ISO/IEC JTC 1/SC 29/WG 03 N0400

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Working draft of amendment to ISO/IEC 14496-12:2021 CDAM1 Improved brand documentation and other improvements

**Status:** Approved

**Date of document:** 2021-11-02

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 26 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

*In 3.1.35 change the name of the term defined from*

**sample description**

structure which defines and describes the format of some number of samples in a track

*to*

**sample entry**

structure which defines and describes the format of some number of samples in a track

*In 3.1 add the following new definitions*

### **external elementary stream**

elementary stream containing access units with external pictures

### **external picture**

picture that is in the external elementary stream in an EST and is needed for inter prediction reference in decoding of the elementary stream in the MST when random accessing from certain EDRAP pictures in the MST

### **external stream track (EST)**

track containing an external elementary stream

### **main stream track (MST)**

track containing a video elementary stream

### **media component**

TBD (related to preselection)

### **preselection**

a set of one or multiple media components representing one version of the media presentation that may be selected by a user for simultaneous decoding/presentation

### **random access**

decoding of an elementary stream starting from a particular picture without decoding of any picture in the elementary stream earlier in decoding order

*In 3.2 add the following abbreviated terms*

EDRAP extended dependent random access point

EST external stream track

MST main stream track

*In section 4.3.1, File-type box, replace*

This box shall be placed as early as possible in the file (e.g. after any obligatory signature, but before any significant variable-size boxes such as a MovieBox, MediaDataBox, or FreeSpaceBox). It identifies which specification is the ‘best use’ of the file, and a minor version of that specification; and also a set of other specifications to which the file complies. Readers implementing this format should attempt to read files that are marked as compatible with any of the specifications that the reader implements. Any incompatible change in a specification should therefore register a new ‘brand’ identifier to identify files conformant to the new specification.

*with*

This box shall be placed as early as possible in the file (e.g. after any obligatory signature, but before any significant variable-size boxes such as a MovieBox, MediaDataBox, or FreeSpaceBox). It identifies which specification is the ‘best use’ of the file (the major\_brand), and a minor version of that specification; and also a set of other specifications to which the file complies (the compatible\_brands); the major\_brand should be repeated in the compatible\_brands list. Readers implementing this format should attempt to read files that are marked as compatible with any of the specifications that the reader implements. Any incompatible change in a specification should therefore register a new ‘brand’ identifier to identify files conformant to the new specification.

*In 6.3.4 change the row in Table 1 for stsd to read*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | stsd | \* | 8.5.2 | *sample description (codec types, initialization etc.)* |

*In 8.3.2 adjust the indentation of the paragraphs under width and height, and change to read:*

width and height fixed-point 16.16 values are track-dependent as follows:

For text and subtitle tracks, they may, depending on the coding format, describe the suggested size of the rendering area. For such tracks, the value 0x0 may also be used to indicate that the data may be rendered at any size, that no preferred size has been indicated and that the actual size may be determined by the external context or by reusing the width and height of another track. For those tracks, the flag track\_size\_is\_aspect\_ratio may also be used.

For non-visual tracks (e.g. audio), they should be set to zero.

For all other tracks, they specify the track's visual presentation size. These need not be the same as the pixel dimensions of the images, which is documented in the sample entries; all images in the sequence are scaled to this size, before any overall transformation of the track represented by the matrix. The pixel dimensions of the images are the default values.

*Add new subclause 8.3.3.4*

**8.3.3.4 Associated external stream track reference**

A track reference of type 'aest' (meaning "associated external stream track") may be included in an MST, referencing an EST.

When an MST has a track reference of type 'aest', the following applies:

* The MST should have at least one sample that contains an EDRAP picture.
* For each sample sampleA in the MST containing an EDRAP picture, there shall be one and only one sample sampleB in the associated EST that has the same decoding time as sampleA, and a number of consecutive samples in the associated EST, starting from sampleB, shall exclusively contain all the external pictures that are needed when random accessing from the EDRAP picture contained in sampleA.

Every sample in the EST shall be identified as a sync sample. The EST track header flags shall have track\_in\_movie and track\_in\_preview both set to 0.

[Ed.Note: What sample entry type should be used for ESTs, taking into account that a sample in an EST may include more than one coded picture?]

*In 8.3.4.3 add the following new value*

track\_group\_type indicates the grouping\_type and shall be set to one of the following values, or a value registered, or a value from a derived specification or registration:

'msrc' indicates that this track belongs to a multi-source presentation. Specified in .

'ster' indicates that this track is either the left or right view of a stereo pair suitable for playback on a stereoscopic display. Specified in 8.3.4.1.3.

'pres' indicates that this track contributes to a preselection. Specified in 8.3.4.4.3.

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.*Add new 8.3.4.4.3*

8.3.4.4.3 Preselection box

8.3.4.4.3.1 Definition

TrackGroupTypeBox with track\_group\_type equal to 'pres' indicates that this track contributes to a preselection.

The tracks that have the same value of track\_group\_id within PreselectionGroupBox are part of the same preselection.

Preselections can be qualified by language, kind or media specific attributes like audio rendering indications or channel layouts. Attributes signaled in a preselection box shall take precedence over attributes signaled in contributing tracks.

All attributes uniquely qualifying a preselection shall be present in at least one Preselection Box of the preselection. If present in more than one Preselection Box of the preselection, the boxes shall be identical.

