International Standard

ISO/IEC 23001-17

First edition
2024-02

Information technology — MPEG systems technologies —

Part 17: Carriage of uncompressed video and images in ISO base media file format

Technologies de l'information — Technologies des systèmes MPEG —

Partie 17: Transport de vidéos et images non compressées dans le format ISO de base pour les fichiers médias

Reference number
ISO/IEC 23001-17:2024(en)

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

Published in Switzerland

Contents

Page

Foreword

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

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

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

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

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, 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 and IEC websites.

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

Information technology — MPEG systems technologies —

Part 17: Carriage of uncompressed video and images in ISO base media file format

1 Scope

This document specifies how uncompressed 2D image and video data is carried in files in the family of standards based on the ISO base media file format (ISO/IEC 14496-12). This includes but is not limited to monochromatic data, colour data, transparency (alpha) information and depth information.

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

2 Normative references

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

ISO/IEC 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*

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

3 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 terminology databases for use in standardization at the following addresses:

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

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

3.1
block
consecutive bytes within the sample data containing one or more component values for one or more pixels and possible padding

3.2
component
part of the image data representing a single channel (or dimension) of the image

Note 1 to entry: In this document, a component may describe visual information such as luminance or chroma, or other information usually not intended for direct display such as depth or transparency.

3.3
frame
two-dimensional rectangular array of pixels contained in the sample data

3.4
interleaving
ordering of pixel or component within the sample data

3.5
pixel
smallest element of an image, comprised of one or more components

3.6
row
horizonal line of pixels within a frame or a tile

3.7
sample data
payload of the media sample when the uncompressed frame is described by a media track, or payload of the item when the uncompressed frame is described by an image item

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

Note 2 to entry: Image item as defined in ISO/IEC 23008-12.

3.8
tile
two-dimensional rectangular array of pixels within a frame

3.9
uncompressed frame
frame for which each value of each component is coded independently from any other component value in the same frame or any other frame

Note 1 to entry: In this document, the uncompressed term is used with some video formats applying sub-sampling of some components for the purpose of data reduction; however, data access to each individual component for such formats is still independent from other components or frames.

3.10
uncompressed image
single uncompressed frame stored as an image item

3.11
uncompressed video
sequence of one or more uncompressed frames

4 Uncompressed video and image formats

4.1 Overview

Uncompressed frames may be stored in ISO base media files as media samples of media tracks or as image items using a generic uncompressed video description defined in this document.

Media tracks, media samples and image items may be further described using the various tools defined in ISO/IEC 14496-12, such as sample group descriptions, `MetaBox`, metadata tracks and sample auxiliary information. User-defined components may be used to carry per-pixel specific information, either in the same sample or item as the described pixels or in a separate track or item.

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, clean aperture, content light level, mirror and rotate properties or track header matrix, etc.

An uncompressed video media sample or item consists of one uncompressed frame. Each uncompressed frame is organized as a set of one or more rectangular, non-overlapping and contiguous (without holes) areas called tiles.

ISOBMF allows constructing files referring to external data. This enables building ISOBMF files describing existing uncompressed image or video files without having to copy the media data, simplifying integration into existing workflows.

4.2 Storage in media tracks

Uncompressed video tracks compliant to this document are tracks compliant to ISO/IEC 14496-12 that use a `VisualSampleEntry` with `codingname` equal to `'uncv'`, hereafter called uncompressed video sample entry.

The uncompressed video sample entry shall contain

— one `UncompressedFrameConfigBox;`

— one `ComponentDefinitionBox` if the `UncompressedFrameConfigBox` version is not 1.

When both `UncompressedFrameConfigBox` and `ComponentDefinitionBox` are present in the sample entry, the `ComponentDefinitionBox` shall precede the `UncompressedFrameConfigBox`.

The `compressorname` field of an uncompressed video sample entry should be set to all 0 (empty string). The `depth` field of an uncompressed video sample entry shall be ignored and should be set to 0, the bit depth per component being indicated by the `UncompressedFrameConfigBox`.

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 documents the exact frame dimension, in pixels of any non-subsampled component, of any sample of the video stream that is described by this sample entry. Consequently, if the frame dimension changes within a video track, multiple sample entries shall be used.

The payload of an uncompressed video media sample consists of one uncompressed frame.

The size in bytes of an uncompressed video media sample shall be at least the size in bytes required to store all the components values documented by the uncompressed video configuration as defined in subclause 5.2.

NOTE The sample data can be larger than the size in bytes required to store all the components values, typically to store information in the trailing data. How such additional bytes are handled by a file reader is out of scope of this document.

