

ISO/IEC JTC 1

Secretariat: **ANSI**

Voting begins on:  
**2011-10-07**

Voting terminates on:  
**2011-12-07**

---

---

## Information technology — JPEG 2000 image coding system —

### Part 12: ISO base media file format

AMENDMENT 3: DASH support and RTP  
reception hint track processing

iTeh STANDARD PREVIEW  
(standards.iteh.ai)

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

*ISO/IEC 15444-12:2008/FDAmd 3*

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

*6576481b1ac/iso-iec-15444-12-2008-fdamd-3*

*AMENDEMENT 3: Processus de suivi de l'encre de réception RTP et  
du support DASH*

Please see the administrative notes on page iii

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.



Reference number  
ISO/IEC 15444-12:2008/FDAM 3:2011(E)

## iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 15444-12:2008/FDAmd 3](https://standards.iteh.ai/catalog/standards/sist/adc17fcf-1b1b-4d89-b1af-6576481b11ac/iso-iec-15444-12-2008-fdamd-3)

<https://standards.iteh.ai/catalog/standards/sist/adc17fcf-1b1b-4d89-b1af-6576481b11ac/iso-iec-15444-12-2008-fdamd-3>

### Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

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](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

In accordance with the provisions of Council Resolution 21/1986, this document is **circulated in the English language only**.

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

ISO/IEC 15444-12:2008/FDAmd 3

<https://standards.iteh.ai/catalog/standards/sist/adc17fcf-1b1b-4d89-b1af-6576481b11ac/iso-iec-15444-12-2008-fdamd-3>

## 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 3 to ISO/IEC 15444-12:2008 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ITOH STANDARD PREVIEW

(standards.iteh.ai)

[ISO/IEC 15444-12:2008/FDAmd 3](https://standards.iteh.ai/catalog/standards/sist/adc17fcf-1b1b-4d89-b1af-6576481b11ac/iso-iec-15444-12-2008-fdamd-3)

<https://standards.iteh.ai/catalog/standards/sist/adc17fcf-1b1b-4d89-b1af-6576481b11ac/iso-iec-15444-12-2008-fdamd-3>

# Information technology — JPEG 2000 image coding system

## Part 12: ISO base media file format

### AMENDMENT 3: DASH support and RTP reception hint track processing

Add the following normative references to Clause 2:

IETF RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*, SCHULZRINNE, H. et al., July 2003

IETF RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*, MILLS, D., et al., June 2010

Add the following terms to Clause 3 in alphabetical sequence, renumbering subclauses as appropriate:

#### 3.x.x segment

portion of an ISO base media file format file, consisting of either (a) a movie box, with its associated media data (if any) and other associated boxes or (b) one or more movie fragment boxes, with their associated media data, and other associated boxes

#### 3.x.x subsegment

time interval of a segment formed from movie fragment boxes, that is also a valid segment

#### 3.x.x leaf subsegment

subsegment that does not contain any indexing information that would enable its further division into subsegments;

At the end of the first paragraph in 6.2.2, add the following:

Values for counters, offsets, times, durations etc. in this format do not 'wrap' to 0 when the maximum value that can be stored in their field is reached; appropriately large fields must be used for all values.

In 6.2.3, add the following to Table 1:

after the first line documenting the 'subs' box, add:

				saiz	8.7.8	sample auxiliary information sizes
				saio	8.7.9	sample auxiliary information offsets

and also, after the line 'trex':

		leva			8.8.13	level assignment
--	--	------	--	--	--------	------------------

and also, before the line 'mfra':

		saiz			8.7.8	sample auxiliary information sizes
		saio			8.7.9	sample auxiliary information offsets
		tfdt			8.8.12	track fragment decode time

at the end of the table, add:

styp						8.16.2	segment type
sidx						8.16.3	segment index
ssix						8.16.4	subsegment index
prft						8.16.5	producer reference time

Remove the line permitting an sdtp box in a track fragment (indented under 'traf', after 'trun'):

		sdtp				8.6.4	independent and disposable samples
--	--	------	--	--	--	-------	------------------------------------

In 8.6.1.4.1, add at the end:

When the Composition to Decode Box is included in the Sample Table Box, it documents the composition and decoding time relations of the samples in the Movie Box only, not including any subsequent movie fragments.

