

---

---

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

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

*Technologies de l'information — Codage des objets audiovisuels —  
Partie 12: Format ISO de base pour les fichiers médias*

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

ISO/IEC 14496-12:2004

<https://standards.iteh.ai/catalog/standards/sist/91f4d619-013b-4900-977a-349bcce5bb11/iso-iec-14496-12-2004>

**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.

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

ISO/IEC 14496-12:2004

<https://standards.iteh.ai/catalog/standards/sist/91f4d619-013b-4900-977a-349bcce5bb11/iso-iec-14496-12-2004>

© ISO/IEC 2004

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

Published in Switzerland

# Contents

Page

Foreword .....	v
Introduction .....	vi
1 Scope .....	1
2 Normative references .....	1
3 Terms and definitions .....	1
4 Object-structured File Organization .....	2
4.1 File Structure .....	2
4.2 Object Structure .....	3
4.3 File Type Box .....	4
5 Design Considerations .....	5
5.1 Usage .....	5
5.1.1 Interchange .....	5
5.1.2 Content Creation .....	5
5.1.3 Preparation for streaming .....	6
5.1.4 Local presentation .....	6
5.1.5 Streamed presentation .....	6
5.2 Design principles .....	7
6 ISO Base Media File organization .....	7
6.1 Presentation structure .....	7
6.1.1 File Structure .....	7
6.1.2 Object Structure .....	8
6.1.3 Meta Data and Media Data .....	8
6.1.4 Track Identifiers .....	8
6.2 Metadata Structure (Objects) .....	8
6.2.1 Box .....	8
6.2.2 Data Types and fields .....	8
6.2.3 Box Order .....	9
7 Streaming Support .....	12
7.1 Handling of Streaming Protocols .....	12
7.2 Protocol 'hint' tracks .....	12
7.3 Hint Track Format .....	13
8 Box Definitions .....	13
8.1 Movie Box .....	13
8.2 Media Data Box .....	14
8.3 Movie Header Box .....	14
8.4 Track Box .....	15
8.5 Track Header Box .....	16
8.6 Track Reference Box .....	17
8.7 Media Box .....	18
8.8 Media Header Box .....	18
8.9 Handler Reference Box .....	19
8.10 Media Information Box .....	20
8.11 Media Information Header Boxes .....	20
8.11.2 Video Media Header Box .....	20
8.11.3 Sound Media Header Box .....	21
8.11.4 Hint Media Header Box .....	21
8.11.5 Null Media Header Box .....	21
8.12 Data Information Box .....	22

8.13	Data Reference Box.....	22
8.14	Sample Table Box.....	23
8.15	Time to Sample Boxes .....	23
8.15.2	Decoding Time to Sample Box.....	24
8.15.3	Composition Time to Sample Box .....	25
8.16	Sample Description Box .....	26
8.17	Sample Size Boxes.....	28
8.17.2	Sample Size Box.....	29
8.17.3	Compact Sample Size Box .....	29
8.18	Sample To Chunk Box .....	29
8.19	Chunk Offset Box .....	30
8.20	Sync Sample Box .....	31
8.21	Shadow Sync Sample Box .....	31
8.22	Degradation Priority Box .....	32
8.23	Padding Bits Box.....	33
8.24	Free Space Box.....	33
8.25	Edit Box .....	34
8.26	Edit List Box.....	34
8.27	User Data Box .....	35
8.28	Copyright Box.....	36
8.29	Movie Extends Box.....	36
8.30	Movie Extends Header Box .....	36
8.31	Track Extends Box .....	37
8.32	Movie Fragment Box .....	38
8.33	Movie Fragment Header Box.....	38
8.34	Track Fragment Box.....	38
8.35	Track Fragment Header Box .....	39
8.36	Track Fragment Run Box.....	40
8.37	Movie Fragment Random Access Box.....	41
8.38	Track Fragment Random Access Box.....	41
8.39	Movie Fragment Random Access Offset Box .....	42
9	Extensibility .....	43
9.1	Objects .....	43
9.2	Storage formats .....	43
9.3	Derived File formats .....	44
10	RTP Hint Track Format .....	44
10.1	Introduction.....	44
10.2	Sample Description Format.....	45
10.3	Sample Format.....	45
10.3.1	Packet Entry format.....	46
10.3.2	Constructor format.....	46
10.4	SDP Information .....	48
10.4.1	Movie SDP information .....	48
10.4.2	Track SDP Information.....	48
10.5	Statistical Information.....	48
Annex A	(informative) Overview and introduction .....	50
A.1	Section Overview.....	50
A.2	Core Concepts .....	50
A.3	Physical structure of the media .....	50
A.4	Temporal structure of the media .....	51
A.5	Interleave.....	51
A.6	Composition.....	51
A.7	Random access .....	52
A.8	Fragmented movie files .....	52
Annex B	(informative) Patent statements.....	54
Bibliography	.....	55

