INTERNATIONAL STANDARD

**ISO/IEC 15444-12**

Fourth edition
2012-07-15

**AMENDMENT 3**
2015-04-01



# Information technology — JPEG 2000 image coding system —

## Part 12: ISO base media file format

## AMENDMENT 3: Font streams and other improvements to file format

*Technologies de l'information — Système de codage d'images JPEG 2000 —*

*Partie 12: Format ISO de base pour les fichiers médias*

*AMENDEMENT 3: Flux de police et autres améliorations de fichiers*

Reference number
ISO/IEC 15444-12:2012/Amd.3:2015(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.

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

# Information technology — JPEG 2000 image coding system —

# Part 12:
# ISO base media file format

## AMENDMENT 3: Font streams and other improvements to file format

*Technologies de l'information -- Système de codage d'images JPEG 2000 -- Partie 12: Format ISO de base pour les fichiers médias -- Amendement 3: Flux de police et autres améliorations de fichiers*

*In 8.3.3.3 before*

— 'hind' this track depends on the referenced hint track, i.e., it should only be used if the referenced hint track is used.

*add*

— 'font' used to indicate that a track uses fonts carried/defined in the referenced track.

*In 8.4.3.3 after*

`'subt'` Subtitle track

*add*

`'fdsm'` Font data stream track

*In 8.5.2.1, after*

— hint tracks an entry format specific to their protocol, with an appropriate name.

*add*

— font tracks FontSampleEntry

*In 8.5.2.2 change*

```
aligned(8) class SampleDescriptionBox (unsigned int(32) handler_type)
      extends FullBox('stsd', version, 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 'subt': // for subtitle tracks
                        SubtitleSampleEntry();
                        break;
                  case 'text': // for plain text tracks
                        TextSampleEntry();
                        break;
                              case 'hint': // Hint track
                        HintSampleEntry();
                        break;
```

```
                    case 'meta': // Metadata track
                          MetadataSampleEntry();
                          break;
                    }
              }
        }
}
```

*to*

```
class FontSampleEntry(codingname) extends SampleEntry (codingname){
       //other boxes from derived specifications
       BitRateBox (); // optional
}
aligned(8) class SampleDescriptionBox (unsigned int(32) handler_type)
       extends FullBox('stsd', version, 0){
       int i ;
       unsigned int(32) entry_count;
       for (i = 1 ; i <= entry_count ; i++){
              SampleEntry ();     // an instance of a class derived from this
       }
}
```

*In 8.6.1.1, add before the paragraph starting "In the following example"*

Some coding systems may allow samples that are used only for reference and not output (e.g. a non-displayed reference frame in video). When any such non-output sample is present in a track, the following applies:

1) A non-output sample shall be given a composition time which is outside the time-range of the samples that are output;

2) An edit list shall be used to exclude the composition times of the non-output samples.

3) When the track includes a CompositionOffsetBox (`'ctts'`),

   a) version 1 of the CompositionOffsetBox shall be used,

   b) the value of `sample_offset` shall be set equal to the most negative number possible (for 32-bit values, $-2^{31}$) for each non-output sample,

   c) the CompositionToDecodeBox (`'cslg'`) should be contained in the SampleTableBox (`'stbl'`) of the track, and

   d) when the CompositionToDecodeBox is present for the track, the value of `leastDecodeToDisplayDelta` field in the box shall be equal to the smallest composition offset in the CompositionOffsetBox excluding the `sample_offset` values for non-output samples.

NOTE Thus, leastDecodeToDisplayDelta is greater than $-2^{31}$.

*In 8.7.2.1 change the header from*

Box Types: `'url '`, `urn '`, `'dref'`

Container: Data Information Box (`'dinf'`)

Mandatory: Yes

Quantity: Exactly one

*to*

Box Types: `'dref'`

Container: Data Information Box (`'dinf'`)

Mandatory: Yes

Quantity: Exactly one



 

Box Types: 'url ', 'urn '

Container: Data Information Box ('dref')

Mandatory: Yes (at least one of 'url ' or 'urn ' shall be present)

Quantity: One or more

*and replace*

The DataEntryBox within the DataReferenceBox shall be either a DataEntryUrnBox or a DataEntryUrlBox.

*with*

The entry_count in the DataReferenceBox shall be 1 or greater; each DataEntryBox within the DataReferenceBox shall be either a DataEntryUrnBox or a DataEntryUrlBox.

