INTERNATIONAL STANDARD

ISO/IEC 14496-12

Second edition
2005-04-01

Corrected version
2005-10-01

AMENDMENT 1
2007-04-01

Information technology — Coding of audio-visual objects —

Part 12: ISO base media file format

AMENDMENT 1: Support for timed metadata, non-square pixels and improved sample groups

Technologies de l'information — Codage des objets audiovisuels —

Partie 12: Format ISO de base pour les fichiers médias

AMENDEMENT 1: Support pour métadonnées temporisées, pixels «non-square» et groupes d'échantillons améliorés

Reference number
ISO/IEC 14496-12:2005/Amd.1:2007(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 14496-12:2005 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 12:
ISO base media file format

AMENDMENT 1: Support for timed metadata, non-square pixels and improved sample groups

Add the following to Clause 2, Normative references:

ISO/IEC 15938-1, *Information technology — Multimedia content description interface — Part 1: Systems*

ISO/IEC 23001-1, *Information technology — MPEG systems technologies — Part 1: Binary MPEG format for XML*

Add to 8.5.1:

The width and height in the track header are measured on a notional 'square' (uniform) grid. Track video data is normalized to these dimensions (logically) before any transformation or placement caused by a layup or composition system. Track (and movie) matrices, if used, also operate in this uniformly-scaled space.

Add to the narrative in 8.9.1:

There is a general handler for metadata streams of any type; the specific format is identified by the sample entry, as for video or audio, for example. If they are in text, then a MIME format is supplied to document their format; if in XML, each sample is a complete XML document, and the namespace of the XML is also supplied.

> Note that MPEG-7 streams, which are a specific kind of metadata stream, have their own handler declared, documented in the MP4 file format [ISO/IEC 14496-14].
>
> Note that metadata tracks are linked to the track they describe using a track-reference of type `'cdsc'`. Metadata tracks use a null media header (`'nmhd'`), as defined in sub-clause 8.9.1.

Add to the handler types in 8.9.3 the 'meta' handler:

`handler_type` when present in a media box, is an integer containing one of the following values, or a value from a derived specification:

`'vide'`	Video track
`'soun'`	Audio track
`'hint'`	Hint track
`'meta'`	Timed Metadata track

Clarify the narrative in 8.11.5:

Streams other than visual and audio (e.g., timed metadata streams) may use a null Media Header Box, as defined here.

Amend 8.16.1 as follows:

change:

For video tracks, a VisualSampleEntry is used; for audio tracks, an AudioSampleEntry. Hint tracks use an entry format specific to their protocol, with an appropriate name.

to:

For video tracks, a VisualSampleEntry is used**,** for audio tracks, an AudioSampleEntry and for metadata tracks, a MetaDataSampleEntry. Hint tracks use an entry format specific to their protocol, with an appropriate name.

Add at the end of the section:

An optional `BitRateBox` may be present at the end of any MetaDataSampleEntry to signal the bit rate information of a stream. This can be used for buffer configuration. In case of XML metadata it can be used to choose the appropriate memory representation format (DOM, STX).

The width and height in the video sample entry document the pixel counts that the codec will deliver; this enables the allocation of buffers. Since these are counts they do not take into account pixel aspect ratio.

The pixel aspect ratio and clean aperture of the video may be specified using the 'pasp' and 'clap' sample entry boxes, respectively. These are both optional; if present, they over-ride the declarations (if any) in structures specific to the video codec, which structures should be examined if these boxes are absent.

In the PixelAspectRatioBox, `hSpacing` and `vSpacing` have the same units, but those units are unspecified: only the ratio matters. `hSpacing` and `vSpacing` may or may not be in reduced terms, and they may reduce to 1/1. Both of them must be positive.

They are defined as the aspect ratio of a pixel, in arbitrary units. If a pixel appears H wide and V tall, then hSpacing/vSpacing is equal to H/V. This means that a square on the display that is n pixels tall needs to be n*vSpacing/hSpacing pixels wide to appear square.