NOTE: Preselections group tracks of the same media type only.

Tracks not containing all required media components for at least one preselection shall have the track\_in\_movie flag set to ‘0’ in their Track Header Boxes. This prevents players not understanding the Preselection Box from playing the track resulting in an incomplete experience.

NOTE: It is good practice to have one track with track\_in\_movie flag set to one. This implies that this track provides at least one complete experience.

8.3.4.4.3.2 Syntax

aligned(8) class PreselectionGroupBox extends TrackGroupTypeBox('pres')   
{  
 if (flags & 1) {  
 unsigned int(8) selectionPriority=1  
 }

PreselectionInformationBox()  
 PreselectionProcessingBox()  
}

8.3.4.4.3.3 Semantics

selection\_priority is an integer that declares the priority of the preselection in cases where no other differentiation such as through the media language is possible. A lower number indicates a higher priority.

8.3.4.4.4 Preselection information box

8.3.4.4.4.1 Definition

Box Type: 'prsi'  
Container: Preselection Box   
Mandatory: Yes  
Quantity: Exactly One

This Box aggregates all semantic information about the preselection.

8.3.4.4.4.2 Syntax

aligned(8) class PreselectionInformationBox   
 extends FullBox('prsi', version=0, 0 ){  
 // Boxes describing the preselection  
}

8.3.4.4.4.3 Semantics

TBD

8.3.4.4.5 Preselection processing box

8.3.4.4.5.1 Definition

Box Type: 'prsp'  
Container: Preselection Box   
Mandatory: Yes  
Quantity: Exactly One

This box contains information how the tracks contributing to the preselection shall be processed. Media type specific boxes may be used to describe further processing.

8.3.4.4.5.2 Syntax

aligned(8) class PreselectionProcessingBox   
 extends FullBox('prsp', version=0, flags ){  
 string preselection\_tag;   
 if (flags & 1) {  
 unsigned int(8) order=0  
 }

// further attributes and Boxes defining additional processing of  
 // the track contributing to the preselection  
}

8.3.4.4.5.3 Semantics

preselection\_tag is an integer that contains an identifier for the label. Labels with the same value belong to a label group. The default value of zero indicates that the label does not belong to any label group.

order specifies the conformance rules for Representations in Adaptation Sets within the Preselection according to ISO/IEC 23009-1 [ref], from the following enumerated set:

0: undefined

1: time-ordered

2: fully-ordered

*Change 8.4.6.1:*

Box Type: 'elng'  
Container: Media Box ('mdia'), Preselection Information Box('prsi')  
Mandatory: No  
Quantity: Zero or one

*In 8.5.2.1 change*

If the ‘format’ field of a SampleEntry is unrecognized, neither the sample description itself, nor the associated media samples, shall be decoded.

*to*

If the ‘format’ field of a SampleEntry is unrecognized, neither the sample entry itself, nor the associated media samples, shall be decoded.

*In 8.5.2.3 change*

data\_reference\_index is an integer that contains the index of the DataEntry to use to retrieve data associated with samples that use this sample description. Data entries are stored in DataReferenceBoxes. The index ranges from 1 to the number of data entries.

*to*

data\_reference\_index is an integer that contains the index of the DataEntry to use to retrieve data associated with samples that use this sample entry. Data entries are stored in DataReferenceBoxes. The index ranges from 1 to the number of data entries.

*In 8.7.2.1 change*

The data reference object contains a table of data references (normally URLs) that declare the location(s) of the media data used within the presentation. The data reference index in the sample description ties entries in this table to the samples in the track. A track may be split over several sources in this way.

*to*

The data reference object contains a table of data references (normally URLs) that declare the location(s) of the media data used within the presentation. The data reference index in the sample entry ties entries in this table to the samples in the track. A track may be split over several sources in this way.

*In 8.7.4.1 change*

Samples within the media data are grouped into chunks. Chunks can be of different sizes, and the samples within a chunk can have different sizes. This table can be used to find the chunk that contains a sample, its position, and the associated sample description.

*to*

Samples within the media data are grouped into chunks. Chunks can be of different sizes, and the samples within a chunk can have different sizes. This table can be used to find the chunk that contains a sample, its position, and the associated sample entry.

*In 8.8.4.1 change*

The movie fragments extend the presentation in time. They provide the information that would previously have been in the MovieBox. The actual samples are in MediaDataBoxes, as usual, if they are in the same file. The data reference index is in the sample description, so it is possible to build incremental presentations where the media data is in files other than the file containing the MovieBox.

*to*

The movie fragments extend the presentation in time. They provide information that, when fragments are not used, would be in the MovieBox. The actual samples are in MediaDataBoxes, as usual, if they are in the same file. The data reference index is in the sample entry, so it is possible to build incremental presentations where the media data is in files other than the file containing the MovieBox.

*In 8.8.12.1 add to the end of the paragraph*

If the time expressed in the TrackFragmentBaseMediaDecodeTimeBox exceeds the sum of the sample durations of the samples in the preceding movie and movie fragments, then the duration of the last sample preceding this track fragment is extended such that the sum now equals the time given in this box. In this way, it is possible to generate a fragment containing a sample when the time of the next sample is not yet known, by assigning it a small or even zero sample duration, that is then overriden by the time expressed in this box in the following fragment.