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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

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

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

ISO/IEC 14496 consists of the following parts, under the general title *Information technology — Coding of audio-visual objects*:

- Part 1: Systems
- Part 2: Visual
- Part 3: Audio
- Part 4: Conformance testing
- Part 5: Reference software
- Part 6: Delivery Multimedia Integration Framework (DMIF)
- Part 7: Optimized reference software for coding of audio-visual objects
- Part 8: Carriage of ISO/IEC 14496 contents over IP networks
- Part 9: Reference hardware description
- Part 10: Advanced Video Coding
- Part 11: Scene description and application engine
- Part 12: ISO base media file format
- Part 13: Intellectual Property Management and Protection (IPMP) extensions
- Part 14: MP4 file format
- Part 15: Advanced Video Coding file format
- Part 16: Animation Framework eXtension (AFX)

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

<https://standards.iteh.ai/catalog/standards/sist/91f4d619-013b-4900-977a-349bce5bb11/iso-iec-14496-12-2004>

## Introduction

The ISO Base Media File Format is designed to contain timed media information for a presentation in a flexible, extensible format that facilitates interchange, management, editing, and presentation of the media. This presentation may be 'local' to the system containing the presentation, or may be via a network or other stream delivery mechanism.

The file structure is object-oriented; a file can be decomposed into constituent objects very simply, and the structure of the objects inferred directly from their type.

The file format is designed to be independent of any particular network protocol while enabling efficient support for them in general.

The ISO Base Media File Format is a base format for media file formats.

It is intended that the ISO Base Media File Format shall be jointly maintained by WG1 and WG11. Consequently, a subdivision of work created ISO/IEC 15444-12 and ISO/IEC 14496-12 in order to document the ISO Base Media File Format and to facilitate the joint maintenance.

This technically identical text is published as ISO/IEC 14496-12 for MPEG-4, and as ISO/IEC 15444-12 for JPEG 2000, and reference to this specification should be made accordingly. The recommendation is to reference one, for example ISO/IEC 14496-12, and append to the reference a parenthetical comment identifying the other, for example "(technically identical to ISO/IEC 15444-12)".

<https://standards.iteh.ai/catalog/standards/sist/91f4d619-013b-4900-977a-349bcce5bb11/iso-iec-14496-12-2004>

# Information technology — Coding of audio-visual objects —

## Part 12:

## ISO base media file format

### 1 Scope

This International Standard specifies the ISO base media file format, which is a general format forming the basis for a number of other more specific file formats. This format contains the timing, structure, and media information for timed sequences of media data, such as audio/visual presentations.

This part of ISO/IEC 14496 is applicable to MPEG-4, but its technical content is identical to that of ISO/IEC 15444-12, which is applicable to JPEG 2000.

### 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 639-2:1998, *Codes for the representation of names of languages — Part 2: Alpha-3 code*

ISO/IEC 11578:1996, *Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)*

<https://standards.iteh.ai/catalog/standards/sist/91f4d619-013b-4900-977a-349b551b11/iso-iec-14496-12-2004>

ISO/IEC 14496-1:2001, *Information technology — Coding of audio-visual objects — Part 1: Systems*<sup>1)</sup>

ITU-T Rec.T.800 | ISO/IEC 15444-1, *Information technology — JPEG 2000 image coding system: Core coding system*

### 3 Terms and definitions

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

#### 3.1

##### **Box**

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

#### 3.2

##### **Chunk**

A contiguous set of samples for one track.

#### 3.3

##### **Container Box**

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

---

1) Refer, in particular, to Clause 14, Syntactic Description Language (SDL).

