Common File Format & Media Formats Specification

Version 2.0.1 22 December 2014

Common File Format & Media Formats Specification Version 2.0.1

Notice:
As of the date of publication, this document is a release candidate specification subject to DECE Member review and final adoption by vote of the Management Committee of DECE in accordance with the DECE LLC Operating Agreement. Unless there is notice to the contrary, this specification will become an adopted "Ecosystem Specification" on 5 February 2015.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Digital Entertainment Content Ecosystem (DECE) LLC ("DECE") and its members disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein. This document is subject to change under applicable license provisions, if any.

Optional Implementation Agreement:
In addition to the UltraViolet License Agreements which cover implementation of the DECE Ecosystem Specifications within the UltraViolet Ecosystem, DECE offers an optional license agreement relating to the implementation of this document outside the Ecosystem ("RAND Agreement"). Entities executing the optional RAND Agreement receive the benefit of the commitments made by DECE's members to license on reasonable and nondiscriminatory terms their patent claims necessary to the implementation of this document in exchange for a comparable patent licensing commitment. Copies of the license agreements are available at the DECE web site referenced below.

Contact Information:
Licensing and contract inquiries and requests should be addressed to us at:
http://www.uvvu.com/uv-for-business

Common File Format & Media Formats Specification Version 2.0.1

Contents

Tables

Figures

1 Introduction

1.1 Scope

This specification defines the Common File Format and the media formats it supports for the storage, delivery and playback of audio-visual content. It includes a common media file format, elementary stream formats, elementary stream encryption formats and metadata designed to optimize the distribution, purchase, delivery from multiple publishers, retailers, and content distribution networks; and enable playback on multiple authorized devices using multiple DRM systems within an ecosystem.

1.2 Document Organization

The Common File Format (CFF) defines a container for audio-visual content based on the ISO Base Media File Format [ISO]. This specification defines the set of technologies and configurations used to encode that audio-visual content for presentation. The core specification addresses the structure, content and base level constraints that apply to all variations of Common File Format content and how it is to be stored within a Digital CFF Container (DCC). This specification defines how video, audio and subtitle content intended for synchronous playback is stored within a compliant file, as well as how one or more co-existing digital rights management systems can be used to protect that content cryptographically.

Media Profiles are defined in the Annexes of this document. These profiles specify additional requirements and constraints that are particular to a given class of content. Over time, additional Media Profiles might be added, but such additions would not typically require modification to the core specification.

1.3 Document Notation and Conventions

The following terms are used to specify conformance elements of this specification. These are adopted from the ISO/IEC Directives, Part 2, Annex H [ISO-P2H]. For more information, please refer to those directives.

- SHALL and SHALL NOT indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.
- SHOULD and SHOULD NOT indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
- MAY and NEED NOT indicate a course of action permissible within the limits of the document.

Terms defined to have a specific meaning within this specification will be capitalized, e.g. “DCC Movie Fragment”, and should be interpreted with their general meaning if not capitalized.

1.4 Normative References

1.4.1 DECE References

[DSystem]	System Specification
[DDMP]	Media Package Specification
[DMeta]	Content Metadata Specification
[DStream]	Common Streaming Protocol Specification

Note: Other DECE documents contain requirements for an UltraViolet-compliant implementation, particularly the Licensee Implementation Requirements as incorporated into the Compliance Rules of Licensee Agreements.