> Note: *When adjusting pixel aspect ratio, normally, the horizontal dimension of the video is scaled, if needed (i.e. if the final display system has a different pixel aspect ratio from the video source).*
>
> Note: *It is recommended that the original pixels, and the composed transform, be carried through the pipeline as far as possible. If the transformation resulting from 'correcting' pixel aspect ratio to a square grid, normalizing to the track dimensions, composition or placement (e.g. track and/or movie matrix), and normalizing to the display characteristics, is a unity matrix, then no re-sampling need be done. In particular, video should not be re-sampled more than once in the process of rendering, if at all possible.*

There are notionally four values in the CleanApertureBox. These parameters are represented as a fraction N/D. The fraction may or may not be in reduced terms. We refer to the pair of parameters `fooN` and `fooD` as `foo`. For `horizOff` and `vertOff`, D must be positive and N may be positive or negative. For `cleanApertureWidth` and `cleanApertureHeight`, both N and D must be positive.

> Note: *these are fractional numbers for several reasons. First, in some systems the exact width after pixel aspect ratio correction is integral, not the pixel count before that correction. Second, if video is resized in the full aperture, the exact expression for the clean aperture may not be integral. Finally, because this is represented using centre and offset, a division by two is needed, and so half-values can occur.*

Considering the pixel dimensions as defined by the VisualSampleEntry width and height. If picture centre of the image is at `pcX` and `pcY`, then `horizOff` and `vertOff` are defined as follows:

```
pcX = horizOff + (width  - 1)/2
pcY = vertOff  + (height - 1)/2;
```

Typically, `horizOff` and `vertOff` are zero, so the image is centred about the picture centre.

The leftmost/rightmost pixel and the topmost/bottommost line of the clean aperture fall at:

```
pcX ± (cleanApertureWidth - 1)/2
pcY ± (cleanApertureHeight - 1)/2;
```

Add to the beginning of 8.16.2:

```
class PixelAspectRatioBox extends Box('pasp'){
   unsigned int(32) hSpacing;
   unsigned int(32) vSpacing;
}

class CleanApertureBox extends Box('clap'){
   unsigned int(32) cleanApertureWidthN;
   unsigned int(32) cleanApertureWidthD;

   unsigned int(32) cleanApertureHeightN;
   unsigned int(32) cleanApertureHeightD;


   unsigned int(32) horizOffN;
   unsigned int(32) horizOffD;


   unsigned int(32) vertOffN;
   unsigned int(32) vertOffD;

}

class BitRateBox extends Box('btrt'){
   unsigned int(32) bufferSizeDB;
   unsigned int(32) maxBitrate;
   unsigned int(32) avgBitrate;
}

class MetaDataSampleEntry(codingname) extends SampleEntry (codingname) {
}

class XMLMetaDataSampleEntry() extends MetaDataSampleEntry ('metx') {
   string   content_encoding; // optional
   string   namespace;
   string   schema_location; // optional
   BitRateBox (); // optional
}

class TextMetaDataSampleEntry() extends MetaDataSampleEntry ('mett') {
   string   content_encoding; // optional
   string   mime_format;
   BitRateBox (); // optional
}
```

Amend VisualSampleEntry in 8.16.2 as follows:

```
class VisualSampleEntry(codingname) extends SampleEntry (codingname){
   unsigned int(16) pre_defined = 0;
   const unsigned int(16) reserved = 0;
   unsigned int(32)[3]  pre_defined = 0;
   unsigned int(16)  width;
   unsigned int(16)  height;
   template unsigned int(32)  horizresolution = 0x00480000; // 72 dpi
   template unsigned int(32)  vertresolution  = 0x00480000; // 72 dpi
   const unsigned int(32)  reserved = 0;
   template unsigned int(16)  frame_count = 1;
   string[32]  compressorname;
   template unsigned int(16)  depth = 0x0018;
   int(16)  pre_defined = -1;
   CleanApertureBox     clap; // optional
   PixelAspectRatioBox  pasp;    // optional
}
```

