
**Information technology — Coding of
audio-visual objects —**

**Part 12:
ISO base media file format**

**AMENDMENT 2: Hint track format for
ALC/LCT and FLUTE transmission and
multiple meta box support**

Technologies de l'information — Codage des objets audiovisuels —

Partie 12: Format ISO de base pour les fichiers médias

*AMENDEMENT 2: Format de piste optimisé pour transmission
ALC/LCT et FLUTE et support de boîte "meta" multiple*

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from 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
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 2 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*.

It provides an update for the ISO base media file format by specifying server extensions for file delivery over ALC/LCT and FLUTE, instructions for streaming servers and support for multiple meta boxes.

Information technology — Coding of audio-visual objects —

Part 12:

ISO base media file format

AMENDMENT 2: Hint track format for ALC/LCT and FLUTE transmission and multiple meta box support

Replace Clause 3 with the following:

3 Terms, definitions and abbreviated terms

For the purposes of this document, the following terms and definitions and abbreviated terms apply.

3.1 Terms and definitions

3.1.1

Box

object-oriented building block defined by a unique type identifier and length (called 'atom' in some specifications, including the first definition of MP4)

3.1.2

Chunk

contiguous set of samples for one track

3.1.3

Container Box

box whose sole purpose is to contain and group a set of related boxes

3.1.4

Hint Track

special track which does not contain media data. Instead it contains instructions for packaging one or more tracks into a streaming channel

3.1.5

Hinters

tool that is run on a file containing only media, to add one or more hint tracks to the file and so facilitate streaming

3.1.6

Movie Box

container box whose sub-boxes define the metadata for a presentation ('moov')

3.1.7

Media Data Box

container box which can hold the actual media data for a presentation ('mdat')

3.1.8

ISO Base Media File

name of the file format described in this specification

3.1.9

Presentation

one or more motion sequences (q.v.), possibly combined with audio

3.1.10

Sample

In non-hint tracks, a sample is an individual frame of video, a series of video frames in decoding order, or a compressed section of audio in decoding order. In hint tracks, a sample defines the formation of one or more streaming packets. No two samples within a track may share the same time-stamp.

3.1.11

Sample Description

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

3.1.12

Sample Table

packed directory for the timing and physical layout of the samples in a track

3.1.13

Track

Collection of related samples (q.v.) in an ISO base media file. For media data, a track corresponds to a sequence of images or sampled audio. For hint tracks, a track corresponds to a streaming channel.

3.2 Abbreviated terms

ALC	Asynchronous Layered Coding
FD	File Delivery
FDT	File Delivery Table
FEC	Forward Error Correction
FLUTE	File Delivery over Unidirectional Transport
IANA	Internet Assigned Numbers Authority
LCT	Layered Coding Transport
MBMS	Multimedia Broadcast/Multicast Service

In 4.2 Object Structure, after "The fields in the objects are stored with the most significant byte first, commonly known as network byte order or big-endian format.", insert the following:

When fields smaller than a byte are defined, or fields span a byte boundary, the bits are assigned from the most significant bits in each byte to the least significant. For example, a field of two bits followed by a field of six bits has the two bits in the high order bits of the byte.

In 4.3.1, add the following at the end of the subclause:

All file format brands defined in this specification are included in Annex E with a summary of which features they require.

In 6.2.3, add the following entries at the end of Table 1 (correctly cross-referenced):

	fiin					8.46.2	file delivery item information
		paen				8.46.2	partition entry
			fpar			8.46.3	file partition
			fecr			8.46.4	FEC reservoir
		segr				8.46.5	file delivery session group
		gitn				8.46.6	group id to name
		tsei				8.48.2	track selection
meco						8.44.9	additional metadata container
	mere					8.44.10	metabox relation

In 8.6.3, add the following *reference_type* to the end of the list:

- ‘hind’ this track depends on the referenced hint track, i.e., it should only be used if the referenced hint track is used.

In 8.9.3, replace:

handler_type when present in a meta box, contains an appropriate value to indicate the format of the meta box contents

with:

handler_type when present in a meta box, contains an appropriate value to indicate the format of the meta box contents. The value ‘null’ can be used in the primary meta box to indicate that it is merely being used to hold resources.

In 8.44.1.1, replace:

“Box type: ... Zero or one”, i.e. the first four lines at the beginning,

with:

Box Type: ‘meta’

Container: File, Movie Box (‘moov’), Track Box (‘trak’), or Additional Metadata Container Box (‘meco’)

Mandatory: No

Quantity: Zero or one (in File, ‘moov’, and ‘trak’), One or more (in ‘meco’)

In 8.44.1.1 Definition, replace:

At most one meta box may occur at each of the file level, movie level, or track level.

with:

At most one meta box may occur at each of the file level, movie level, or track level, unless they are contained in an additional metadata container box (‘meco’).