1.4.2 External References

[AAC]	ISO/IEC 14496-3:2009, “Information technology — Coding of audio-visual objects — Part 3: Audio” with: Amendment 1, Amendment 2, Amendment 3, Amendment 4, Corrigendum 1, Corrigendum 2, Corrigendum 3
[AACC]	ISO/IEC 14496-26:2010, " Information technology — Coding of audio-visual objects — Part 26: Audio conformance” with Amendment 2, Corrigendum 2, Corrigendum 3, Corrigendum 4, Corrigendum 5, Corrigendum 6
[AES]	Advanced Encryption Standard, Federal Information Processing Standards Publication 197, FIPS-197, http://www.nist.gov
[CENC]	ISO/IEC 23001-7:2014, Second edition, "Information technology - MPEG systems technologies - Part 7: Common encryption in ISO base media file format files"
[DASH]	ISO/IEC 23009-1:2014, Second edition 2014-05-15, “Information technology — Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats”
[DTS]	ETSI TS 102 114 v1.4.1 (2012-09), “DTS Coherent Acoustics; Core and Extensions with Additional Profiles"
[EAC3]	ETSI TS 102 366 v. 1.3.1 (2014-08), “Digital Audio Compression (AC-3, Enhanced AC-3) Standard”
[H264]	ISO/IEC 14496-10:2014, Eighth edition 2014-09-01, "Information technology - Coding of audio-visual objects - Part 10: Advanced Video Coding"
[H265]	ISO/IEC 23008-2:2013, First edition 2013-12-01, “Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding”
[IANA-LANG]	IANA Language Subtag Registry http://www.iana.org/assignments/language-subtag-registry

[ISO]	ISO/IEC 14496-12:2012, Fourth edition 2012-07-15, Corrected version 2012-09-15, "Information technology - Coding of audio-visual objects – Part 12: ISO Base Media File Format" with Amendment 1, Amendment 2, Corrigendum 1, Corrigendum 2, Corrigendum 3, Corrigendum 4
[ISOVIDEO]	ISO/IEC 14496-15:2014, Third edition 2014-07-01, "Information technology - Coding of audio-visual objects - Part 15: Carriage of NAL unit structured video in the ISO Base Media File Format " with: Corrigendum 1
[ISOTEXT]	ISO/IEC 14496-30:2014, First edition 2014-03-15, "Timed text and other visual overlays in ISO base media file format" with Corrigendum 1
[ISO-P2H]	ISO/IEC Directives, Part 2, Annex H http://www.iec.ch/tiss/iec/Directives-part2-Ed5.pdf
[MHP]	ETSI TS 101 812 V1.3.1, "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.3", available from www.etsi.org.
[MLP]	Meridian Lossless Packing, Technical Reference for FBA and FBB streams, Version 1.0, October 2005, Dolby Laboratories, Inc.
[MLPISO]	MLP (Dolby TrueHD) streams within the ISO Base Media File Format, Version 1.0, Dolby Laboratories, Inc.
[MP4]	ISO/IEC 14496-14:2003, First edition 2003-11-15, "Information technology - Coding of audio-visual objects - Part 14: MP4 file format" with: Amendment 1, Corrigendum 1
[MP4RA]	Registration authority for code-points in the MPEG-4 family, http://www.mp4ra.org
[MPEG4S]	ISO/IEC 14496-1:2010, Fourth edition 2010-06-01, "Information technology - Coding of audio-visual objects - Part 1: Systems" with: Amendment 1:2010
[MPS]	ISO/IEC 23003-1:2007, "Information technology — MPEG audio technologies — Part 1: MPEG Surround" with Corrigendum 1, Corrigendum 2, Corrigendum 3, Corrigendum 4
[R601]	ITU-R Recommendation BT.601-7, "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios"
[R709]	ITU-R Recommendation BT.709-5, "Parameter values for the HDTV standards for production and international programme exchange"
[R1700]	ITU-R Recommendation BT.1700, "Characteristics of composite video signals for conventional analogue television systems"
[R1886]	ITU-R Recommendation BT.1886, "Reference electro-optical transfer function for flat panel displays used in HDTV studio production"
[RFC2141]	"URN Syntax", R.Moats, May 1997, http://www.ietf.org/rfc/rfc2141.txt
[RFC4122]	Leach, P., et al, A Universally Unique Identifier (UUID) URN Namespace, July 2005 http://www.ietf.org/rfc/rfc4122.txt