In 8.6.1.4.3, change the definition of compositionEndTime as follows:

compositionEndTime: the composition time plus the composition duration, of the sample with the largest computed composition time (CTS) in the media of this track; if this field takes the value 0, the composition end time is unknown.

In 8.6.4.1 change the header to:

Box Types: 'sdtp'  
 Container: Sample Table Box ('stbl')  
 Mandatory: No  
 Quantity: Zero or one

iTech STANDARD PREVIEW  
 (standards.iteh.ai)

add this paragraph before the paragraph beginning "The size of the table...":

<https://standards.iteh.ai/catalog/standards/sist/adc17fcf-1b1b-4d89-b1af-6576481b113e/iso-iec-15444-12-2008/fdam-3>

For tracks with a handler\_type that is not 'vide', 'soun', 'hint' or 'auxv', if another sample with sample\_depends\_on=2 or another sample tagged as a "Sync Sample" has already been processed and unless specified otherwise, a sample tagged with sample\_depends\_on=2, and sample\_has\_redundancy=1 can be discarded, and its duration added to the duration of the preceding one, to maintain the timing of subsequent samples.

and delete this line from the text:

A sample dependency Box may also occur in the track fragment Box.

In 8.6.6.1, insert at the end:

A non-empty edit may insert a portion of the media timeline that is not present in the initial movie, and is present only in subsequent movie fragments. The segment\_duration of this edit may be zero, whereupon the edit provides the offset from media composition time to movie presentation time, for the movie and subsequent movie fragments. It is recommended that such an edit be used to establish a presentation time of 0 for the first presented sample, when composition offsets are used.

For example, if the composition time of the first composed frame is 20, then the edit that maps the media time from 20 onwards to movie time 0 onwards, would read:

Entry-count = 1  
 Segment-duration = 0  
 Media-Time = 20  
 Media-Rate = 1

At the end of 8.7.2.1, add this paragraph:

When a file that has data entries with the flag set indicating that the media data is in the same file, is split into segments for transport, the value of this flag does not change, as the file is (logically) reassembled after the transport operation.

Insert the following Subclauses into 8.7, numbered consecutively from the existing Subclauses:

## 8.7.8 Sample Auxiliary Information Sizes Box

### 8.7.8.1 Definition

Box Type: 'sai<sub>z</sub>'  
 Container: Sample Table Box ('stbl') or Track Fragment Box ('traf')  
 Mandatory: No  
 Quantity: Zero or More

Per-sample sample auxiliary information may be stored anywhere in the same file as the sample data itself; for self-contained media files, this is typically in a MediaData box or a box from a derived specification. It is stored either (a) in multiple chunks, with the number of samples per chunk, as well as the number of chunks, matching the chunking of the primary sample data or (b) in a single chunk for all the samples in a movie sample table (or a movie fragment). The Sample Auxiliary Information for all samples contained within a single chunk (or track run) is stored contiguously (similarly to sample data).

Sample Auxiliary Information, when present, is always stored in the same file as the samples to which it relates as they share the same data reference ('dref') structure. However, this data may be located anywhere within this file, using auxiliary information offsets ('sai<sub>o</sub>') to indicate the location of the data.

Whether sample auxiliary information is permitted or required may be specified by the brands or the coding format in use. The format of the sample auxiliary information is determined by `aux_info_type`. If `aux_info_type` and `aux_info_type_parameter` are omitted then the implied value of `aux_info_type` is either (a) in the case of transformed content, such as protected content, the `scheme_type` included in the Protection Scheme Information box or otherwise (b) the sample entry type. The default value of the `aux_info_type_parameter` is 0. Some values of `aux_info_type` may be restricted to be used only with particular track types. A track may have multiple streams of sample auxiliary information of different types. The types are registered at the registration authority.