*this sentence*

The time expressed in the TrackFragmentBaseMediaDecodeTimeBox shall not be less than the sum of the sample durations of the samples in the preceding movie and movie fragments.

*In 8.9.3.1 add at the end of the section*

If version equals 3, the sample group description describes essential information for the associated samples, and file processors shall not attempt to decode any track for which unrecognized sample group descriptions marked as essential are present.

*In 8.10.1.1 change*

Box Type: 'udta'  
Container: MovieBox, TrackBox, MovieFragmentBox, ~~or~~ TrackFragmentBox or PreselectionInformationBox  
Mandatory: No  
Quantity: Zero or one

*In 8.10.4.1 change*

Box Type: 'kind'  
Container: Container User data box ('udta') in a track or a Preselection Information Box('prsi')  
Mandatory: No  
Quantity: Zero or one

*In 8.11.1.1 change*

The MetaBox is required to contain a HandlerBox indicating the structure or format of the MetaBox contents.

All other contained boxes are specific to the format specified by the HandlerBox.

*to*

When the MetaBox does not contain a PrimaryItemBox, it is required to contain a HandlerBox indicating the structure or format of the MetaBox contents. When the MetaBox does contain a PrimaryItemBox, if that item has a HandlerProperty and there is a HandlerBox present, they shall identify the same handler type. Otherwise, when the primary data is identified by a primary item without a HandlerProperty, and that primary item has an item information entry with an item\_type, the handler type may be the same as the item\_type.

When a HandlerBox is present, all other contained boxes are specific to the format specified by that HandlerBox.

*Change 8.11.1.2 to contain:*

aligned(8) class MetaBox (handler\_type)  
 extends FullBox('meta', version = 0, 0) {  
 HandlerBox(handler\_type) 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  
 ItemDataBox item\_data; // optional  
 Box other\_boxes[]; // optional  
}

*Delete clause 8.11.1.3*

*In 8.11.5.1 add after*

The ItemProtectionBox provides an array of item protection information, for use by the ItemInfoBox.

*the following*

The ProtectionSchemeInfoBoxes shall not contain an OriginalFormatBox when present in an ItemProtectionBox.

*In 8.11.14.1 change*

For a given handler, the primary data may be one of the referenced items when it is desired that it be stored elsewhere, or divided into extents; or the primary metadata may be contained in the MetaBox (e.g. in an XMLBox). Either this box shall occur, or there shall be a box within the MetaBox (e.g. an XMLBox) containing the primary information in the format required by the identified handler.

*to*

The primary data may be one of the referenced items when it is desired that it be stored elsewhere, or divided into extents; or the primary metadata may be contained in the MetaBox (e.g. in an XMLBox). Either this box shall occur, or there shall be a box within the MetaBox (e.g. an XMLBox) containing the primary information in the format required by the identified handler.

*Add the following as clause 8.11.16*

**8.11.16 Handler property**

**8.11.16.1 Definition**

Box Type: 'hdlp'  
Property Type: Descriptive item property  
Container: ItemPropertyContainerBox  
Mandatory: No  
Quantity: zero or more

HandlerProperty provides a mapping of a media handler with an item in a MetaBox. Items that are alternatives of each other shall have the same handler property, or none.

**8.11.16.2 Syntax**

aligned(8) class HandlerProperty extends ItemFullProperty('hdlp', version=0, flags=0) {  
 unsigned int(32) handler\_type;  
}

**8.11.16.3 Semantics**

handler\_type is a four-character-code which corresponds to a media handler type. When the HandlerProperty is associated with the primary item, handler\_type of the HandlerProperty shall be equal to the handler\_type of the MetaBox. When no specific handler type needs to be signalled for an item, the HandlerProperty for the item may be absent or the handler\_type may be 'null'.

*In 8.12.1 change*

1. The four character code of the sample description is replaced with a four character code indicating protection encapsulation: these codes vary only by media-type. For example, 'mp4v' is replaced with 'encv' and 'mp4a' is replaced with 'enca'.
2. A ProtectionSchemeInfoBox (*defined below*) is added to the sample description, leaving all other boxes unmodified.

*to*

1. The four character code of the sample entry is replaced with a four character code indicating protection encapsulation: these codes vary only by media-type. For example, 'mp4v' is replaced with 'encv' and 'mp4a' is replaced with 'enca'.
2. A ProtectionSchemeInfoBox (*defined below*) is added to the sample entry, leaving all other boxes unmodified.

*In 8.12.2.2 change*

aligned(8) class ProtectionSchemeInfoBox(fmt) extends Box('sinf') {  
 OriginalFormatBox(fmt) original\_format;  
  
 SchemeTypeBox scheme\_type\_box; // optional  
 SchemeInformationBox info; // optional  
}

*to*

aligned(8) class ProtectionSchemeInfoBox(fmt) extends Box('sinf') {  
 OriginalFormatBox(fmt) original\_format;  
 // mandatory for sample protection,   
 // shall not be present for item protection  
  
 SchemeTypeBox scheme\_type\_box; // optional  
 SchemeInformationBox info; // optional  
}

*In 8.12.3.1 change*

The OriginalFormatBox contains the four character code of the original un-transformed sample description.

*to*

The OriginalFormatBox contains the four character code of the original un-transformed sample entry.