[RFC4151]	“The 'tag' URI Scheme”, T. Kindberg and S. Hawke, October 2005 http://www.ietf.org/rfc/rfc4151.txt
[RFC5646]	“Tags for Identifying Languages” A.Philips and M. Davis, September, 2009, http://www.ietf.org/rfc/rfc5646.txt
[RFC6381]	“The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types” R. Gellens, et al, August, 2011, http://www.ietf.org/rfc/rfc6381.txt
[SMPTE428]	SMPTE 428-3-2006, “D-Cinema Distribution Master Audio Channel Mapping and Channel Labeling” (c) SMPTE 2006
[SMPTE-TT]	SMPTE ST2052-1:2010, “Timed Text Format (SMPTE-TT)”
[SMPTE-608]	SMPTE RP2052-10:2012, “Conversion from CEA-608 Data to SMPTE-TT”
[SMPTE-708]	SMPTE RP2052-11, “Conversion from CEA-708 Data to SMPTE-TT”
[XML]	“XML Schema Part 1: Structures Second Edition”, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-1/ “XML Schema Part 2: Datatypes Second Edition”, Paul Biron and Ashok Malhotra, W3C Recommendation 28 October 2004, http://www.w3.org/TR/xmlschema-2/
[UNICODE]	UNICODE 6.0.0, “The Unicode Standard Version 6.0”, http://www.unicode.org/versions/Unicode6.0.0/

Note: Readers are encouraged to investigate the most recent publications for their applicability.

1.5 Informative References

[ATSC]	A/153 Part-7:2009, “ATSC-Mobile DTV Standard, Part 7 — AVC and SVC Video System Characteristics”
[RFC3986]	“Uniform Resource Identifier (URI): Generic Syntax” T. Berners-Lee, R. Fielding and L. Masinter, January 2005. http://www.ietf.org/rfc/rfc3986.txt
[W3C-TT]	Timed Text Markup Language (TTML) 1.0 (Second Edition), http://www.w3.org/TR/2013/PER-ttaf1-dfxp-20130709/

1.6 Terms, Definitions, and Acronyms

AAC As defined in [AAC], “Advanced Audio Coding.”

AAC LC A low complexity audio tool used in AAC profile, defined in [AAC].

access unit, AU As defined in [MPEG4S], “smallest individually accessible portion of data within an elementary stream to which unique timing information can be attributed.”

active picture area In a video track, the active picture area is the rectangular set of pixels that can contain video content at any point throughout the duration of the track, absent of any additional matting that is not considered by the content publisher to be an integral part of the video content.

ADIF As defined in [AAC], “Audio Data Interchange Format.”

ADTS	As defined in [AAC], “Audio Data Transport Stream.”
AES-CTR	Advanced Encryption Standard, Counter Mode
audio stream	A sequence of synchronized audio frames.
audio frame	A component of an audio stream that corresponds to a certain number of PCM audio samples.
AVC	Advanced Video Coding [H264].
AVC level	A set of performance constraints specified in Annex A.3 of [H264], such as maximum bit rate, maximum number of macroblocks, maximum decoding buffer size, etc.
AVC profile	A set of encoding tools and constraints defined in Annex A.2 of [H264].
box	As defined in [ISO], “object-oriented building block defined by a unique type identifier and length.”
CBR	As defined in [H264], “Constant Bit Rate.”
CFF	Common File Format. (See “Common File Format.”)
CFF-TT	“Common File Format Timed Text” is the Subtitle format defined by this specification.
chunk	As defined in [ISO], “contiguous set of samples for one track.”
coded video sequence (CVS)	As defined in [H264] for AVC video tracks and as defined in [H265] for HEVC video tracks.
Common File Format (CFF)	The standard DECE content delivery file format, encoded in one of the approved Media Profiles and packaged (encoded and encrypted) as defined by this specification.
container box	As defined in [ISO], “box whose sole purpose is to contain and group a set of related boxes.”
core	In the case of DTS, a component of an audio frame conforming to [DTS].
CPE	As defined in [AAC], an abbreviation for `channel_pair_element()`.
DCC Footer	The collection of boxes defined by this specification that might form the end of a Digital CFF Container (DCC), defined in Section 2.1.4.
DCC Header	The collection of boxes defined by this specification that form the beginning of a Digital CFF Container (DCC), defined in Section 2.1.2.
DCC Movie Fragment	The collection of boxes defined by this specification that form a *fragment* of a media track containing one type of media (i.e. audio, video, subtitles), defined by Section 2.1.3.
DECE	Digital Entertainment Content Ecosystem
Digital CFF Container (DCC)	An instance of Content published in the Common File Format.
descriptor	As defined in [MPEG4S], “data structure that is used to describe particular aspects of an elementary stream or a coded audio-visual object.”
DRM	Digital Rights Management.
extension	In the case of DTS, a component of an audio frame that might or might not exist in sequence with other extension components or a core component.
file format	A definition of how data is codified for storage in a specific type of file.