While `aux_info_type` determines the format of the auxiliary information, several streams of auxiliary information having the same format may be used when their value of `aux_info_type_parameter` differs. The semantics of `aux_info_type_parameter` for a particular `aux_info_type` value must be specified along with specifying the semantics of the particular `aux_info_type` value and the implied auxiliary information format.

This box provides the size of the auxiliary information for each sample. For each instance of this box, there must be a matching `SampleAuxiliaryInformationOffsetsBox` with the same values of `aux_info_type` and `aux_info_type_parameter`, providing the offset information for this auxiliary information.

NOTE For discussions on the use of sample auxiliary information versus other mechanisms, see Annex C.8.

### 8.7.8.2 Syntax

```
aligned(8) class SampleAuxiliaryInformationSizesBox
  extends FullBox('saiz', version = 0, flags)
{
  if (flags & 1) {
    unsigned int(32) aux_info_type;
    unsigned int(32) aux_info_type_parameter;
  }
  unsigned int(8) default_sample_info_size;
  unsigned int(32) sample_count;
  if (default_sample_info_size == 0)
  {
    unsigned int(8) sample_info_size[ sample_count ];
  }
}
```

### 8.7.8.3 Semantics

`aux_info_type` is an integer that identifies the type of the sample auxiliary information. At most one occurrence of this box with the same values for `aux_info_type` and `aux_info_type_parameter` shall exist in the containing box.

`aux_info_type_parameter` identifies the “stream” of auxiliary information having the same value of `aux_info_type` and associated to the same track. The semantics of `aux_info_type_parameter` are determined by the value of `aux_info_type`.

`default_sample_info_size` is an integer specifying the sample auxiliary information size for the case where all the indicated samples have the same sample auxiliary information size. If the size varies then this field shall be zero.

`sample_count` is an integer that gives the number of samples for which a size is defined. For a Sample Auxiliary Information Sizes box appearing in the Sample Table Box this must be the same as, or less than, the `sample_count` within the Sample Size Box or Compact Sample Size Box. For a Sample Auxiliary Information Sizes box appearing in a Track Fragment box this must be the same as, or less than, the sum of the `sample_count` entries within the Track Fragment Run boxes of the Track Fragment. If this is less than the number of samples, then auxiliary information is supplied for the initial samples, and the remaining samples have no associated auxiliary information.

`sample_info_size` gives the size of the sample auxiliary information in bytes. This may be zero to indicate samples with no associated auxiliary information.

## 8.7.9 Sample Auxiliary Information Offsets Box

### 8.7.9.1 Definition

Box Type: 'saio'  
 Container: Sample Table Box ('stbl') or Track Fragment Box ('traf')  
 Mandatory: No  
 Quantity: Zero or More

For an introduction to sample auxiliary information, see the definition of the Sample Auxiliary Information Size Box.

This box provides the position information for the sample auxiliary information, in a way similar to the chunk offsets for sample data.



### 8.7.9.2 Syntax

```
aligned(8) class SampleAuxiliaryInformationOffsetsBox
  extends FullBox('saio', version, flags)
{
  if (flags & 1) {
    unsigned int(32) aux_info_type;
    unsigned int(32) aux_info_type_parameter;
  }
  unsigned int(32) entry_count;
  if ( version == 0 )
  {
    unsigned int(32) offset[ entry_count ];
  }
  else
  {
    unsigned int(64) offset[ entry_count ];
  }
}
```

### 8.7.9.3 Semantics

`aux_info_type` and `aux_info_type_parameter` are defined as in the `SampleAuxiliaryInformationSizesBox`

`entry_count` gives the number of entries in the following table. For a Sample Auxiliary Information Offsets box appearing in a Sample Table Box this must be equal to one or to the value of the `entry_count` field in the Chunk Offset Box or Chunk Large Offset Box. For a Sample Auxiliary Information Offsets Box appearing in a Track Fragment box, this must be equal to one or to the number of Track Fragment Run boxes in the Track Fragment Box.