Each uncompressed video media sample shall be marked as a sync sample. As a consequence, the `SyncSampleBox`, `ShadowSyncSampleBox`, `CompositionOffsetBox` and `CompositionToDecodeBox` shall not be present in the track.

Media tracks containing only non-visual components should be marked as not present in the presentation, i.e. `track_in_movie` flag should not be set.

Media tracks containing user-defined components providing per-pixel information for pixels in another track should use a track reference of type `'cdsc'` to the track they describe.

4.3 Storage in image items

An uncompressed image compliant to this document is an image item compliant to ISO/IEC 23008-12 with the `item_type 'unci'`.

An uncompressed image shall be associated with:

— an `UncompressedFrameConfigBox` essential item property, i.e., `essential` shall be equal to 1 for an `UncompressedFrameConfigBox` item property associated with an image item of type `'unci'`;

— a `ComponentDefinitionBox` essential item property if the `UncompressedFrameConfigBox` version is not 1;

— an `ImageSpatialExtentsProperty` whose `image_width` and `image_height` fields shall document the exact frame dimension, in pixels, of the reconstructed image, i.e. the size of the image before applying any associated transformative properties.

The payload of an uncompressed image consists of one uncompressed frame.

The size in bytes of an uncompressed image item shall be at least the size in bytes required to store all the components values documented by the uncompressed video configuration as defined in subclause 5.2.

NOTE The image item data can be larger than the size in bytes required to store all the components values, typically to store information in the trailing data. How such additional bytes are handled by a file reader is out of scope of this document.

An uncompressed image can be further documented using the various tools defined in ISO/IEC 14496-12 or ISO/IEC 23008-12, such as descriptive item properties.

Uncompressed images containing only non-visual components should be marked as hidden items, i.e. have (`flags` & 1) equal to 1 in their `ItemInfoEntry`.

Uncompressed images containing user-defined components providing per-pixel information for pixels in another item should use an item reference of type `'cdsc'` to the item they describe.

If an uncompressed item is associated with an `AuxiliaryTypeProperty` indicating alpha (resp. depth), the uncompressed item shall have one and only one component of type alpha (resp. depth).

5 Uncompressed frame description

5.1 Component Definition

5.1.1 Definition

Box Type: `'cmpd'`
Container: Video sample entry, `ItemPropertyContainerBox`
Mandatory: see below
Quantity: At most one per uncompressed video sample entry or at most one associated per uncompressed image item

The `ComponentDefinitionBox` is used to document the types of components present in samples or items associated with this box through the sample entry or through item property association.

Components defined in the `ComponentDefinitionBox` are referenced by indexes in various boxes in this document. Care has to be taken while removing components from an uncompressed video or image to also remove in other boxes any reference to the removed components. There is no requirement that all components defined in the `ComponentDefinitionBox` are referenced by other boxes.

For all boxes referring to components defined in the `ComponentDefinitionBox`, the associated `ComponentDefinitionBox` (resp. the associated `UncompressedFrameConfigBox`) is defined as:

— for an uncompressed video, the `ComponentDefinitionBox` (resp. `UncompressedFrameConfigBox`) present in the same sample entry as the referring box;

— for an uncompressed image, the `ComponentDefinitionBox` (resp. `UncompressedFrameConfigBox`) associated, through `ItemPropertyAssociationBox`, to the same image item as the referring box.

This box shall be present if the version of the associated `UncompressedFrameConfigBox` is 0.

This box may be present if the version of the associated `UncompressedFrameConfigBox` is 1.

When the `ComponentDefinitionBox` is not present, the associated `ComponentDefinitionBox` is implicitly defined as indicated by the profile of the associated `UncompressedFrameConfigBox`, see subclause 5.3.

The component_index field value defined in boxes using an associated ComponentDefinitionBox indicates the index in the list of components, with value 0 indicating the first component listed in the associated ComponentDefinitionBox. The component_index field value shall be strictly less than the component_count field value of the associated ComponentDefinitionBox.

A ComponentDefinitionBox may describe two or more components with the same type (e.g. two monochrome components representing different portions of the electromagnetic spectrum), 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 document.