fragment	A segment of a track representing a single, continuous portion of the total duration of content (i.e. video, audio, subtitles) stored within that track.
HD	High Definition; Picture resolution of one million or more pixels like HDTV.
HE AAC	MPEG-4 High Efficiency AAC profile, defined in [AAC].
HEVC	High Efficiency Video Coding [H265]
HEVC tier and level	A set of performance constraints specified in Annex A.4 of [H265], such as maximum bit rate, maximum number of macroblocks, maximum decoding buffer size, etc.
HEVC profile	A set of encoding tools and constraints defined in Annex A.3 of [H265].
hint track	As defined in [ISO], "special track which does not contain media data, but instead contains instructions for packaging one or more tracks into a streaming channel."
horizontal sub-sample factor	Sub-sample factor for the horizontal dimension. See 'sub-sample factor', below.
IMDCT	Inverse Modified Discrete Cosine Transform.
ISO	In this specification "ISO" is used to refer to the ISO Base Media File format defined in [ISO], such as in "ISO container" or "ISO media file". It is also the acronym for "International Organization for Standardization".
ISO Base Media File	File format defined by [ISO].
Kbps	1x10^3 bits per second.
LFE	Low Frequency Effects.
late binding	The combination of separately stored audio, video, subtitles, metadata, or DRM licenses with a preexisting video file for playback as though the late bound content was incorporated in the preexisting video file.
luma	As defined in [H264], "An adjective specifying that a sample array or single sample is representing the monochrome signal related to the primary colours."
Mbps	1x10^6 bits per second.
media format	A set of technologies with a specified range of configurations used to encode "media" such as audio, video, pictures, text, animation, etc. for audio-visual presentation.
Media Profile	Requirements and constraints such as resolution and subtitle format for content in the Common File Format.
MPEG	Moving Picture Experts Group.
MPEG-4 AAC	Advanced Audio Coding, MPEG-4 Profile, defined in [AAC].
NAL Structured Video	Network Abstraction Layer Structured Video; a technical approach to format the Video Coding Layer (VCL) representation of the video such that header information is conveyed in a manor which is appropriate for a variety of transport layers.
PD	Portable Definition; intended for portable devices such as cell phones and portable media players.

presentation As defined in [ISO], “one or more motion sequences, possibly combined with audio.”

progressive download The initiation and continuation of playback during a file copy or download, beginning once sufficient file data has been copied by the playback device.

PS As defined in [AAC], “Parametric Stereo.”

sample As defined in [ISO], “all the data associated with a single timestamp.” (Not to be confused with an element of video spatial sampling.)

sample aspect ratio, SAR As defined in [H264], “the ratio between the intended horizontal distance between the columns and the intended vertical distance between the rows of the *luma* sample array in a frame. Sample aspect ratio is expressed as *h*:*v*, where *h* is horizontal width and *v* is vertical height (in arbitrary units of spatial distance).”

sample description As defined in [ISO], “structure which defines and describes the format of some number of samples in a track.”

SBR As defined in [AAC], “Spectral Band Replication.”

SCE As defined in [AAC], an abbreviation for `single_channel_element()`.

SD Standard Definition; used on a wide range of devices including analog television

sub-sample factor A value used to determine the constraints for choosing valid `width` and `height` field values for a video track, specified in Section 4.5.1.1.

sub-sampling In video, the process of encoding picture data at a lower resolution than the original source picture, thus reducing the amount of information retained.

substream In audio, a sequence of synchronized audio frames comprising only one of the logical components of the audio stream.

track As defined in [ISO], “timed sequence of related samples (q.v.) in an ISO base media file.”

track fragment A combination of metadata and sample data that defines a single, continuous portion (“fragment”) of the total duration of a given track.

VBR As defined in [H264], “Variable Bit Rate.”

vertical sub-sample factor Sub-sample factor for the vertical dimension. See ‘sub-sample factor’, above.