`offset` gives the position in the file of the Sample Auxiliary Information for each Chunk or Track Fragment Run. If `entry_count` is one, then the Sample Auxiliary Information for all Chunks or Runs is contiguous in the file in chunk or run order. When in the Sample Table Box, the offsets are absolute. In a track fragment box, this value is relative to the base offset established by the track fragment header box (<http://dx.doi.org/10.25358/iso-iec-15444-12-2008-fdamd-3>) in the same track fragment (see 8.8.14).

*At the end of 8.8.4.1, add this note:*

NOTE There is no requirement that any particular movie fragment extend all tracks present in the movie header, and there is no restriction on the location of the media data referred to by the movie fragments. However, derived specifications may make such restrictions.

*In 8.8.5.1, after:*

The movie fragment header contains a sequence number, as a safety check. The sequence number usually starts at 1 and must increase for each movie fragment in the file, in the order in which they occur. This allows readers to verify integrity of the sequence; it is an error to construct a file where the fragments are out of sequence.

*insert:*

NOTE There is no requirement that the sequence numbers be consecutive, only that the value in a given movie fragment be greater than in any preceding movie fragment.

*At the end of 8.8.8.1, add:*

The composition offset values in the composition time-to-sample box and in the track run box may be signed or unsigned. The recommendations given in the composition time-to-sample box concerning the use of signed composition offsets also apply here.

Replace the contents of 8.8.8.2 with the following:

```
aligned(8) class TrackRunBox
    extends FullBox('trun', version, tr_flags) {
    unsigned int(32) sample_count;
    // the following are optional fields
    signed int(32) data_offset;
    unsigned int(32) first_sample_flags;
    // all fields in the following array are optional
    {
        unsigned int(32) sample_duration;
        unsigned int(32) sample_size;
        unsigned int(32) sample_flags;
        if (version == 0)
            { unsigned int(32) sample_composition_time_offset; }
        else
            { signed int(32) sample_composition_time_offset; }
    } [ sample_count ]
}
```

In 8.8, add the following:

**8.8.12 Track fragment decode time**

**8.8.12.1 Definition**

iTeh STANDARD PREVIEW

Box Type: 'tfdt'  
 Container: Track Fragment box ('traf')  
 Mandatory: No  
 Quantity: Zero or one

<https://standards.iteh.ai/catalog/standards/sist/adc176cf-1b1b-4d89-b1af-6576481b11ac/iso-iec-15444-12-2008-fdamd-3>

The Track Fragment Base Media Decode Time Box provides the absolute decode time, measured on the media timeline, of the first sample in decode order in the track fragment. This can be useful, for example, when performing random access in a file; it is not necessary to sum the sample durations of all preceding samples in previous fragments to find this value (where the sample durations are the deltas in the Decoding Time to Sample Box and the sample\_durations in the preceding track runs).

The Track Fragment Base Media Decode Time Box, if present, shall be positioned after the Track Fragment Header Box and before the first Track Fragment Run box.

NOTE The decode timeline is a media timeline, established before any explicit or implied mapping of media time to presentation time, for example by an edit list or similar structure.

**8.8.12.2 Syntax**

```
aligned(8) class TrackFragmentBaseMediaDecodeTimeBox
    extends FullBox('tfdt', version, 0) {
    if (version==1) {
        unsigned int(64) baseMediaDecodeTime;
    } else { // version==0
        unsigned int(32) baseMediaDecodeTime;
    }
}
```

**8.8.12.3 Semantics**

version is an integer that specifies the version of this box (0 or 1 in this specification).  
 baseMediaDecodeTime is an integer equal to the sum of the decode durations of all earlier samples in the media, expressed in the media's timescale. It does not include the samples added in the enclosing track fragment.