Table 1 — Component types

Value	Description
0	Monochrome component
1	Luma component (Y)
2	Chroma component (Cb / U)
3	Chroma component (Cr / V)
4	Red component (R)
5	Green component (G)
6	Blue component (B)
7	Alpha/transparency component (A)
8	Depth component (D)
9	Disparity component (Disp)
10	Palette component (P) The component_format value for this component shall be 0.
11	Filter Array (FA) component such as Bayer, RGBW, etc.
12	Padded component (unused bits/bytes)
13	Cyan component (C)
14	Magenta component (M)
15	Yellow component (Y)
16	Key (black) component (K)
17 to 0x7FFF	ISO/IEC reserved
0x8000 to 0xFFFF	User-defined component type(s)

Component types reflect only a nominal characterization of the pixel data and how that pixel data should be displayed; precise bandpass limits, for example, are not implied and may differ from image to image. For some component types, some other boxes can provide additional information, such as ColourInformationBox or MasteringDisplayColourVolumeBox. For other component types, the signalling of such information is out of scope of this document.

if a ColourInformationBox is present for the media sample or item:

— if the ColourInformationBox describes components by numbers, there shall be at least as many components in the reconstructed image as indicated by the ColourInformationBox, and components are matched by indices;

— if the ColourInformationBox describes components by labels (e.g. Y, Cb / U, Cr / V or R, G, B), these components shall be present in the image after applying palette or filter array, if any.

If the bands need to be identified beyond the component type, it is recommended to use URIs in place of the enumerated component_type value.

5.1.2 Syntax

```
aligned(8) class ComponentDefinitionBox extends Box('cmpd') {
   unsigned int(32) component_count;
   {
      unsigned int(16) component_type;
      if (component_type>=0x8000) {
         utf8string component_type_uri;
      }
   } [component_count];
}
```

5.1.3 Semantics

`component_count` indicates the number of components described in this box.

`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.

5.2 Uncompressed Frame Configuration

5.2.1 Definition

5.2.1.1 Overview

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 `'unci'`
Quantity: One per uncompressed video sample entry or one associated per uncompressed image item

The `UncompressedFrameConfigBox` describes uncompressed frames that are composed of one or more components. RGB or YUV formats are typical examples of such uncompressed frames for which each colour component is a component of the uncompressed frame. Other component types can also be described (e.g. disparity, transparency, infra-red). The type of each component is specified by the `component_type` field in the associated `ComponentDefinitionBox`.

Component values may be absolute values or indexes into a colour palette, with adjustable black and white reference levels. Pattern-based sensor data, such as Bayer, can be described through user-defined patterns, together with various sensor information (polarization, non-uniformity correction, broken pixels, etc.).

For each component, this box specifies the numerical format (e.g. unsigned integer, IEEE 754 binary32 floating point) and bit depth through the `component_format` and `component_bit_depth_minus_one` fields.

Pixel data may be interleaved per component, per pixel, per row or per tile, as specified by the `interleave_type` field, with byte alignment for each row and tile specified by the fields `row_align_size` and `tile_align_size`. Pixel data may also be grouped together in blocks, typically to respect endianness constraints, as specified by the `block_size`, `block_pad_lsb` and `block_little_endian` fields.

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

The `UncompressedFrameConfigBox` may indicate a profile, allowing faster identification of the class of video data used. Profiles are defined in subclause 5.3. Some profiles may use version 1 of the `UncompressedFrameConfigBox`, in which case only the profile is given and the `ComponentDefinitionBox`, if absent, is implicitly defined by the profile.

5.2.1.2 Component ordering and constraints

Each pixel in a frame is made of one or more components, where each component is assigned a type (e.g. 'Y', 'U', 'V'), as detailed in subclause 5.1.

The order in which components are specified in the `UncompressedFrameConfigBox` indicates the order in which components are placed into the sample data or within blocks, prior to blocking and endian conversion.

Some formats, such as Bayer image data, contain a 2D array of single-component values, with each individual component value assigned to a component type using a fixed pattern. Such formats are described as mono-component data with no subsampling, use a component type of 11 ('FA'). There shall be at most one component with type 11 ('FA') present in the component list. There may be additional components of other types present, for example to associate an alpha component with a Bayer image. If a component with type 11 ('FA') is present, there shall be a `ComponentPatternDefinitionBox` present in the video sample entry or associated to the item to indicate the pattern of component values.

Some formats code colours according to a set of predefined colours, or palette. Such formats are described as single-component with no subsampling, use a component type of 10 ('P'). There shall be at most one component with type 10 ('P') present in the component list. There may be additional components of other types present, for example to associate per-pixel alpha component with a palette image. If a component with type 10 ('P') is present, there shall be a `ComponentPaletteBox` present in the video sample entry or associated to the item to indicate the palette values.

5.2.1.3 Component size and numerical format

The variable `component_bit_depth` for a component is defined as (`component_bit_depth_minus_one` + 1).

For a given component, the binary representation of each value is given by `component_bit_depth` (the size in bits of each component value) and `component_format`.

The possible values for `component_format` field is defined in Table 2.