*In 8.15.2 revise*

A restricted sample entry is defined as a sample entry on which the following transformation procedure has been applied:

1. The four character code of the sample entry is replaced by a new sample entry code as defined in table X below.
2. A RestrictedSchemeInfoBox is added to the sampleentry, leaving all other boxes unmodified.
3. The original sample entry type is stored within an OriginalFormatBox contained in the RestrictedSchemeInfoBox.
4. **Table X — Restricted sample-entry codes**

|  |  |  |
| --- | --- | --- |
| **Stream (Track) Type** | **Sample-Entry Code** | **SampleEntry Class** |
| Video | resv | VisualSampleEntry |
| Audio | resa | AudioSampleEntry or AudioSampleEntryV1 |
| Metadata | resm | MetaDataSampleEntry |
| Text | rest | SimpleTextSampleEntry |
| Subtitle | resu | XMLSubtitleSampleEntry |
| Systema | ress |  |
| Font | resf | FontSampleEntry |
| Haptics | resp | HapticSampleEntry |
| Volumetric visual | res3 | VolumetricVisualSampleEntry |
| a System streams are defined in ISO/IEC 14496-14[22]. | | |

*In 8.15.4 replace*

The following applies when the StereoVideoBox is used:

* In the TrackHeaderBox
  + width and height specify the visual presentation size of a single view after unpacking.
* In the SampleDescriptionBox

*with*

The following applies when the StereoVideoBox is used:

* In the TrackHeaderBox
  + width and height specify the visual presentation size of a single view after unpacking.
* In a SampleEntry in the SampleDescriptionBox

*In 8.17.2 replace*

A CompleteTrackInfoBox is added to the sample description, leaving all other boxes unmodified.

*with*

A CompleteTrackInfoBox is added to the sample entry, leaving all other boxes unmodified.

*Add new 8.18.4 and 8.18.5*

8.18.4 Label and group label box

8.18.4.1 Definition

Box Type: 'labl'  
Container: Container User data box (‘udta’) in a track, Preselection Information Box ('prsi')   
Mandatory: No  
Quantity: Zero or more

Labels provide the ability to annotate data structures in an ISOBMFF file to provide a description of the context of the element to which the label is assigned. Such labels may for example be used by playback clients to provide a selection choice to the user. The label may also be used for simple annotation in another context.

In addition, a GroupLabel element may be added on a higher level in order to provide a summary or title of the labels collected in a group. An example may be that this is used in a menu in order to provide a context of the menu of the labels.

Multiple Labels can be used to provide the textual description. To annotate the preselection to a multilingual audience, the annotation can be provided in a language different from that of the preselection.

If the is\_group\_label is set to a value different from zero, the label text in this box specifies a summary or title of all labels with the same label\_id. This may be used as the title on a selection menu containing a collection of labels.

8.18.4.2 Syntax

aligned(8) class LabelBox   
 extends FullBox('labl', version=0, 0 ){  
 unsigned int(8) is\_group\_label = 0;  
 unsigned int(16) label\_id = 0;  
 utf8string language;  
 utf8string label;  
}

8.18.4.3 Semantics

is\_group\_label specifies if the label contains a summary label for a group of labels.

label\_id is an integer that contains an identifier for the label. Labels with the same value belong to a label group. The default value of zero indicates that the label does not belong to any label group.

language is a NULL-terminated C string containing an RFC 4646 (BCP 47) compliant language tag string, such as "en-US", "fr-FR", or "zh-CN", the language being the language the label is targeted at.

label is a NULL-terminated C string containing the textual description.

8.18.5 Audio rendering indication box

8.18.5.1 Definition

Box Type: 'ardi'  
Container: Preselection Information Box ('prsi')   
Mandatory: No  
Quantity: Zero or one

The audio rendering indication box contains a hint for a preferred reproduction channel layout.

8.18.5.2 Syntax

aligned(8) class AudioRenderingIndicationBox   
 extends FullBox('ardi', version=0, 0 ){  
 unsigned int(8) audio\_rendering\_indication = 0;  
}

8.18.5.3 Semantics

audio\_rendering\_indication contains a hint for a preferred reproduction channel layout, coded according to table 2.

Table 2 — Coding of audio rendering indication

|  |  |
| --- | --- |
| **audio\_rendering\_indication** | **Description** |
| 0 | no preference given for the reproduction channel layout |
| 1 | preferred reproduction channel layout is stereo |
| 2 | preferred reproduction channel layout is two-dimensional (e.g. 5.1 multi-channel) |
| 3 | preferred reproduction channel layout is three-dimensional |
| 4 | content is pre-rendered for consumption with headphones |
| 5 to 255 | reserved for future use |

*Amended table 1 from section 8.14.4 removed in this contribution – unchanged compared to N20538.*

*Change the titles of 9.1.2, 9.2.3, 9.3.3, 9.4.1.2, 9.4.2.3, 9.4.3.2, 9.4.4.3 to* "Sample entry format"

*In 9.1.3 change*

The sampledescription mode allows sending of sample descriptions (which would contain elementary stream descriptors), by reference, as part of an RTP packet. The index is the index of a SampleEntry in a SampleDescriptionBox, and the offset is relative to the beginning of that SampleEntry.

*to*

The sampledescription mode allows sending of sample entries (which would contain elementary stream descriptors), by reference, as part of an RTP packet. The index is the index of a SampleEntry in a SampleDescriptionBox, and the offset is relative to the beginning of that SampleEntry.