### 3.4

#### **Hint Track**

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

### 3.5

#### **Hinter**

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

### 3.6

#### **Movie Box**

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

### 3.7

#### **Media Data Box**

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

### 3.8

#### **ISO Base Media File**

The name of the file format described in this specification.

### 3.9

#### **Presentation**

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

### 3.10

#### **Sample**

In non-hint tracks, a sample is an individual frame of video, a time-contiguous series of video frames, or a time-contiguous compressed section of audio. 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.

### 3.11

#### **Sample Description**

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

### 3.12

#### **Sample Table**

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

### 3.13

#### **Track**

A 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.

## 4 Object-structured File Organization

### 4.1 File Structure

Files are formed as a series of objects, called boxes in this specification. All data is contained in boxes; there is no other data within the file. This includes any initial signature required by the specific file format.

All object-structured files conformant to this section of this specification (all Object-Structured files) shall contain a File Type Box.



## 4.2 Object Structure

An object in this terminology is a box.

Boxes start with a header which gives both size and type. The header permits compact or extended size (32 or 64 bits) and compact or extended types (32 bits or full UUIDs). The standard boxes all use compact types (32-bit) and most boxes will use the compact (32-bit) size. Typically only the Media Data Box(es) need the 64-bit size.

The size is the entire size of the box, including the size and type header, fields, and all contained boxes. This facilitates general parsing of the file.

The definitions of boxes are given in the syntax description language (SDL) defined in MPEG-4 (see reference in clause 2). Comments in the code fragments in this specification indicate informative material.

The fields in the objects are stored with the most significant byte first, commonly known as network byte order or big-endian format.

```
aligned(8) class Box (unsigned int(32) boxtype,
    optional unsigned int(8)[16] extended_type) {
    unsigned int(32) size;
    unsigned int(32) type = boxtype;
    if (size==1) {
        unsigned int(64) largesize;
    } else if (size==0) {
        // box extends to end of file
    }
    if (boxtype=='uuid') {
        unsigned int(8)[16] usertype = extended_type;
    }
}
```

ISO/IEC 14496-12:2004

<https://standards.iteh.ai/catalog/standards/sist/91f4d619-013b-4900-977a-549bce5bb11/iso-iec-14496-12-2004>

The semantics of these two fields are:

**size** is an integer that specifies the number of bytes in this box, including all its fields and contained boxes; if size is 1 then the actual size is in the field **largesize**; if size is 0, then this box is the last one in the file, and its contents extend to the end of the file (normally only used for a Media Data Box)

**type** identifies the box type; standard boxes use a compact type, which is normally four printable characters, to permit ease of identification, and is shown so in the boxes below. User extensions use an extended type; in this case, the type field is set to 'uuid'.

Boxes with an unrecognized type shall be ignored and skipped.

Many objects also contain a version number and flags field:

```
aligned(8) class FullBox(unsigned int(32) boxtype, unsigned int(8) v, bit(24) f)
    extends Box(boxtype) {
    unsigned int(8) version = v;
    bit(24) flags = f;
}
```

The semantics of these two fields are:

**version** is an integer that specifies the version of this format of the box.

**flags** is a map of flags

Boxes with an unrecognized version shall be ignored and skipped.

### 4.3 File Type Box

#### 4.3.1 Definition

**Box Type:** `ftyp`  
**Container:** File  
**Mandatory:** Yes  
**Quantity:** Exactly one

A media-file structured to this part of this specification may be compatible with more than one detailed specification, and it is therefore not always possible to speak of a single 'type' or 'brand' for the file. This means that the utility of the file name extension and mime type are somewhat reduced.

This box must 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 Movie Box, Media Data Box, or Free Space). 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.

The minor version is informative only. It does not appear for compatible-brands, and must not be used to determine the conformance of a file to a standard. It may allow more precise identification of the major specification, for inspection, debugging, or improved decoding.

The type 'isom' (ISO Base Media file) is defined in this section of this specification, as identifying files that conform to the ISO Base Media File Format. More specific identifiers can be used to identify precise versions of specifications providing more detail. This brand should not be used as the major brand; this base file format should be derived into another specification to be used. There is therefore no defined normal file extension, or mime type assigned to this brand nor definition of the minor version when 'isom' is the major brand.

