored within the sample data, and is further detailed in Table 4.

Table 1 - Component types

|  |  |
| --- | --- |
| Value | Description |
| 0 | Visible monochrome component |
| 1 | (Y) Luma component |
| 2 | (Cb / U) Chroma component |
| 3 | (Cr / V) Chroma component |
| 4 | (R) Red component |
| 5 | (G) Green component |
| 6 | (B) Blue component |
| 7 | (W) White component |
| 8 | (A) Alpha/transparency component |
| 9 | Depth component |
| 10 | Disparity component |
| 11 | Padded component (unused bits/bytes) |
| 12 | Gamma |
| 13 | X-ray |
| 14 | VUV (Vacuum Ultraviolet) |
| 15 | UVC (Ultraviolet C) |
| 16 | UVB (Ultraviolet B) |
| 17 | UVA (Ultraviolet A) |
| 18 | NIR (Near Infra-Red) |
| 19 | SWIR (Short Wavelength Infra-Red) |
| 20 | MWIR (Middle Wavelength Infra-Red) |
| 21 | LWIR (Long Wavelength Infra-Red) |
| 21-0xFFFE | ISO/IEC reserved |
| 0xFFFF | User-defined |

For a given component, the binary representation of each value is given by its component\_format as defined in Table 2, and is uniform across all pixels.

Each pixel of the frame contains exactly the number of components defined.

A component value may be coded for a pixel or for a group of pixels, depending on the sampling\_type. For every pixel, or for every group of pixels described by sampling\_type, a given component value is coded in the same way.

NOTE Components may use different coding bit length, either explicitly by using different component\_format, or implicitly as indicated by the sampling\_type.

Table 2 - Component formats

|  |  |
| --- | --- |
| Value | Description |
| 0 | ISO/IEC reserved |
| 1 – 32 | Component value is an unsigned integer coded on the given number of bits using little-endian byte ordering if more than 8 bits. |
| 33 – 55 | Component value is an unsigned integer coded on the given number of bits minus 23 bits using big-endian byte ordering |
| 56-86 | Component value is a signed integer coded on the given number of bits minus 54 using little-endian byte ordering if more than 8 bits. |
| 89-119 | Component value is a signed integer coded on the given number of bits minus 87 using big-endian byte ordering. |
| 120 | Component value is a binary float number coded on 32 bits as defined by IEEE 754 |
| 121 | Component value is a binary float number coded on 64 bits as defined by IEEE 754 |
| 122 – 255 | ISO/IEC reserved for future definition |

When a component with a component\_type value of 1, 2 or 3 is present, there should not be any component with a component\_type value of 4, 5, 6 or 7 present.

When a component with a component\_type value of 4, 5, 6 or 7 is present, there should not be any component with a component\_type value of 1, 2 or 3 present.

Some formats, such as YUV video, do not use the same 2D resolution for each of the component of the frame. This is indicated by the sampling\_type field.

The sampling\_type field only covers common sub-sampling or pattern-based layout of components. This field shall be ignored when interpreting component with component\_type different from 2, 3, 4, 5, 6 or 7; for these types of components, sampling\_type is always interpreted as having the value 0, i.e. the 2D dimension of the component is the one given in the sample entry or in the image item.

Table 3 – Components sampling types

|  |  |
| --- | --- |
| Value | Definition |
| 0 | No subsampling, all components have the same visual size  (e.g. RGB, YYV 4:4:4) |
| 1 | YCbCr 4:2:2  Shall only be used if component\_type values 1, 2 and 3 are present. |
| 2 | YCbCr 4:2:0  Shall only be used if component\_type values 1, 2 and 3 are present.  The packing\_type field shall be set to 0, 1 or 4. |
| 3 | YCbCr 4:1:1  Shall only be used if component\_type values 1, 2 and 3 are present. |
| 4 | Bayer pattern  A block of 2x2 pixels with top-left pixel aligned on even line and even column is made of components  {C1, none, none } { none, C2, none }  { none, C2, none } { none, none, C3}  with C1, C2, C3 the first three components listed and 'none' indicating absence of component value. The component\_format values shall be the same for C1, C2, and C3.  The packing\_type field shall be set to 0.  The packing of the first three components listed shall be done as described by packing\_type value 2.  The packing of remaining components, if any, shall be done as described by packing\_type value 0, the first additional component value being located at the byte immediately following the last component value C3. |
| 5-0xFFFF | reserved |

The packing of pixels is indicated by packing\_type, as defined in Table 4.