*In 9.1.4.3 change*

At the track level, the structure is similar; however, we already know that this track is an RTP hint track, from the sample description. Therefore the child box merely specifies the description format.

*to*

At the track level, the structure is similar; however, we already know that this track is an RTP hint track, from the sample entry. Therefore the child box merely specifies the format.

*In 9.3.2.3 change*

The optional TSTimingBox in the sample description can be used

*to*

The optional TSTimingBox in the sample entry can be used

*In 9.3.2.5 replace*

It is recommended that the PSI/SI be in the Sample Description so that true random-access with just the media data is possible.

*with*

It is recommended that the PSI/SI be in the sample entry so that true random-access with just the media data is possible.

*In 9.3.2.6 change*

The format of the reception hint samples is indicated by the sample description for the reception hint track.

*to*

The format of the reception hint samples is indicated by the sample entry for the reception hint track.

*In 9.3.3.1 change*

The sample description for an MPEG2-TS reception hint track contains all static metadata that describe the stream or a portion thereof, especially the PSI/SI tables. MPEG-2 TS reception hint tracks use an entry-format in the sample description of 'rm2t' (which indicates *MPEG-2 transport stream*). The entry-format for MPEG2‑TS server hint tracks is 'sm2t'.

*to*

The sample entry for an MPEG2-TS reception hint track contains all static metadata that describe the stream or a portion thereof, especially the PSI/SI tables. MPEG-2 TS reception hint tracks use an entry-format in the sample entry of 'rm2t' (which indicates *MPEG-2 transport stream*). The entry-format for MPEG2‑TS server hint tracks is 'sm2t'.

*In 9.3.4.1 change*

Each MPEG-2 TS packet in the sample may be preceded with a preheader (precedingbytes), or followed by a posttrailer (trailingbytes), as detailed in the Sample Description Format. The size of the preheader and the posttrailer are specified by precedingbyteslen and trailingbyteslen, respectively, in the sample description to allow compact sample tables with fewer chunks.

*to*

Each MPEG-2 TS packet in the sample may be preceded with a preheader (precedingbytes), or followed by a posttrailer (trailingbytes), as detailed in the sample entry. The size of the preheader and the posttrailer are specified by precedingbyteslen and trailingbyteslen, respectively, in the sample entry to allow compact sample tables with fewer chunks.

*In 9.4.1.2 change*

The entry-format in the sample description for the RTP reception hint tracks is 'rrtp'. The syntax of the sample entry is the same as for RTP server hint tracks having the entry-format 'rtp '.

class ReceivedRtpHintSampleEntry() extends HintSampleEntry ('rrtp') {  
 uint(16) hinttrackversion = 1;  
 uint(16) highestcompatibleversion = 1;  
 uint(32) maxpacketsize;  
}

The entry-format identifier in the sample description of the RTP reception hint track is different from the entry-format in the sample description of the RTP server hint track, in order to avoid using an RTP reception hint track that contains errors as a valid server hint track.

…

The SSRC value shall equal the SSRC value in the header of all recorded SRTP packets described by the sample description.

*to*

The entry-format in the sample entry for the RTP reception hint tracks is 'rrtp'. The syntax of the sample entry is the same as for RTP server hint tracks having the entry-format 'rtp '.

class ReceivedRtpHintSampleEntry() extends HintSampleEntry ('rrtp') {  
 uint(16) hinttrackversion = 1;  
 uint(16) highestcompatibleversion = 1;  
 uint(32) maxpacketsize;  
}

The entry-format identifier in the sample entry of the RTP reception hint track is different from the entry-format in the sample entry of the RTP server hint track, in order to avoid using an RTP reception hint track that contains errors as a valid server hint track.

…

The SSRC value shall equal the SSRC value in the header of all recorded SRTP packets described by the sample entry.

*In 9.4.2.3 change*

The entry-format in the sample description for the RTCP reception hint tracks is 'rtcp'. It is otherwise identical in structure to the sample entry format for RTP. There are no defined boxes for the additionaldata field.

*to*

The entry-format in the sample entry for the RTCP reception hint tracks is 'rtcp'. It is otherwise identical in structure to the sample entry format for RTP. There are no defined boxes for the additionaldata field.

*Change the title of 9.4.3.2.1 to "Sample entry" and in it change*

The sample description format for SRTP reception hint tracks is identical to that for RTP reception hint tracks with the exception that the sample entry name is changed from 'rrtp' to 'rsrp' and that it may contain additional boxes:

class ReceivedSrtpHintSampleEntry() extends HintSampleEntry ('rsrp') {  
 uint(16) hinttrackversion = 1;  
 uint(16) highestcompatibleversion = 1;  
 uint(32) maxpacketsize;  
}

Fields and boxes are identical to those of the ReceivedRtpHintSampleEntry ('rrtp'). The addtionaldata[] of each sample description entry of a SRTP reception hint track shall contain exactly one ReceivedSsrcBox.

*to*

The sample entry format for SRTP reception hint tracks is identical to that for RTP reception hint tracks with the exception that the sample entry name is changed from 'rrtp' to 'rsrp' and that it may contain additional boxes:

class ReceivedSrtpHintSampleEntry() extends HintSampleEntry ('rsrp') {  
 uint(16) hinttrackversion = 1;  
 uint(16) highestcompatibleversion = 1;  
 uint(32) maxpacketsize;  
}

Fields and boxes are identical to those of the ReceivedRtpHintSampleEntry ('rrtp'). The addtionaldata[] of each sample entry of a SRTP reception hint track shall contain exactly one ReceivedSsrcBox.