<https://standards.iteh.ai/catalog/standards/sist/91f4d619-013b-4900-977a-349bcce5bb11/iso-iec-14496-12-2004>

Files would normally be externally identified (e.g. with a file extension or mime type) that identifies the 'best use' (major brand), or the brand that the author believes will provide the greatest compatibility.

#### 4.3.2 Syntax

```
aligned(8) class FileTypeBox
{
    extends Box('ftyp') {
        unsigned int(32) major_brand;
        unsigned int(32) minor_version;
        unsigned int(32) compatible_brands[];    // to end of the box
    }
}
```

#### 4.3.3 Semantics

This box identifies the specifications to which this file complies.

Each brand is a printable four-character code, registered with ISO, that identifies a precise specification. Only one brand is defined here: 'isom' (ISO Base Media File), identifies files structurally conformant to this media-independent part of this specification.

`major_brand` – is a brand identifier  
`minor_version` – is an informative integer for the minor version of the major brand  
`compatible_brands` – is a list, to the end of the box, of brands

## 5 Design Considerations

### 5.1 Usage

The file format is intended to serve as a basis for a number of operations. In these various roles, it may be used in different ways, and different aspects of the overall design exercised.

#### 5.1.1 Interchange

When used as an interchange format, the files would normally be self-contained (not referencing media in other files), contain only the media data actually used in the presentation, and not contain any information related to streaming. This will result in a small, protocol-independent, self-contained file, which contains the core media data and the information needed to operate on it.

The following diagram gives an example of a simple interchange file, containing two streams.

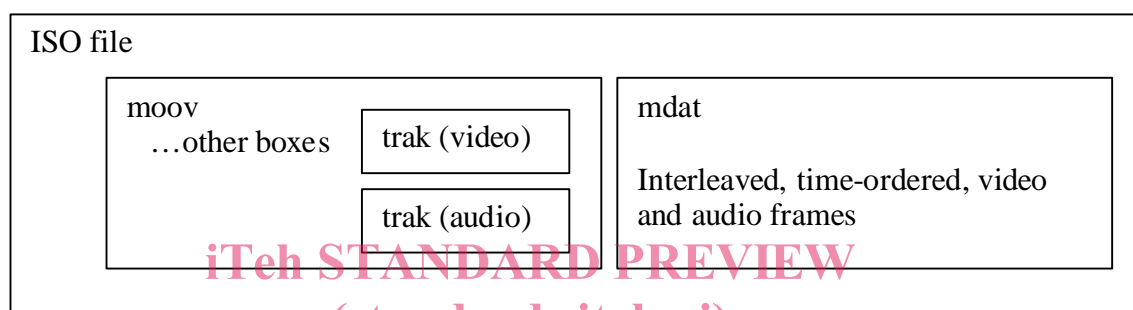


Figure 1 — Simple interchange file

#### 5.1.2 Content Creation

During content creation, a number of areas of the format can be exercised to useful effect, particularly:

- the ability to store each elementary stream separately (not interleaved), possibly in separate files.
- the ability to work in a single presentation that contains media data and other streams (e.g. editing the audio track in the uncompressed format, to align with an already-prepared video track).

These characteristics mean that presentations may be prepared, edits applied, and content developed and integrated without either iteratively re-writing the presentation on disc – which would be necessary if interleave was required and unused data had to be deleted; and also without iteratively decoding and re-encoding the data – which would be necessary if the data must be stored in an encoded state.

In the following diagram, a set of files being used in the process of content creation is shown.