### 8.8.13 Level Assignment Box

#### 8.8.13.1 Definition

Box Type: `leva`  
 Container: Movie Extends Box (`mvex`)  
 Mandatory: No  
 Quantity: Zero or one

Levels specify subsets of the file. Samples mapped to level  $n$  may depend on any samples of levels  $m$ , where  $m \leq n$ , and shall not depend on any samples of levels  $p$ , where  $p > n$ . For example, levels can be specified according to temporal level (e.g., `temporal_id` of SVC or MVC).

Levels cannot be specified for the initial movie. When the Level Assignment box is present, it applies to all movie fragments subsequent to the initial movie.

For the context of the Level Assignment box, a fraction is defined to consist of one or more Movie Fragment boxes and the associated Media Data boxes, possibly including only an initial part of the last Media Data Box. Within a fraction, data for each level shall appear contiguously. Data for levels within a fraction shall appear in increasing order of level value. All data in a fraction shall be assigned to levels.

NOTE In the context of DASH (ISO/IEC 23009-1), each subsegment indexed within a Subsegment Index box is a fraction.

The Level Assignment box provides a mapping from features, such as scalability layers, to levels. A feature can be specified through a track, a sub-track within a track, or a sample grouping of a track.

When `padding_flag` is equal to 1, this indicates that a conforming fraction can be formed by concatenating any positive integer number of levels within a fraction and padding the last Media Data box by zero bytes up to the full size that is indicated in the header of the last Media Data box. For example, `padding_flag` can be set equal to 1 when the following conditions are true:

- Each fraction contains two or more AVC, SVC, or MVC [ISO/IEC 14496-15] tracks of the same video bitstream.
- The samples for each track of a fraction are contiguous and in decoding order in a Media Data box.
- The samples of the first AVC, SVC, or MVC level contain extractor NAL units for including the video coding NAL units from the other levels of the same fraction.

#### 8.8.13.2 Syntax

```
aligned(8) class LevelAssignmentBox extends FullBox('leva', 0, 0)
{
    unsigned int(8)    level_count;
    for (j=1; j <= level_count; j++) {
        unsigned int(32) track_id;
        unsigned int(1)  padding_flag;
        unsigned int(7)  assignment_type;
        if (assignment_type == 0)
            unsigned int(32) grouping_type;
        else if (assignment_type == 1) {
            unsigned int(32) grouping_type;
            unsigned int(32) grouping_type_parameter;
        }
        else if (assignment_type == 2) {} // no further syntax elements needed
        else if (assignment_type == 3) {} // no further syntax elements needed
        else if (assignment_type == 4)
            unsigned int(32) sub_track_id;
        // other assignment_type values are reserved
    }
}
```

### 8.8.13.3 Semantics

`level_count` specifies the number of levels each fraction is grouped into. `level_count` shall be greater than or equal to 2.

`track_id` for loop entry `j` specifies the track identifier of the track assigned to level `j`.

`padding_flag` equal to 1 indicates that a conforming fraction can be formed by concatenating any positive integer number of levels within a fraction and padding the last Media Data box by zero bytes up to the full size that is indicated in the header of the last Media Data box. The semantics of `padding_flag` equal to 0 are that this is not assured.

`assignment_type` indicates the mechanism used to specify the assignment to a level.

`assignment_type` values greater than 4 are reserved, while the semantics for the other values are specified as follows. The sequence of `assignment_types` is restricted to be a set of zero or more of type 2 or 3, followed by zero or more of exactly one type.

- 0: sample groups are used to specify levels, i.e., samples mapped to different sample group description indexes of a particular sample grouping lie in different levels within the identified track; other tracks are not affected and must have all their data in precisely one level;
- 1: as for `assignment_type` 0 except assignment is by a parameterized sample group;
- 2, 3: level assignment is by track (see the Subsegment Index Box for the difference in processing of these levels)
- 4: the respective level contains the samples for a sub-track. The sub-tracks are specified through the Sub Track box; other tracks are not affected and must have all their data in precisely one level;

`grouping_type` and `grouping_type_parameter`, if present, specify the sample grouping used to map sample group description entries in the Sample Group Description box to levels. Level `n` contains the samples that are mapped to the sample group description entry having index `n` in the Sample Group Description box having the same values of `grouping_type` and `grouping_type_parameter`, if present, as those provided in this box.