*In 9.4.4.3 change*

The entry-format in the sample description for the SRTCP reception hint tracks is 'stcp'.

*to*

The entry-format in the sample entry for the SRTCP reception hint tracks is 'stcp'.

*In 10.8.1, change the following:*

A dependent random access point (DRAP) sample is a sample after which all samples in decoding order can be correctly decoded if the closest *initial* sample preceding the DRAP sample is available for reference.

*to the following (adding "*and in output order*"):*

A dependent random access point (DRAP) sample is a sample after which all samples in decoding order and in output order can be correctly decoded if the closest *initial* sample preceding the DRAP sample is available for reference.

*In 10.8.1, change the following:*

* The DRAP sample and all samples following the DRAP sample in output order can be correctly decoded when starting decoding at the DRAP sample after having decoded the closest preceding SAP sample of type 1, 2 or 3 marked as such by being a Sync sample or by the SAP sample group.

*to the following (adding "in decoding order and"):*

* The DRAP sample and all samples following the DRAP sample in decoding order and in output order can be correctly decoded when starting decoding at the DRAP sample after having decoded the closest preceding SAP sample of type 1, 2 or 3 marked as such by being a Sync sample or by the SAP sample group.

*In 10.8.1, change the following:*

NOTE DRAP samples can only be used in combination with SAP samples of type 1, 2 and 3. This is in order to enable the functionality of creating a decodable sequence of samples by concatenating the preceding SAP sample with the DRAP sample and the samples following the DRAP sample in output order

*to the following (adding "* in decoding order and*", and adding period '.' at the end of the sentence):*

NOTE DRAP samples can only be used in combination with SAP samples of type 1, 2 and 3. This is in order to enable the functionality of creating a decodable sequence of samples by concatenating the preceding SAP sample with the DRAP sample and the samples following the DRAP sample in decoding order and in output order.

*Add new subclause 10.11*

**10.11 Extended DRAP (EDRAP) sample group**

**10.11.1 Definition**

This sample group is similar to the DRAP sample group as specified in subclause 10.8; however, it enables more flexible cross-RAP referencing.

An EDRAP sample is a sample after which all samples in decoding order and in output order can be correctly decoded if the closest initial sample preceding the EDRAP sample and one or more other identified EDRAP samples earlier in decoding order than the EDRAP sample are available for reference.

The initial sample is a SAP sample of SAP type 1, 2 or 3 that is marked as such either by being a Sync sample or by the SAP sample group. For example, if the 32nd sample in a file is an initial sample consisting of an I-picture, the 48th sample may consist of a P-picture and be marked as a member of the EDRAP sample group, thereby indicating that random access can be performed at the 48th sample by first decoding the 32nd sample (ignoring samples 33-47) and then continuing to decode from the 48th sample.

NOTE: EDRAP samples can only be used in combination with SAP samples of type 1, 2 and 3. This is in order to enable the functionality of creating a decodable sequence of samples by concatenating the preceding SAP sample and zero or more other identified EDRAP samples earlier in decoding order than the EDRAP sample with the EDRAP sample and the samples following the EDRAP sample in decoding order and in output order.

A sample can be a member of the EDRAP Sample Group only if the following conditions are true:

* The EDRAP sample references only the closest preceding initial sample and one or more other identified EDRAP samples earlier in decoding order than the EDRAP sample.
* The EDRAP sample and all samples following the EDRAP sample in decoding order and output order can be correctly decoded when starting decoding at the EDRAP sample after having decoded the closest preceding SAP sample of type 1, 2 or 3 marked as such by being a Sync sample or by the SAP sample group and after having decoded the zero or more other identified EDRAP samples earlier in decoding order than the EDRAP sample.

**10.11.2 Syntax**

class VisualEdrapEntry()  
extends VisualSampleGroupEntry('edrp') {  
 unsigned int(3) edrap\_type;  
 unsigned int(3) num\_ref\_edrap\_pics;  
 unsigned int(26) reserved = 0;  
 for(i=0; i<num\_ref\_edrap\_pics; i++)  
 unsigned int(16) ref\_edrap\_idx\_delta[i];  
}

**10.11.3 Semantics**

edrap\_type is a non-negative integer. When edrap\_type is in the range of 1 to 3 it indicates the SAP\_type (as specified in Annex I) that the EDRAP sample would have corresponded to, had it not depended on the closest preceding SAP or other EDRAP samples. Other type values are reserved.

num\_ref\_edrap\_pics indicates the number of other EDRAP samples that are earlier in decoding order than the EDRAP sample and are needed for reference to be able to correctly decode the EDRAP sample and all samples following the EDRAP sample in both decoding and output order when starting decoding from the EDRAP sample.

reserved shall be equal to 0. The semantics of this subclause only apply to sample group description entries with reserved equal to 0. Parsers shall allow and ignore sample group description entries with reserved greater than 0 when parsing this sample group.

ref\_edrap\_idx\_delta[i] indicates the difference between the sample group index (i.e., the index to the list of all samples in this sample group in decoding order) of this EDRAP sample and the sample group index of the i-th RAP sample that is earlier in decoding order than the EDRAP sample and is needed for reference to be able to correctly decode the EDRAP sample and all samples following the EDRAP sample in both decoding and output order when starting decoding from this EDRAP sample. The value 1 indicates that the i-th RAP sample is the latest RAP sample in the sample group and preceding this EDRAP sample in decoding order, the value 2 indicates that the i-th RAP sample is the second latest RAP sample in the sample group and preceding this EDRAP sample in decoding order, and so on.