The packing type applies to all listed components unless explicitly defined otherwise by the sampling\_type.

Table 4 - Components packing types

|  |  |
| --- | --- |
| Value | Packing description |
| 0 | Planar (No packing), byte-aligned  The first value of component *i+1* is located immediately after the last value of component *i* (“planar” mode). Values of a component each start on a byte boundary, and with unused bits, if any, set to 0: individual values that do not fall on natural byte boundaries, such as 10, 12 or 14 bits values, shall be aligned to the LSB, with padding occurring at the MSBs. |
| 1 | Planar (No packing), not byte-aligned  The first value of component *i+1* is located immediately after the last value of component *i* (“planar” mode). Values of a component follow each other with no padding bits in-between them.  Integer component values shall be stored starting with the most significant bit up to the least significant bit, and endianness indicated by component\_format shall be ignored. |
| 2 | component packing, byte aligned  for a given pixel, the component values are located one after the other, in their order of declaration, each component value starting on a byte boundary and unused bits, if any, set to 0 (i.e. individual values that do not fall on natural byte boundaries, such as 10, 12 or 14 bits values, shall be aligned to the LSB, with padding occurring at the MSBs). |
| 3 | component packing, not byte-aligned  for a given pixel, the component values are located one after the other, in their order of declaration, with no padding bits in-between them.  Integer component values shall be stored starting with the most significant bit up to the least significant bit, and endianness indicated by component\_format shall be ignored. |
| 4 | Mixed planar/packed, byte-aligned  The packing of the first component listed shall be done as defined by packing\_type value 0.  The packing of components 2 and 3 shall be done as defined by packing\_type value 2, the first value of component 2 being located at the byte immediately following the last value of the first component.  The packing of remaining components, if present, shall be done as defined by packing\_type value 0, the first additional component value being located at the byte immediately following the last value of component 3.  NOTE In this mode, all component values are byte-aligned. This mode is typically used to store YUV 420 data with planar Y and packed U and V components. |
| 5 | Planar (No packing), byte-aligned LSB padding  The first value of component *i+1* is located immediately after the last value of component *i* (“planar” mode). Values of a component each start on a byte boundary, and with unused bits, if any, set to 0: individual values that do not fall on natural byte boundaries, such as 10, 12 or 14 bits values, shall be aligned to the MSB, with padding occurring at the LSBs. |
| 6 | component packing, byte aligned LSB padding  for a given pixel, the component values are located one after the other, in their order of declaration, each component value starting on a byte boundary and unused bits, if any, set to 0 (i.e. individual values that do not fall on natural byte boundaries, such as 10, 12 or 14 bits values, shall be aligned to the MSB, with padding occurring at the LSBs). |
| 7-0xFF | Reserved |

For packing\_type values 2 and 3, if sampling\_type value is 1 (4:2:2 sampling), the Y components of two consecutive horizontal pixels are present in the packing pattern, once at the location indicated in the component listing, and once after the component following the Y component in the component listing. For example, if the component list indicates 2, 1, 3 (Cb, Y, Cr) and a packing\_type value 2, the packing pattern of components will be 2, 1, 3, 1 (Cb, Y0, Cr, Y1), describing two pixels.

For packing\_type values 2 and 3, if sampling\_type value is 3 (4:1:1 sampling), the Y components of four consecutive horizontal pixels are present in the packing pattern, once at the location indicated in the component listing, once at the location immediately following the previous Y component, once after the component following the Y component in the component listing and once at the location immediately following the previous Y component. For example, if the component list indicates 2, 1, 3 (Cb, Y, Cr) and a packing\_type value 2, the packing pattern of components will be 2, 1, 1, 3, 1, 1 (Cb, Y0, Y1, Cr, Y2, Y3), describing four pixels.

#### Syntax

aligned(8) class UncompressedVideoConfigBox extends FullBox('uncC', 0, 0) {  
 unsigned int(32) profile;   
 unsigned int(16) num\_component;   
 {  
 unsigned int(16) component\_type;  
 if (component\_type==0xFFFF) {

utf8string component\_type\_uri;  
 }

unsigned int(8) component\_format;  
 } [num\_components];  
 unsigned int(16) sampling\_type;  
 unsigned int(8) packing\_type;

}

#### Semantics

profile indicates the predefined configuration for this uncompressed sample entry. Value 0 indicates this sample entry may or may not be compliant to a profile. If not 0, all remaining fields in the box shall have the values indicated by the profile.

num\_components indicates the number of components present in the sample

component\_type indicates the type of the component, as defined in Table 1.

component\_type\_uri indicates a URI describing the user-defined component type.

component\_format indicates the format of the component, as defined in Table 2

sampling\_type indicates the sampling type of the components, as defined in Table 3

packing\_type indicates the packing type of the components, as defined in Table 4

## Profiles for uncompressed video configurations

### Overview

The four-character codes defined in this subclause identify common video formats for which a profile is defined. This profile may be indicated in the UncompressedVideoConfigBox to simplify pixel format detection by file processors.