Change the definition of SampleDescriptionBox in 8.16.2, adding the lines for case 'meta':

```
aligned(8) class SampleDescriptionBox (unsigned int(32) handler_type)
      extends FullBox('stsd', 0, 0){
      int i ;
      unsigned int(32) entry_count;
      for (i = 1 ; i <= entry_count ; i++){
            switch (handler_type){
                  case 'soun': // for audio tracks
                        AudioSampleEntry();
                        break;
                  case 'vide': // for video tracks
                        VisualSampleEntry();
                        break;
                  case 'hint': // Hint track
                        HintSampleEntry();
                        break;
                  case 'meta': // Metadata track
                        MetadataSampleEntry();
                        break;        }
      }
}
```

Add to 8.16.3:

`hSpacing`, `vSpacing`: define the relative width and height of a pixel;
`cleanApertureWidthN`, `cleanApertureWidthD`: a fractional number which defines the exact clean aperture width, in counted pixels, of the video image
`cleanApertureHeightN`, `cleanApertureHeightD`: a fractional number which defines the exact clean aperture height, in counted pixels, of the video image
`horizOffN`, `horizOffD`: a fractional number which defines the horizontal offset of clean aperture centre minus (width-1)/2. Typically 0.
`vertOffN`, `vertOffD`: a fractional number which defines the vertical offset of clean aperture centre minus (height-1)/2. Typically 0.
`content_encoding` - is a null-terminated string in UTF-8 characters, and provides a MIME type which identifies the content encoding of the timed metadata. It is defined in the same way as for an ItemInfoEntry in this specification. If not present (an empty string is supplied) the timed metadata is not encoded. An example for this field is 'application/zip'. Note that no MIME types for BiM

[ISO/IEC 23001-1] and TeM [ISO/IEC 15938-1] currently exist. Thus the experimental MIME types 'application/x-BiM' and 'text/x-TeM' shall be used to identify these encoding mechanisms.

`namespace` - gives the namespace of the schema for the timed XML metadata. This is needed for identifying the type of metadata, e.g. gBSD or AQoS [MPEG-21-7] and for decoding using XML aware encoding mechanisms such as BiM.

`schema_location` - optionally provides an URL to find the schema corresponding to the namespace. This is needed for decoding of the timed metadata by XML aware encoding mechanisms such as BiM.

`mime_format` - provides a MIME type which identifies the content format of the timed metadata. Examples for this field are 'text/html' and 'text/plain'.

`bufferSizeDB` gives the size of the decoding buffer for the elementary stream in bytes.

`maxBitrate` gives the maximum rate in bits/second over any window of one second.

`avgBitrate` gives the average rate in bits/second over the entire presentation.

In 8.40.3.3.1 replace the note as follows:

Note: In version 0 of the entries the base classes for sample group description entries are neither boxes nor have a size is signaled. For this reason, use of version 0 entries is deprecated. When defining derived classes, ensure either that they have a fixed size, or that the size is explicitly indicated with a length field. An implied size (e.g. achieved by parsing the data) is not recommended as this makes scanning the array difficult.

Replace 8.40.3.3.2 with:

```
// Sequence Entry
abstract class SampleGroupDescriptionEntry (unsigned int(32) grouping_type)
{
}


abstract class VisualSampleGroupEntry (unsigned int(32) grouping_type) extends
SampleGroupDescriptionEntry (grouping_type)
{
}


abstract class AudioSampleGroupEntry (unsigned int(32) grouping_type) extends
SampleGroupDescriptionEntry (grouping_type)
{
}


abstract class HintSampleGroupEntry (unsigned int(32) grouping_type) extends
SampleGroupDescriptionEntry (grouping_type)
{
}
```