`sub_track_id` specifies that the sub-track identified by `sub_track_id` within loop entry `j` is mapped to level `j`.

### 8.8.14 Sample Auxiliary Information in Movie Fragments

When sample auxiliary information (8.7.8 and 8.7.9) is present in the Movie Fragment box, the offsets in the Sample Auxiliary Information Offsets Box are treated the same as the `data_offset` in the Track Fragment Run box, that is, they are relative to any base data offset established for that track fragment. If movie fragment relative addressing is used (no base data offset is provided in the track fragment header) and auxiliary information is present, then the `default_base_is_moof` flag must also be set in the flags of that track fragment header.

If only one offset is provided, then the Sample Auxiliary Information for all the track runs in the fragment is stored contiguously, otherwise exactly one offset must be provided for each track run.

If the field `default_sample_info_size` is non-zero in one of these boxes, then the size of the auxiliary information is constant for the identified samples.

In addition, if:

- this box is present in the movie box,
- and `default_sample_info_size` is non-zero in the box in the movie box,
- and the sample auxiliary information sizes box is absent in a movie fragment,

then the auxiliary information has this same constant size for every sample in the movie fragment also; it is then not necessary to repeat the box in the movie fragment.

*In 8.9.3.1, change header as follows:*

Box Type: `sgpd`  
 Container: Sample Table Box (`stbl`) or Track Fragment Box (`traf`)  
 Mandatory: No  
 Quantity: Zero or more, with one for each Sample to Group Box.

*Add to the end of 8.9.4:*

Zero or more SampleGroupDefinition boxes may also be present in a Track Fragment Box. These definitions are additional to the definitions provided in the Sample Table of the track in the Movie Box. Group definitions within a movie fragment can also be referenced and used from within that same movie fragment.

Within the SampleToGroup box in that movie fragment, the group description indexes for groups defined within the same fragment start at 0x10001, i.e. the index value 1, with the value 1 in the top 16 bits. This means there must be fewer than 65536 group definitions for this track and grouping type in the sample table in the Movie Box.

When changing the size of movie fragments, or removing them, these fragment-local group definitions will need to be merged into the definitions in the movie box, or into the new movie fragments, and the index numbers in the SampleToGroup box(es) adjusted accordingly. It is recommended that, in this process, identical (and hence duplicate) definitions not be made in any SampleGroupDescription box, but that duplicates be merged and the indexes adjusted accordingly.

*In 8.12.1.1, change the box header:*

Box Types: `sinf`  
 Container: Protected Sample Entry, or Item Protection Box (`ipro`)  
 Mandatory: Yes  
 Quantity: ~~Exactly one~~ One or More

ISO/IEC 15444-12:2008/FDAMd 3  
<https://standards.iteh.ai/catalog/standards/sist/adc17fcf-1b1b-4d89-b1af-6576481b11ac/iso-iec-15444-12-2008-fdamd-3>

*And add the following paragraph:*

At least one protection scheme information box must occur in a protected sample entry. When more than one occurs, they are equivalent, alternative, descriptions of the same protection. Readers should choose one to process.

*Following 8.15, add:*

## 8.16 Segments

### 8.16.1 Introduction

Media presentations may be divided into segments for delivery, for example, it is possible (e.g. in HTTP streaming) to form files that contain a segment – or concatenated segments – which would not necessarily form ISO base media file format compliant files (e.g. they do not contain a movie box).

This Subclause defines specific boxes that may be used in such segments.

### 8.16.2 Segment Type Box

Box Type: `styp`  
 Container: File  
 Mandatory: No  
 Quantity: Zero or more