In 8.44.6.1, insert the following text at the end of the subclause:

Two versions of the item info entry are defined. Version 1 includes additional information to version 0 as specified by an extension type. For instance, it shall be used with extension type 'fdel' for items that are referenced by the file partition box ('fpar'), which is defined for source file partitionings and applies to file delivery transmissions.

If no extension is desired, the box may terminate without the extension_type field and the extension; if, in addition, content_encoding is not desired, that field also may be absent and the box terminate before it. If an extension is desired without an explicit content_encoding, a single null byte, signifying the empty string, must be supplied for the content_encoding, before the indication of extension_type.

In 8.44.6.2, replace the entire text with the following:

```
aligned(8) class ItemInfoExtension(unsigned int(32) extension_type)
{
}

aligned(8) class FDIItemInfoExtension() extends ItemInfoExtension ('fdel')
{
    string          content_location;
    string          content_MD5;
    unsigned int(64) content_length;
    unsigned int(64) transfer_length;
    unsigned int(8)  entry_count;
    for (i=1; i <= entry_count; i++)
        unsigned int(32) group_id;
}

aligned(8) class ItemInfoEntry
    extends FullBox('infe', version, 0) {
    if ((version == 0) || (version == 1)) {
        unsigned int(16) item_ID;
        unsigned int(16) item_protection_index;
        string          item_name;
        string          content_type;
        string          content_encoding; //optional
    }
    if (version == 1) {
        unsigned int(32) extension_type; //optional
        ItemInfoExtension(extension_type); //optional
    }
}

aligned(8) class ItemInfoBox
    extends FullBox('iinf', version = 0, 0) {
    unsigned int(16) entry_count;
    ItemInfoEntry[ entry_count ] item_infos;
}
```

In 8.44.6.3, replace the entire text with the following:

item_id contains either 0 for the primary resource (e.g., the XML contained in an 'xml' box) or the ID of the item for which the following information is defined.

item_protection_index contains either 0 for an unprotected item, or the one-based index into the item protection box defining the protection applied to this item (the first box in the item protection box has the index 1).

item_name is a null-terminated string in UTF-8 characters containing a symbolic name of the item (source file for file delivery transmissions).

`content_type` is a null-terminated string in UTF-8 characters with the MIME type of the item. If the item is content encoded (see below), then the content type refers to the item after content decoding.

`content_encoding` is an optional null-terminated string in UTF-8 characters used to indicate that the binary file is encoded and needs to be decoded before interpreted. The values are as defined for Content-Encoding for HTTP/1.1. Some possible values are "gzip", "compress" and "deflate". An empty string indicates no content encoding. Note that the item is stored after the content encoding has been applied.

`extension_type` is a printable four-character code that identifies the extension fields of version 1 w.r.t. version 0 of the Item information entry.

`content_location` is a null-terminated string in UTF-8 characters containing the URI of the file as defined in HTTP/1.1 (RFC 2616).

`content_MD5` is a null-terminated string in UTF-8 characters containing an MD5 digest of the file. See HTTP/1.1 (RFC 2616) and RFC 1864.

`content_length` gives the total length (in bytes) of the (un-encoded) file.

`transfer_length` gives the total length (in bytes) of the (encoded) file. Note that transfer length is equal to content length if no content encoding is applied (see above).

`entry_count` provides a count of the number of entries in the following array.

`group_ID` indicates a file group to which the file item (source file) belongs. See 3GPP TS 26.346 for more details on file groups.

Add the following subclauses before 8.45:

8.44.9 Additional Metadata Container Box

8.44.9.1 Definition

Box Type: 'meco'
 Container: File, Movie Box ('moov'), or Track Box ('trak')
 Mandatory: No
 Quantity: Zero or one

The additional metadata container box includes one or more meta boxes. It can be carried at the top level of the file, in the Movie Box ('moov'), or in the Track Box ('trak') and shall only be present if it is accompanied by a meta box in the same container. A meta box that is not contained in the additional metadata container box is the preferred (primary) meta box. Meta boxes in the additional metadata container box complement or give alternative metadata information. The usage of multiple meta boxes may be desirable when, e.g., a single handler is not capable of processing all metadata. All meta boxes at a certain level, including the preferred one and those contained in the additional metadata container box, must have different handler types.

A meta box contained in an additional metadata container box shall contain a primary Item box or the primary data box required by the handler (e.g., an XML Box). It shall not include boxes or syntax elements concerning items other than the primary item indicated by the present primary item box or XML box. URLs in a meta box contained in an additional metadata container box are relative to the context of the preferred meta box.

8.44.9.2 Syntax

```
aligned(8) class AdditionalMetadataContainerBox extends Box('meco') {
}
```