ISO/IEC JTC 1/SC 29/WG 03 N0341

**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: Improved brand documentation and other improvements

**Status:** Approved

**Date of document:** 2021-07-29

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

**No. of pages:** 2 (with cover page)

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

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

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N0341**

**July 2021, Virtual**

|  |  |
| --- | --- |
| **Title** | **Working draft of amendment to ISO/IEC 14496-12: Improved brand documentation and other improvements** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **20688** |

 sample entry vs. sample description, consistency/correctness

M57443

M57441

*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 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.

*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 8.11.1.1 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 replace*

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

*with*

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

*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.

*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 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 12.4.1 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 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.