*Add new subclause 10.12*

**10.12 Essential descriptions hierarchy sample grouping**

**10.12.1 Definition**

The essential descriptions hierarchy sample group ('esgh') indicates the processing order of the essential sample group descriptions applying to a given sample. This sample group description is an essential sample group description and shall use version 3. It shall be present if at least one essential sample group description with grouping\_type other than 'esgh' is present.

Each essential sample group description, except the essential descriptions hierarchy sample group itself, shall be listed in the EssentialDescriptionsHierarchyEntry.

The grouping\_type\_paramater for an essential descriptions hierarchy sample group description is not defined, and its value shall be set to 0.

The syntax of EssentialDescriptionsHierarchyEntry is the same for all media types.

Samples associated with essential sample groups shall use a restricted sample entry indicating the original media type (e.g. 'resv', 'resa') with a scheme\_type equal to 'essg'. In a sample entry, there shall be at most one sample entry transformation with a scheme\_type equal to 'essg'. If such a transformation is present:

* The transformation shall be the first sample entry transformation,
* There shall be either no other sample entry transformations, or at most one sample entry transformation of type protection, as defined in 8.12.1 (e.g., 'encv')

The transformations given in sample\_group\_description\_type are listed in the order in which a file reader shall apply each transformation: any sample processing described by a sample group of type sample\_group\_description\_type[i] shall be applied before any sample processing described by a sample group of type sample\_group\_description\_type[i+1].

In the sample\_group\_description\_type list, the following transformation values are reserved:

* 'stsd' : indicates the position of the decoding process in the transformation chain.
* 'cenc' : indicates the position of the protection process in the transformation chain.

If 'stsd' is absent from the list of sample\_group\_description\_type, all listed transformations shall apply to decoded samples. If 'cenc' is present in the list, 'stsd' shall be present.

**10.12.2 Syntax**

Class EssentialDescriptionsHierarchyEntry ()  
extends SampleGroupDescriptionEntry ('esgh')  
{  
 unsigned int(32) num\_groupings;  
 unsigned int(32) sample\_group\_description\_type[num\_groupings];  
}

**10.12.3 Semantics**

num\_groupings indicates the number of essential sample group description types listed.

sample\_group\_description\_type indicates the four-character code of the essential sample group description (i.e. with version 3) or reserved transformation values applying to the associated samples.

*In 11 change*

As defined above, the Sample Description format may be extended with optional or required boxes. The usual syntax for doing this would be to define a new box with a specific name, extending (for example) Visual Sample Entry, and containing new boxes.

*to*

As defined above, a sample entry may be extended with optional or required boxes. The usual syntax for doing this would be to define a new box with a specific name, extending (for example) VisualSampleEntry, and containing new boxes.

*In 12.1.3.3 change*

width and height are the maximum visual width and height of the stream described by this sample description, in pixels

*to*

width and height are the maximum visual width and height of the stream described by this sample entry, in pixels

*In 12.2.4.1 change*

Box Type: 'chnl'  
Container: Audio sample entry or Preselection Information Box  
Mandatory: No  
Quantity: Zero or one

*In 12.4.1 change*

Hint tracks are used to describe elementary stream data in the file. Each protocol or each family of related protocols has its own hint track format. A server hint track format and a reception hint track format for the same protocol are distinguishable from the associated four character code of the sample description entry. In other words, a different four character code is used for a server hint track and a reception hint track of the same protocol. The syntax of the server hint track format and the reception hint track format for the same protocol should be the same or compatible so that a reception hint track can be used for re-sending of the stream provided that the potential degradations of the received streams are handled appropriately. Most protocols will need only one sample description format for each track.

Servers find their hint tracks by first finding all hint tracks, and then looking within that set for server hint tracks using their protocol (sample description format). If there are choices at this point, then the server chooses on the basis of preferred protocol or by comparing features in the hint track header or other protocol-specific information in the sample descriptions. Particularly in the absence of server hint tracks, servers may also use reception hint tracks of their protocol. However, servers should handle potential degradations of the received stream described by the used reception hint track appropriately.

*to*

Hint tracks are used to describe elementary stream data in the file. Each protocol or each family of related protocols has its own hint track format. A server hint track format and a reception hint track format for the same protocol are distinguishable from the associated four character code of the sample entry. In other words, a different four character code is used for a server hint track and a reception hint track of the same protocol. The syntax of the server hint track format and the reception hint track format for the same protocol should be the same or compatible so that a reception hint track can be used for re-sending of the stream provided that the potential degradations of the received streams are handled appropriately. Most protocols will need only one sample entry format for each track.

Servers find their hint tracks by first finding all hint tracks, and then looking within that set for server hint tracks using their protocol (identified by the sample entry). If there are choices at this point, then the server chooses on the basis of preferred protocol or by comparing features in the hint track header or other protocol-specific information in the sample entries. Particularly in the absence of server hint tracks, servers may also use reception hint tracks of their protocol. However, servers should handle potential degradations of the received stream described by the used reception hint track appropriately.