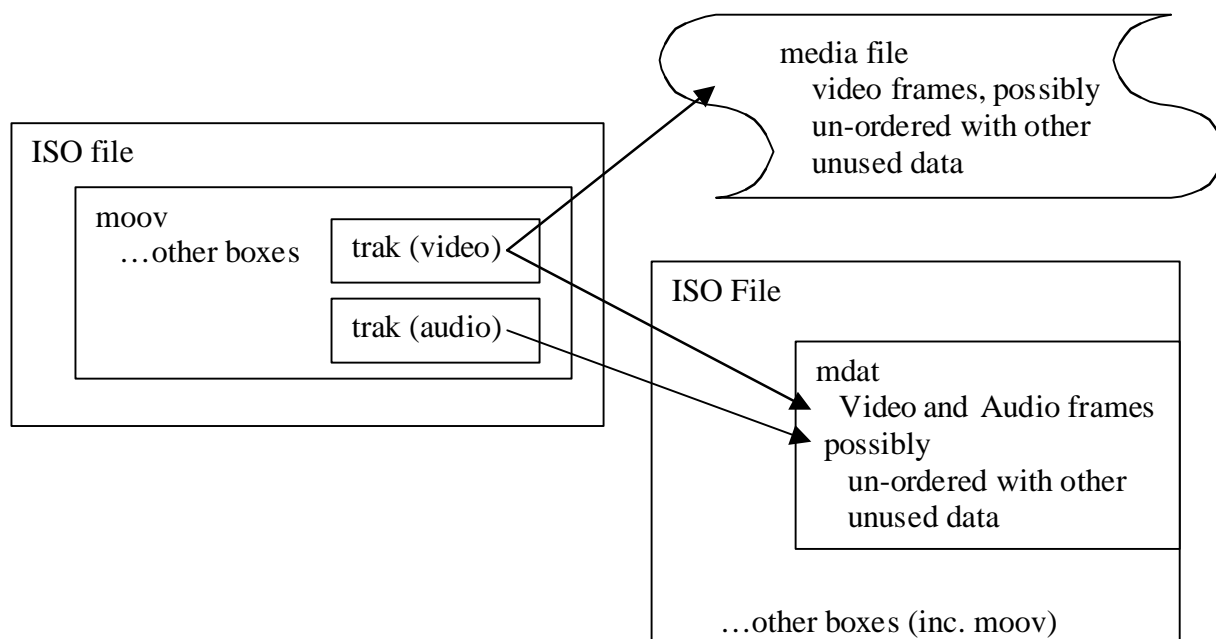


Figure 2 — Content Creation File

### 5.1.3 Preparation for streaming

When prepared for streaming, the file must contain information to direct the streaming server in the process of sending the information. In addition, it is helpful if these instructions and the media data are interleaved so that excessive seeking can be avoided when serving the presentation. It is also important that the original media data be retained unscathed, so that the files may be verified, or re-edited or otherwise re-used. Finally, it is helpful if a single file can be prepared for more than one protocol, so differing servers may use it over disparate protocols.

<https://standards.iteh.ai/catalog/standards/sist/914d619-013b-4900-977a-349bcce5bb11/iso-iec-14496-12-2004>

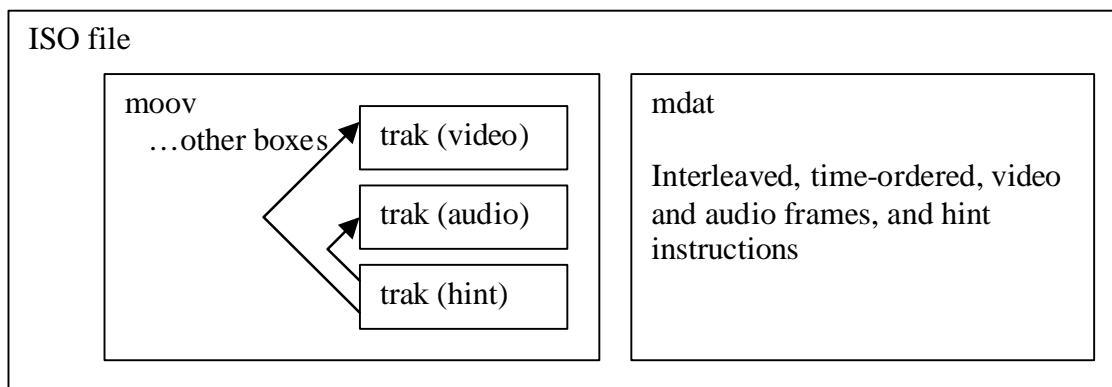
### 5.1.4 Local presentation

'Locally' viewing a presentation (i.e. directly from the file, not over a streamed interconnect) is an important application; it is used when a presentation is distributed (e.g. on CD or DVD ROM), during the process of development, and when verifying the content on streaming servers. Such local viewing must be supported, with full random access. If the presentation is on CD or DVD ROM, interleave is important as seeking may be slow.