Profiles only provide identification of pixel format used in frame (sampling type and packing). Image reconstruction might need additional information, such as colour information or chroma location, present in children boxes of the sample entry.

NOTE These profiles are introduced to document existing practices in the industry prior to the definition of this specification.

### Predefined configurations

In Table 5 below, the values for the UncompressedVideoConfigBox are listed as:

{profile, num\_components, [{component\_type, component\_format}], sampling\_type, packing\_type}

And the array of {component\_type, component\_format} is given in component order.

Table 5 - Predefined uncompressed video formats

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Profile identifier | Description | | Field values for UncompressedVideoConfigBox | |
| '2vuy' | 8 bits YUV 422 packed  Cb Y0 Cr Y1 | | {'2vuy', 3, [ {2, 8}, {1, 8}, {3, 8} ], 1, 2} | |
| 'yuv2' | 8 bits YUV 422 packed  Y0 Cb Y1 Cr | | {'yuv2', 3, [ {1, 8}, {2, 8}, {3, 8} ], 1, 2} | |
| 'yvyu' | 8 bits YUV 422 packed  Y0 Cr Y1 Cb | | {'yvyu', 3, [ {1, 8}, {3, 8}, {2, 8} ], 1, 2} | |
| 'vyuy' | 8 bits YUV 422 packed  Cr Y0 Cb Y1 | | {'vyuy', 3, [ {3, 8}, {1, 8}, {2, 8} ], 1, 2} | |
| 'yuv1' | 8 bits YUV 411 packed  Y0 Y1 Cb Y2 Y3 Cr | | {'yuv1', 3, [ {1, 8}, {2, 8}, {3, 8} ], 3, 2} | |
| 'v308' | 8 bits YUV 444 packed  Cr Y Cb | {'v308', 3, [ {3, 8}, {1, 8}, {2, 8} ], 0, 2} | | |
| 'v408' | 8 bits YUVA 444 packed  Cb Y Cr A | {'v408', 4, [ {2, 8}, {1, 8}, {3, 8}, {7, 8}], 0, 2} | | |
| 'y210' | 10 bits YUV 422 packed  Y0 Cb Y1 Cr | {'y210', 3, [ {1, 10}, {2, 10}, {3, 10}], 1, 2} | | |
| 'v410' | 10 bits YUV 444 packed, 2 unused bits at LSB byte 0  Cr Y Cb | Not mappable due to endianness packing, unless an adhoc packing type is introduced to pack on 32 bit little endian. | | |
| 'v210' | YUV 422 10bits | Not mappable, unless an adhoc packing type is introduced | | |
| 'rgb3' | RGB 24 bits  ('rgb ' as defined in QT) | {'rgb3', 3, [ {4, 8}, {5, 8}, {6, 8} ], 0, 2} | | |
| 'i420' | YUV 420 8 bits planar  Y Cb Cr full scale (JPEG) | {'i420', 3, [ {1, 8}, {2, 8}, {3, 8} ], 2, 0} | | |
| 'rgba' | Packed RGBA 32bits | {'rgba', 4, [ {4, 8}, {5, 8}, {6, 8}, {7, 8}], 0, 1} | | |
| 'abgr' | Packed RGBA 32bits | {'abgr', 4, [ {7, 8}, {6, 8}, {5, 8}, {4, 8}], 0, 1} | | |

EDITOR’S NOTE1: decide which one we keep, add other configurations, finalize profile names.



## MIME type sub-parameters

When the ‘codecs’ parameter of a MIME type is used, as defined in RFC 6381, its value shall be formatted as follows (each element in the value is separated from the previous one using a dot ‘.’):

* The first element of the value shall be set to ‘uncv’
* If the profile in the UncompressedVideoConfigBox is not 0, an element equal to the profile four-character code string (case sensitive) is added
* Otherwise:
  + an element equal to the sampling\_type expressed in hexadecimal is added
  + an element equal to the packing\_type expressed in hexadecimal is added
  + for each component, an element equal to ‘aLb’ is added, with ‘a’ the hexadecimal value of the component type and ‘b’ the hexadecimal value of the component format.