Table 2 — Component formats

Value	Description
0	Component value is an unsigned integer coded on `component_bit_depth` bits.
1	Component value is an IEEE 754 binary float number coded on `component_bit_depth` bits (e.g. if `component_bit_depth` is 16, then the component value is coded as IEEE 754 'binary16'). For this component format, `component_bit_depth` values shall be 16, 32, 64, 128 or 256; other values are forbidden.
2	Component value is a complex number coded on `component_bit_depth` bits, where the first `component_bit_depth`/2 bits represent the real part and the next `component_bit_depth`/2 bits represent the imaginary part. Each part is coded as an IEEE 754 binary float of the size `component_bit_depth`/2. For this component format, `component_bit_depth` values shall be 32, 64, 128 or 256; other values are forbidden.
3 to 255	ISO/IEC reserved for future definition.

If `component_align_size` is 0, the component value is coded on `component_bit_depth` bits exactly.

Otherwise (`component_align_size` is not 0), the component value is coded as a word WC of `component_align_size` bytes, starting on a byte boundary. This implies that some pre-alignment padding bits may be present after the previous component value stored; if such pre-alignment padding bits are present, they shall be set to 0. The least significant bit of the component value is located at the least significant bit of WC. Alignment padding bits, if present, are located at the most significant bits and shall be set to 0. If `components_little_endian` is 0, `WC` is stored as a big-endian word. Otherwise (`components_little_endian` is 1), WC is stored as a little-endian word.

NOTE For example, a pixel with 10-bit unaligned (`component_align_size`=0) R, G and B components, followed by a 1-byte aligned 7-bit A component, stored using pixel interleaving mode, would exist in the sample data as 30 consecutive bits containing R, G and B, followed by 2 pre-alignment padding bits for byte alignment, followed by one alignment padding bit then followed by the 7-bit A value (bringing the A value aligned on 1 byte), for a total of 5 bytes.

For each component, `component_align_size` shall be either 0 or such that `component_align_size*8` is greater than `component_bit_depth`. If `components_little_endian` is 1, `block_little_endian` shall be 0 and `component_align_size` of each component shall be different from 0.

Storage of aligned component values is illustrated in Figure 27.

5.2.1.4 Tiling

The uncompressed frame data is structured in a 2D grid of tiles, where the number of tiles in the horizontal direction (resp. vertical direction) is specified by the variable `num_tile_cols_minus_one`+1 (resp. `num_tile_rows_minus_one`+1). Tiles allow grouping together the component values of pixels close to each-other (i.e., in the same spatial region of the frame).

All tiles have the same width and height. The frame width (resp. height) shall be a multiple of `num_tile_cols_minus_one`+1 (resp. `num_tile_rows_minus_one`+1). The tile width is `w/(num_tile_cols_minus_one+1)`, with `w` the frame width, and the tile height is `h/(num_tile_rows_minus_one+1)`, with `h` the frame height.

If the width (resp. height) in a source image is not an integer multiple of `num_tile_cols_minus_one`+1 (resp. `num_tile_rows_minus_one`+1), an application creating the tiled frame shall pad the source image with an appropriate number of columns to the right (resp. rows to the bottom), and the original image dimension shall be documented using a `CleanApertureBox` for media tracks or a clean aperture transformative item property for image items. The width and height fields of the sample entry or the `image_width` and `image_height` fields of the `ImageSpatialExtentsProperty` shall specify the padded width and height.

Tiles shall be stored in raster-scan order: the top-left tile is stored first, followed by the tile to its right, and the first tile of a tile row is stored after the last tile of the previous tile row.

Within a tile, component values shall be stored in raster-scan order, left-to-right and top to bottom: for a given component, the value for pixel {x+1,y} is stored after (but possibly not contiguous with) the value for pixel {x,y}, and the value for pixel {x,y+1} is stored after (but possibly not contiguous with) the value for pixel {x, y}.

5.2.1.5 Sampling mode

5.2.1.5.1 Definition

All components in a frame either have the same dimensions or use pre-defined sampling modes, indicated by the `sampling_type` field. Possible values for this field are described in Table 3.

NOTE 1 A sampling mode can restrict the possible interleaving modes since the number of values is not the same for each component.

NOTE 2 Derived specifications can further restrict the usage of sampling modes with tiling.

Table 3 — Sampling type values

Value	Sampling mode
0	No subsampling
1	YCbCr 4:2:2 subsampling
2	YCbCr 4:2:0 subsampling
3	YCbCr 4:1:1 subsampling
4 to 0xFF	Reserved

5.2.1.5.2 No subsampling

In this mode, all components have the same width and height as the frame.