*In 8.7.7.1 change the definition of the sub-sample box to read*

Box Type: 'subs'

Container: Sample Table Box ('stbl') or Track Fragment Box ('traf')

Mandatory: No

Quantity: Zero or more

*and add at the end of 8.7.7.1:*

When more than one Sub-Sample Information box is present in the same container box, the value of flags shall differ in each of these Sub-Sample Information boxes. The semantics of flags, if any, shall be supplied for a given coding system. If flags have no semantics for a given coding system, the flags shall be 0.

*In 8.7.7.2 change*

```
aligned(8) class SubSampleInformationBox
       extends FullBox('subs', version, 0) {
       unsigned int(32) entry_count;
       int i,j;
       for (i=0; i < entry_count; i++) {
              unsigned int(32) sample_delta;
              unsigned int(16) subsample_count;
              if (subsample_count > 0) {
                     for (j=0; j < subsample_count; j++) {
                            if(version == 1)
                            {
                                   unsigned int(32) subsample_size;
                            }
                            Else
                            {
                                   unsigned int(16) subsample_size;
                            }
                            unsigned int(8) subsample_priority;
                            unsigned int(8) discardable;
                            unsigned int(32) reserved = 0;
                     }
              }
       }
}
```

*to*

```
aligned(8) class SubSampleInformationBox
       extends FullBox('subs', version, flags) {
       unsigned int(32) entry_count;
       int i,j;
       for (i=0; i < entry_count; i++) {
              unsigned int(32) sample_delta;
              unsigned int(16) subsample_count;
              if (subsample_count > 0) {
                     for (j=0; j < subsample_count; j++) {
                            if(version == 1)
                            {
                                   unsigned int(32) subsample_size;
                            }
                            Else
                            {
                                   unsigned int(16) subsample_size;
                            }
                            unsigned int(8) subsample_priority;
                            unsigned int(8) discardable;
                            unsigned int(32) reserved = 0;
                     }
              }
       }
}
```

*In 8.8.7.1, add before "The following flags" the following paragraph*

The base-data-offset, if explicitly provided, is a data offset that is identical to a chunk offset in the Chunk Offset Box, i.e. applying to the complete file (e.g. starting with a file-type box and movie box). In circumstances when the complete file does not exist or its size is unknown, it may be impossible to use an explicit base-data-offset; then, offsets need to be established relative to the movie fragment.

*Add to the end of 8.8.12.1*

If the time expressed in the track fragment decode time ('tfdt') box exceeds the sum of the 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.

In particular, an empty track fragment (with no samples, but with a track fragment decode time box) may be used to establish the duration of the last sample.

*In 8.9.2.3 semantics, replace*

If the sum of the sample count in this box is less than the total sample count, then the reader should effectively extend it with an entry that associates the remaining samples with no group.

*with*

If the sum of the sample count in this box is less than the total sample count, or there is no sample-to-group box that applies to some samples (e.g. it is absent from a track fragment), then the reader should associates the samples that have no explicit group association with the default group defined in the SampleDescriptionGroup box, if any, or else with no group.

*In 8.9.3.2 replace the definition of* `SampleGroupDescriptionBox` *with the following*

```
aligned(8) class SampleGroupDescriptionBox (unsigned int(32) handler_type)
       extends FullBox('sgpd', version, 0){
       unsigned int(32) grouping_type;
       if (version>=1) { unsigned int(32) default_length; }
       if (version>=2) {
              unsigned int(32) default_sample_description_index;
       }
       unsigned int(32) entry_count;
```

```
        int i;
        for (i = 1 ; i <= entry_count ; i++){
                if (version>=1) {
                        if (default_length==0) {
                                unsigned int(32) description_length;
                        }
                }
                switch (handler_type){
                        case 'vide': // for video tracks
                                VisualSampleGroupEntry (grouping_type);
                                break;
                        case 'soun': // for audio tracks
                                AudioSampleGroupEntry(grouping_type);
                                break;
                        case 'subt': // for subtitle tracks
                                SubtitleSampleGroupEntry(grouping_type);
                                break;
                        case 'text': // for text tracks
                                TextSampleGroupEntry(grouping_type);
                                break;
                        case 'hint': // for hint tracks
                                HintSampleGroupEntry(grouping_type);
                                break;
                }
        }
}
```

*In 8.9.3.3 add*

`default_sample_description_index`: specifies the index of the sample group description entry which applies to all samples in the track for which no sample to group mapping is provided through a SampleToGroup box. The default value of this field is zero (indicating that the samples are mapped to no group of this type).

*In 8.9.3.3 add at the end of the grouping_type semantics:*

If grouping_type_parameter is not defined for a given grouping_type, then there shall be only one occurrence of this box with this grouping_type.