Hexadecimal value shall not be prefixed with ‘0x’, and leading zeros should be omitted.

Examples:

* For an uncompressed video with profile yvyu: codecs=uncv.yvyu
* For an uncompressed video with monochrome and alpha, same sampling (sampling\_type=0), each stored on 10 bits little endian with byte-aligned packing (packing\_type=2): codecs=uncv.0.2.0LA.8LA
* For an uncompressed video with 2 bits alpha, 10 bits R, B, G, same sampling (sampling\_type =0) stored with packing\_type =3 (packed no byte alignment): codecs=uncv.0.4.8L2.4LA.5LA.6LA

# Video sample entry extensions

### Overview

The boxes defined in this subclause can be used to provide further information on one or more components present in the uncompressed video format.

Each of the defined box can be:

* added to an uncompressed video sample entry with type 'uncv'
* associated with an uncompressed video item with item type 'uncv'

NOTE The syntaxes in this section are given for a video sample entry container and the defined boxes therefore extend Box or FullBox. When used in an ItemPropertyContainerBox, the same syntax applies but the defined boxes extend ItemProperty or ItemFullProperty.

EDITOR’S NOTE: we don’t have anything for alpha or other component types. Some YUV+A format assume that alpha values range in 16-235 for 8bit, other assume full range for alpha. Additionally, not all components will be interpreted as a linear value. Contributions on this topic are welcome.

### Chroma Location

#### Definition

Box Type: 'cloc'   
Container: Video sample entry, ItemPropertyContainerBox  
Mandatory: No  
Quantity: Zero or one per video sample entry or per item

The ChromaLocationBox may be used to describe the chroma subsampling location method used for uncompressed video using a sampling\_type value of 2. It shall not be present for other uncompressed video configurations.

If ChromaLocationBox is not present for an uncompressed video using sampling\_type value of 2, the associated Chroma420SampleLocType is 0.

If ChromaLocationBox is used as an item property, it shall be marked as essential.

EDITOR’S NOTE: we don’t have anything for chroma location for 422 and 410. Do we want any other configurations than co-sited for these ? If so, do we want to extend values here (for 410) or in CICP ?

#### Syntax

aligned(8) class ChromaLocationBox extends Box('cloc') {  
 unsigned int(5) reserved = 0b00000;  
 unsigned int(3) chroma\_location;  
}

#### Semantics

chroma\_location indicates the Chroma420SampleLocType as defined in ISO/IEC 23091-2.

### Interlaced Content Type

#### Definition

Box Type: 'fiel'   
Container: Video sample entry, ItemPropertyContainerBox  
Mandatory: No  
Quantity: Zero or one per video sample entry or per item

The InterlaceContentTypeBox is used to describe field ordering in uncompressed video configurations. When absent, the video is assumed to be progressive.

If InterlaceContentTypeBox is used as an item property, it shall be marked as essential.

#### Syntax

class InterlacedContentTypeBox extends Box('fiel’) {

unsigned int(8) field\_count; //1 or 2   
 unsigned int(8) field\_ordering;

}

#### Semantics

field\_count indicates the number of fields. Value shall be either 1 (progressive video) or 2 (interlaced video).

field\_ordering: when the field count is 2, specifies which field contains the topmost scan-line, which field should be displayed earliest, and which is stored first in each sample. Each sample consists of two distinct images, each coding one field: the field with the topmost scan-line, T, and the other field, B. The following defines the permitted variants:

0: There is only one field.

1: T is displayed earliest, T is stored first in the file.

6: B is displayed earliest, B is stored first in the file.

9: B is displayed earliest, T is stored first in the file.

14: T is displayed earliest, B is stored first in the file.

### Frame Packing Information

#### Definition

Box Type: 'pack'   
Container: Video sample entry, ItemPropertyContainerBox  
Mandatory: No  
Quantity: Zero or one per video sample entry or per item

TheFramePackingInformationBox can be used to describe how two pictures are packed into an uncompressed video frame for stereoscopic imagery.

If not present, each uncompressed video frame consists in a single picture.

If FramePackingInformationBox is used as an item property, it shall be marked as essential.

#### Syntax

class FramePackingInfoBox extends Box('pack') {  
 unsigned int(4) video\_frame\_packing;  
 unsigned int(4) PackedContentInterpretationType  
 unsigned int(1) QuincunxSamplingFlag;  
 unsigned int(7) reserved = 0;  
}

#### Semantics

video\_frame\_packing, QuincunxSamplingFlag, PackedContentInterpretationType as defined in ISO/IEC 23091-2.