The tile width and height are not restricted. Derived specifications can further restrict this, for example to enforce width and height to be multiples of 2 in case Y, U and V components are present.

5.2.1.5.3 YCbCr 4:2:2 Subsampling

This sampling mode shall only be used if the following conditions are all true:

— all three `component_type` values 'Y', 'U' and 'V' are present in the component list;

— the tile width is a multiple of 2;

— the `interleave_type` field is set to 0 (component interleaving mode), 2 (mixed interleaving mode) or 5 (multi-Y pixel interleaving mode).

The width and height of the 'Y' component are the width and height of the frame. The height of the 'U' and 'V' component is the same as the height of the 'Y' component. The width of the 'U' and 'V' component is half the width of the 'Y' component. Components of other types may be present, and have the same dimension as the 'Y' component.

Pixels {x,y} and {x+1,y}, with x%2==0, share the same component values 'U' and 'V'.

If `row_align_size` is not 0 and `interleave_type` is 0:

— `row_align_size` shall be a multiple of 2;

— the row alignment for components of type 'U' and 'V', as defined in 5.2.1.7, shall be done using `row_align_size/2`.

If `tile_align_size` is not 0:

— `tile_align_size` shall be a multiple of 2;

— the tile alignment for components of type 'U' and 'V', as defined in 5.2.1.7, shall be done using `tile_align_size/2`.

5.2.1.5.4 YCbCr 4:2:0 Subsampling

This sampling mode shall only be used if the following conditions are all true:

— all three `component_type` values 'Y', 'U' and 'V' are present in the component list;

— both tile width and height are multiple of 2;

— the `interleave_type` field is set to 0 (component interleaving mode) or 2 (mixed interleaving mode).

The width and height of the 'Y' component are the width and height of the frame. The width of the 'U' and 'V' component is half the width of the 'Y' component'. The height of the 'U' and 'V' component is half the height of the 'Y' component. Components of other types may be present, and have the same dimension as the 'Y' component.

Pixels {x,y}, {x+1,y}, {x,y+1} and {x+1,y+1} with x%2==0 and y%2==0, share the same component values 'U' and 'V'.

If `row_align_size` is not 0 and `interleave_type` is 0:

— `row_align_size` shall be a multiple of 2;

— the row alignment for components of type 'U' and 'V', as defined in 5.2.1.7, shall be done using `row_align_size/2`.

If `tile_align_size` is not 0:

— `tile_align_size` shall be a multiple of 4;

— the tile alignment for components of type 'U' and 'V', as defined in 5.2.1.7, shall be done using `tile_align_size/4`.

5.2.1.5.5 YCbCr 4:1:1 Subsampling

This sampling mode shall only be used if the following conditions are all true:

— all three `component_type` values 'Y', 'U' and 'V' are present in the component list;

— tile width is a multiple of 4;

— the `interleave_type` field is set to 0 (component interleaving mode), 2 (mixed interleaving mode) or 5 (Multi-Y pixel interleaving mode).

The width and height of the 'Y' component are the width and height of the frame. The height of the 'U' and 'V' component is the same as the height of the 'Y' component. The width of the 'U' and 'V' component is the width of the 'Y' component divided by 4. Components of other types may be present, and have the same dimension as the 'Y' component.

Pixels {x,y} , {x+1,y} , {x+2,y} , {x+3,y}, with x%4==0, share the same component values 'U' and 'V'.

If `row_align_size` is not 0 and `interleave_type` is 0:

— `row_align_size` shall be a multiple of 4;

— the row alignment for components of type 'U' and 'V', as defined in 5.2.1.7, shall be done using `row_align_size/4`.

If `tile_align_size` is not 0:

— `tile_align_size` shall be a multiple of 4;

— the tile alignment for components of type 'U' and 'V', as defined in 5.2.1.7, shall be done using `tile_align_size/4`.

5.2.1.6 Interleaving modes

5.2.1.6.1 Definition

The interleaving mode of pixels within a tile (or within the frame if a single tile is used) is indicated by `interleave_type`, as defined in Table 4. The `interleave_type` shall apply to all listed components unless stated otherwise in the interleaving mode definition.

NOTE The interleaving describes the layout of values within the sample data. Positioning of sample data within the container file is done according to ISO/IEC 14496-12.

Table 4 — Interleaving type values

Value	Interleaving mode
0	Component interleaving
1	Pixel interleaving
2	Mixed interleaving
3	Row interleaving
4	Tile-component interleaving
5	Multi-Y pixel interleaving
6 to 0xFF	Reserved

5.2.1.6.2 Component interleaving

For a given component, values for all pixels of a tile shall be located sequentially in the sample data. Component values shall be located in the order the components were declared.