### 5.1.5 Streamed presentation

When a server operates from the file to make a stream, the resulting stream must be conformant with the specifications for the protocol(s) used, and should contain no trace of the file-format information in the file itself. The server needs to be able to random access the presentation. It can be useful to re-use server content (e.g. to make excerpts) by referencing the same media data from multiple presentations; it can also assist streaming if the media data can be on read-only media (e.g. CD) and not copied, merely augmented, when prepared for streaming.

The following diagram shows a presentation prepared for streaming over a multiplexing protocol, only one hint track is required.



**Figure 3 — Hinted Presentation for Streaming**

## 5.2 Design principles

The file structure is object-oriented; a file can be decomposed into constituent objects very simply, and the structure of the objects inferred directly from their type.

Media-data is not 'framed' by the file format; the file format declarations that give the size, type and position of media data units are not physically contiguous with the media data. This makes it possible to subset the media-data, and to use it in its natural state, without requiring it to be copied to make space for framing. The metadata is used to describe the media data by reference, not by inclusion.

Similarly the protocol information for a particular streaming protocol does not frame the media data; the protocol headers are not physically contiguous with the media data. Instead, the media data can be included by reference. This makes it possible to represent media data in its natural state, not favoring any protocol. It also makes it possible for the same set of media data to serve for local presentation, and for multiple protocols.

The protocol information is built in such a way that the streaming servers need to know only about the protocol and the way it should be sent; the protocol information abstracts knowledge of the media so that the servers are, to a large extent, media-type agnostic. Similarly the media-data, stored as it is in a protocol-unaware fashion, enables the media tools to be protocol-agnostic.

The file format does not require that a single presentation be in a single file. This enables both sub-setting and re-use of content. When combined with the non-framing approach, it also makes it possible to include media data in files not formatted to this specification (e.g. 'raw' files containing only media data and no declarative information, or file formats already in use in the media or computer industries).

The file format is based on a common set of designs and a rich set of possible structures and usages. The same format serves all usages; translation is not required. However, when used in a particular way (e.g. for local presentation), the file may need structuring in certain ways for optimal behavior (e.g. time-ordering of the data). No normative structuring rules are defined by this specification, unless a restricted profile is used.

## 6 ISO Base Media File organization

### 6.1 Presentation structure

#### 6.1.1 File Structure

A presentation may be contained in several files. One file contains the metadata for the whole presentation, and is formatted to this specification. This file may also contain all the media data, whereupon the presentation is self-contained. The other files, if used, are not required to be formatted to this specification; they are used to contain media data, and may also contain unused media data, or other information. This specification concerns the structure of the presentation file only. The format of the media-data files is

constrained by this specification only in that the media-data in the media files must be capable of description by the metadata defined here.

These other files may be ISO files, image files, or other formats. Only the media data itself, such as JPEG 2000 images, is stored in these other files; all timing and framing (position and size) information is in the ISO base media file, so the ancillary files are essentially free-format.

If an ISO file contains hint tracks, the media tracks that reference the media data from which the hints were built shall remain in the file, even if the data within them is not directly referenced by the hint tracks; after deleting all hint tracks, the entire un-hinted presentation shall remain. Note that the media tracks may, however, refer to external files for their media data.

Annex A provides an informative introduction, which may be of assistance to first-time readers.

## 6.1.2 Object Structure

The file is structured as a sequence of objects; some of these objects may contain other objects. The sequence of objects in the file shall contain exactly one presentation metadata wrapper (the Movie Box). It is usually close to the beginning or end of the file, to permit its easy location. The other objects found at this level may be a File-Type box, Free Space Boxes, Movie Fragments, or Media Data Boxes.

## 6.1.3 Meta Data and Media Data

The metadata is contained within the metadata wrapper (the Movie Box); the media data is contained either in the same file, within Media Data Box(es), or in other files. The media data is composed of images or audio data; the media data objects, or media data files, may contain other un-referenced information.

## 6.1.4 Track Identifiers

The track identifiers used in an ISO file are unique within that file; no two tracks shall use the same identifier.

The next track identifier value stored in `next_track_ID` in the Movie Header Box generally contains a value one greater than the largest track identifier value found in the file. This enables easy generation of a track identifier under most circumstances. However, if this value is equal to ones (32-bit unsigned maxint), then a search for an unused track identifier is needed for all additions.