*In 12.4.4.1 change*

For hint tracks, the sample description contains appropriate declarative data for the streaming protocol being used, and the format of the hint track. The definition of the sample description is specific to the protocol.

The ‘protocol’ and ‘codingname’ fields are registered identifiers that uniquely identify the streaming protocol or compression format decoder to be used. A given protocol or codingname may have optional or required extensions to the sample description (e.g. codec initialization parameters). All such extensions shall be within boxes; these boxes occur after the required fields. Unrecognized boxes shall be ignored.

*to*

For hint tracks, the sample entry contains appropriate declarative data for the streaming protocol being used, and the format of the hint track. The definition of the sample entry is specific to the protocol.

The protocol (codingname) field is a registered identifier that uniquely identifies the streaming protocol or compression format decoder to be used. A given protocol may have optional or required extensions to the sample entry (e.g. codec initialization parameters). All such extensions shall be within boxes; these boxes occur after the required fields. Unrecognized boxes shall be ignored.

*In A.4 change*

Each track has one or more **sample description**s; each sample in the track is tied to a description by reference.

to

Each track has one or more **sample entries**; each sample in the track is tied to an entry by reference.

*In A.10 change*

The basic 'shape' of the movie is set in initial MovieBox: the number of tracks, the available sample descriptions, width, height, composition, and so on.

*to*

The basic 'shape' of the movie is set in initial MovieBox: the number of tracks, the available sample entries, width, height, composition, and so on.

*In B.2.2 change*

These operations might include the obvious reading tracks, finding the data and timing for samples, and their sample description and track type, and so on.

*to*

These operations might include the obvious reading tracks, finding the data and timing for samples, and their sample entry and track type, and so on.

*In Annex D replace*

File format sample description and sample format identifiers (also known as codec names).

*with*

File format sample entry and sample format identifiers (also known as codec names).

*In E.2 change*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | stsd | *sample descriptions (codec types, initialization etc.)* |

*to*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | stsd | *sample description (codec types, initialization etc.)* |

*In E.10, replace the bullet "Recognizing incomplete tracks" with*

* Recognizing incomplete tracks by detecting the following sample entries for incomplete tracks: 'icpv', 'icpa', 'icpt', 'icps', 'icph', 'icpp', 'icp3' and 'icpm'.

Note The process of detecting when a track becomes incomplete (before the transformation specified in subclause 8.17.2) and handling incomplete tracks in playback are outside the scope of this specification.

*In E.14, after*

The brand 'isob' requires support for all features of the 'isoa' brand.

*insert*

Support for the following boxes is required under this brand:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | ttyp |  |  |  |  |  | *track type of the track* |
|  |  |  | brnd |  |  |  |  | *brand property* |

Insert at the beginning of E.15.2

Support for the following boxes is required under this brand:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| imda |  |  |  |  |  |  |  | *identified media data* |
|  |  |  | imdt |  |  |  |  | *data entry of imda* |
|  |  |  | snim |  |  |  |  | *data entry sequence number for imda* |

*In section K.3 replace bullet 3 with*

3. When dataFormat indicates a transformed media track:

1. If the transformation type indicates an essential sample group (scheme\_type equal to 'essg'), the value of the codecs MIME parameter is appended by the four-character codes listed in the Essential Descriptions Hierarchy sample description, from the first entry up to but excluding the first occurrence of 'stsd' or 'cenc'. A dot ('.') shall be used to separate the four-character codes.
2. Otherwise, the value of the codecs MIME parameter is appended by the scheme\_type four-character code contained in the SchemeTypeBox of schemeInfoContainerBox.

*In K.4 change the note from*

NOTE This document requires that the major brand be repeated in the compatible-brands, but this requirement is relaxed in the 'profiles' parameter for compactness.

*to*

NOTE This document recommends that the major brand be repeated in the compatible-brands, but this requirement is relaxed in the 'profiles' parameter for compactness.

*Add K.6*

**K.6 Use of the 'essential' parameter**

For files containing essential sample group descriptions, the ‘essential’ parameter, when used, is composed of one or more comma-separated essential hierarchy descriptions.

Each essential hierarchy description is composed of one or more four-character code of essential sample group descriptions, separated with a dot.

* If the ‘codecs’ parameter includes description of the transformation used, the listed four-character codes shall be the ones listed in the Essential Descriptions Hierarchy sample group description, in the same order, from the first code following the last occurrence of 'stsd' until the last listed code.
* Otherwise, the listed four-character codes shall be the ones listed in the Essential Descriptions Hierarchy sample group description in the same order.

Example:

An HEVC sample is encrypted by means other than CENC, signaled through an essential sample group of type 'FOOv'. The resulted decoded sample shall have a post-processing filter applied, signaled through an essential sample group of type 'BARv'. The EssentialDescriptionsHierarchyEntry will list the transformations as ['FOOv', 'stsd', 'BARv'].

The ‘codecs’ and ‘essential’ mime type sub-parameter may be:

codecs=resv.FOOv.hvc1.1.6.L186.80

essential=BARv

or

codecs=hvc1.1.6.L186.80

essential=FOOv.stsd.BARv