### Disparity Information

#### Definition

Box Type: 'disi'   
Container: Video sample entry, ItemPropertyContainerBox  
Mandatory: No  
Quantity: Zero or one per video sample entry or per item

TheDisparityInformationBox can be used to describe how values of a disparity map should be interpreted. If not present, the mapping from disparity value to distance in meters uses the default values (parallax\_zero=2N-1, parallax\_scale=256, dref=300, wref=100) as defined in ISO/IEC 23002-3.

If this box is used as an item property, it shall be marked as essential.

EDITOR’S NOTE: there is no disparity-related aspects in CICP, maybe merging things from 23002-3 in CICP video should be considered

#### Syntax

class DisparityInformationBox extends FullBox('disi', 0, 0) {  
 unsigned int(16) parallax\_zero;  
 unsigned int(16) parallax\_scale;  
 unsigned int(16) dref;  
 unsigned int(16) wref;  
}

#### Semantics

parallax\_zero, parallax\_zero, parallax\_zero, parallax\_zero: as defined in ISO/IEC 23002-3.

### Depth Mapping Information

#### Definition

Box Type: 'depi'   
Container: Video sample entry, ItemPropertyContainerBox  
Mandatory: No  
Quantity: Zero or one per video sample entry or per item

The DepthMappingInformationBox may be used to describe how values in a depth map are transformed into distance values. If not present, the mapping from depth value to distance values follow the default values (nknear=128, nkfar=128) as defined in ISO/IEC 23002-3.

If this box is used as an item property, it shall be marked as essential.

EDITOR’S NOTE: there is no depth-related aspects in CICP, maybe merging things from 23002-3 in CICP video should be considered

#### Syntax

class DepthInfoBox extends FullBox('depi', 0, 0) {  
 unsigned int(8) nknear;  
 unsigned int(8) nkfar;  
}

#### Semantics

nknear, nkfar near and far distances as defined in ISO/IEC 23002-3.

# Multiple track storage

## Overview

Multiple track storage allows describing uncompressed video streams whose components are in multiple tracks. This process may be used in various cases:

* simplify editing, by modifying only one component (e.g. alpha or depth) without modifying the other components (colour)
* provide players a way to play a subpart of the components (e.g. colour) instead of failing playback due to unsupported configuration
* enable efficient disk and network usage for partial access (spatial, temporal) of a subset of the components. Note that this might not always be possible depending on component packing types.
* Use components with different spatial resolutions

## Uncompressed video track group

Uncompressed video tracks containing components related to the same media source may be grouped using a TrackGroupBox with the track\_group\_type value of 'unvg'.

The pair of track\_group\_id and track\_group\_type identifies a track group within the file. The tracks that contain a particular TrackGroupTypeBox having the same value of track\_group\_id and track\_group\_type belong to the same track group.

Tracks belonging to such a group may have different resolutions, but shall have:

* The same aspect ratio
* The same temporal layout, i.e. there shall not be any time for which no sample is defined for one or more tracks of the group while a sample is defined for other tracks of the group.

NOTE Derived specification may further restrict the kind of components allowed in 'unvg' track groups, the timescale used, the visual dimension of the tracks, etc…

All tracks belonging to a track group with track\_group\_type value of 'unvg' shall be uncompressed video tracks as defined in Clause 5. The display or processing of (part of) a track group with track\_group\_type value of 'unvg' is application specific.

EDITOR’S NOTE1: we don’t have any proposal for multiple item storage, maybe Entity Groups could be used if feature is needed. Contributions on the topic are welcome.

EDITOR’S NOTE2: the proposed group could be used for a mix of compressed and uncompressed video (e.g. color compressed, alpha / depth uncompressed) by relaxing the constraints that all members of the track group shall be ‘uncv’ tracks.

## Image tiling

Image data may be spatially split in different rectangular zones, or tiles, each tile being stored in its own track or image item rather than storing the entire image data as a single track or item.

Video tracks containing tiles of the same media source may be grouped using a TrackGroupBox with the track\_group\_type value of 'unvt'.

The pair of track\_group\_id and track\_group\_type identifies a track group within the file. The tracks that contain a particular TrackGroupTypeBox having the same value of track\_group\_id and track\_group\_type belong to the same track group, i.e. they contain different tiles of the same source image. Tile layouts must be indicated using the width, height and matrix of the track header. Tracks belonging to the same track group with type 'unvt' shall not spatially overlap each other.

For image items, tile layouts can be indicated using the grid derived image item.