## 6.2 Metadata Structure (Objects)

### 6.2.1 Box

Type fields not defined here are reserved. Private extensions shall be achieved through the 'uuid' type. In addition, the following types are not and will not be used, or used only in their existing sense, in future versions of this specification, to avoid conflict with existing content using earlier pre-standard versions of this format:

```
clip, crgn, matt, kmat, pnot, ctab, load, imap;  
these track reference types (as found in the reference_type of a Track Reference Box): tmcd, chap,  
sync, scpt, ssrc.
```

A number of boxes contain index values into sequences in other boxes. These indexes start with the value 1 (1 is the first entry in the sequence).

### 6.2.2 Data Types and fields

In a number of boxes in this specification, there are two variant forms: version 0 using 32-bit fields, and version 1 using 64-bit sizes for those same fields. In general, if a version 0 box (32-bit field sizes) can be used, it should be; version 1 boxes should be used only when the 64-bit field sizes they permit, are required.

For convenience during content creation there are creation and modification times stored in the file. These can be 32-bit or 64-bit numbers, counting seconds since midnight, Jan. 1, 1904, which is a convenient date for leap-year calculations. 32 bits are sufficient until approximately year 2040. These times shall be expressed in UTC, and therefore may need adjustment to local time if displayed.

Fixed-point numbers are signed or unsigned values resulting from dividing an integer by an appropriate power of 2. For example, a 30.2 fixed-point number is formed by dividing a 32-bit integer by 4.

Fields shown as “template” in the box descriptions are optional in the specifications that use this specification. If the field is used in another specification, that use must be conformant with its definition here, and the specification must define whether the use is optional or mandatory. Similarly, fields marked “pre-defined” were used in an earlier version of this specification. For both kinds of fields, if a field of that kind is not used in a specification, then it should be set to the indicated default value. If the field is not used it must be copied un-inspected when boxes are copied, and ignored on reading.

Matrix values which occur in the headers specify a transformation of video images for presentation. Not all derived specifications use matrices; if they are not used, they shall be set to the identity matrix. If a matrix is used, the point (p,q) is transformed into (p', q') using the matrix as follows:

$$\begin{pmatrix} p & q & 1 \end{pmatrix} * \begin{vmatrix} a & b & u \\ c & d & v \\ x & y & w \end{vmatrix} = \begin{pmatrix} m & n & z \end{pmatrix}$$

$$m = ap + cq + x; \quad n = bp + dq + y; \quad z = up + vq + w;$$

$$p' = m/z; \quad q' = n/z$$

The coordinates {p,q} are on the decompressed frame, and {p',q'} are at the rendering output. Therefore, for example, the matrix {2,0,0, 0,2,0, 0,0,1} exactly doubles the pixel dimension of an image. The co-ordinates transformed by the matrix are not normalized in any way, and represent actual sample locations. Therefore {x,y} can, for example, be considered a translation vector for the image.

The co-ordinate origin is located at the upper left corner, and X values increase to the right, and Y values increase downwards. {p,q} and {p',q'} are to be taken as absolute pixel locations relative to the upper left hand corner of the original image (after scaling to the size determined by the track header's width and height) and the transformed (rendering) surface, respectively.

Each track is composed using its matrix as specified into an overall image; this is then transformed and composed according to the matrix at the movie level in the MovieHeaderBox. It is application-dependent whether the resulting image is ‘clipped’ to eliminate pixels, which have no display, to a vertical rectangular region within a window, for example. So for example, if only one video track is displayed and it has a translation to {20,30}, and a unity matrix is in the MovieHeaderBox, an application may choose not to display the empty “L” shaped region between the image and the origin.

All the values in a matrix are stored as 16.16 fixed-point values, except for u, v and w, which are stored as 2.30 fixed-point values.

The values in the matrix are stored in the order {a,b,u, c,d,v, x,y,w}.

### 6.2.3 Box Order

An overall view of the normal encapsulation structure is provided in the following table.

The table shows those boxes that may occur at the top-level in the left-most column; indentation is used to show possible containment. Thus, for example, a Track Header Box (tkhd) is found in a Track Box (trak), which is found in a Movie Box (moov). Not all boxes need be used in all files; the mandatory boxes are marked with an asterisk (\*). See the description of the individual boxes for a discussion of what must be assumed if the optional boxes are not present.