*In 8.11.1.1 delete the sentence*

There should be no conflicts during this process – no duplicate item_id, for example – and if there is, it is unspecified which item takes precedence.

*and add after the paragraph which contained it, the following:*

If, during this merge, there are either (a) meta-data items with the same item_ID or (b) user-data items with the same type, then the following applies:

a) all occurrences of the data (user-data box or meta-data item) must be 'true' for the entire movie including all fragments;

b) the occurrences in higher-numbered movie fragments ('later' occurrences) may be more accurate or 'preferred';

c) in particular, data in an empty initial movie atom may be only estimates or 'not to exceed' values, and data in a final otherwise empty movie fragment may be the 'final' or most accurate values.

*At the end of 8.11.12.1 add*

An item reference of type `'font'` may be used to indicate that an item uses fonts carried/defined in the referenced item.

*Add a new section in Annex E*

**The 'iso8' brand**

The brand ‘iso8’ requires support for all features of the ‘iso7’ brand.

Support for the following boxes is required under this brand:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | sthd | | | | subtitle media header, overall information (subtitle track only) |

Support for the following is required under this brand:

— Support for ‘meta’ box in movie fragments

— Support for one or more ‘subs’ box per track

*Replace E.10 in Annex E*

**The ‘iso7’ brand**

The brand ‘iso7’ requires support for all features of the ‘iso6’ brand.

Support for the following boxes is required under this brand:

— Support for 32-bit `item_ID` and `item_count` values

— Recognizing incomplete tracks.

*With*

The brand ‘iso7’ requires support for all features of the ‘iso6’ brand.

Support for the following boxes is required under this brand:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | trep | | | | | | track extension properties |
| | | | assp | | | | | alternative startup sequence properties |

Support for the following is required under this brand:

— Support for 32-bit `item_ID` and `item_count` values in ‘meta’ box

— Recognizing incomplete tracks.

— Support for ‘meta’ box in movie fragments is not required

— Support for only ‘subs’ box per track is required here

*Add to the end of each E.4, E.6, E.7, E.8, E.9 in Annex E*

**The ‘iso2’, ‘iso3’, ‘iso4’, ‘iso5’, ‘iso6’ brand**

— Support for only 16-bit `item_ID` and `item_count` values in ‘meta’ box is required here; 32-bit `item_ID` and `item_count` values in ‘meta’ box is not required

— Support for ‘meta’ box in movie fragments is not required

— Support for only ‘subs’ box per track is required here

*Add the following Annex as Annex K*



 

# Annex K

(informative)

# Segment Index Examples

## K.1 Introduction

This annex gives some examples of the use of the segment index box, and what values are inserted in it when it is used in various different 'styles' or configurations.

In the following examples, the size of i-th 'sidx' box is defined as $S_{i,index}$, the size of i-th subsegment, e.g. i-th 'moof' and 'mdat' boxes, is defined as $S_{i,media}$, the duration of i-th subsegment is defined as $D_i$, the number of the last subsegment is defined as $N$, and the duration of the segment is defined as $D_{segment}$.

## K.2 Examples

### K.2.1 Simple one-level indexing

This example shows a simple segment index (Figure K.1). All entries of the top level sidx point to media content (segments comprising one or more movie fragments), i.e. `reference_type` is equal to 0. The value of `referenced_size` and `subsegment_duration` of each entry are calculated as Table K.1.

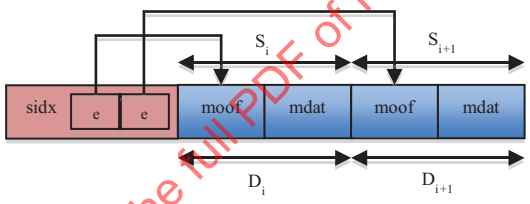


**Figure K.1 — Simple Segment Index**

**Table K.1 — Simple Segment Index**

| sidx | entries | `referenced_size` | `subsegment_duration` |
|---|---|---|---|
| | $e_0$ | $S_i$ | $D_i$ |
| | $e_1$ | $S_{i+1}$ | $D_{i+1}$ |

### K.2.2 Hierarchical

This example shows hierarchical segment index (Figure K.2). All entries of the top level sidx point to another 'sidx' box, i.e. `reference_type` is equal to 1, and all entries of the second level sidx point to media content, i.e. reference_type is equal to 0. The value of `referenced_size` and `subsegment_duration` of each entry are calculated as Table K.2.