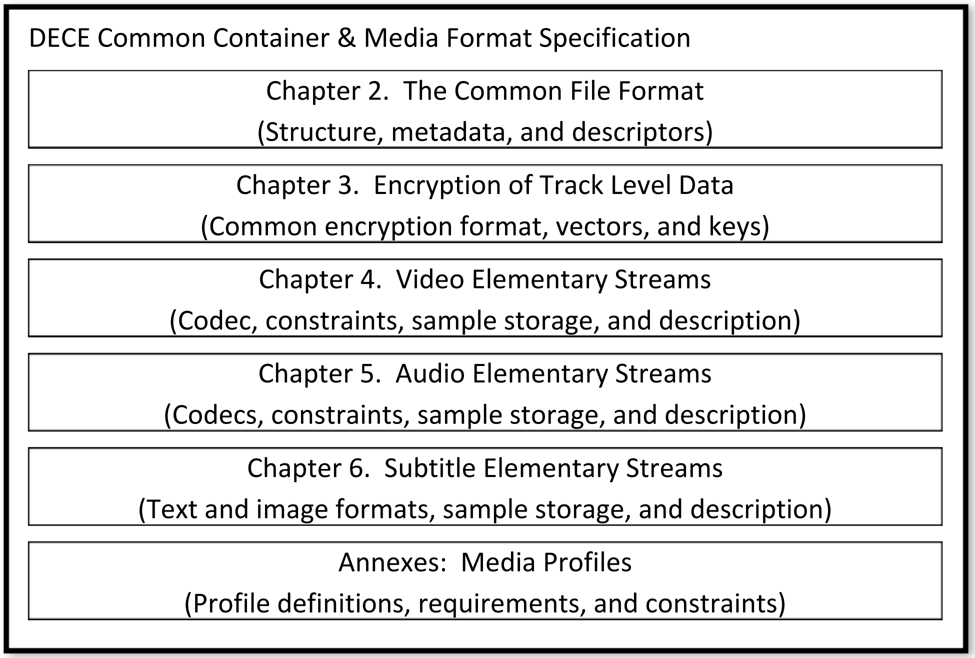
XLL A logical element within the DTS elementary stream containing compressed audio data that will decode into a bit-exact representation of the original signal.

1.7 Architecture (Informative)

The following subsections describe the components of a Digital CFF Container (DCC) and how they are combined or “layered” to make a complete file. The specification itself is organized in sections corresponding to layers, also incorporating normative references, which combine to form the complete specification.

1.7.1 Media Layers

This specification can be thought of as a collection of layers and components. This document and the normative references it contains are organized based on those layers.



DECE Common Container & Media Format Specification

Chapter 2. The Common File Format
(Structure, metadata, and descriptors)

Chapter 3. Encryption of Track Level Data
(Common encryption format, vectors, and keys)

Chapter 4. Video Elementary Streams
(Codec, constraints, sample storage, and description)

Chapter 5. Audio Elementary Streams
(Codecs, constraints, sample storage, and description)

Chapter 6. Subtitle Elementary Streams
(Text and image formats, sample storage, and description)

Annexes: Media Profiles
(Profile definitions, requirements, and constraints)

Figure 1-1 – Structure of the Common File Format & Media Formats Specification

1.7.2 Common File Format

Section 2 of this specification defines the *Common File Format* (CFF) derived from the ISO Base Media File Format and `‘iso6’` brand specified in [ISO]. This section specifies restrictions and additions to the file format and clarifies how content streams and metadata are organized and stored.

The `‘iso6’` brand of the ISO Base Media File Format consists of a specific collection of *boxes*, which are the logical containers defined in the ISO specification. Boxes contain *descriptors* that hold parameters derived from the contained content and its structure. One of the functions of this specification is to equate or map the parameters defined in elementary stream formats and other normative specifications to descriptors in ISO boxes, or to elementary stream samples that are logically contained in *media data boxes*. Physically, the ISO Base Media File Format allows storage of elementary stream *access units* in any sequence and any grouping, intact or subdivided into packets, within or externally to the file. Access units defined in each elementary stream are mapped to logical *samples* in the ISO media file using references to byte positions inside the file where the access units are stored. The logical sample information allows access units to be decoded and presented synchronously on a timeline, regardless of storage, as long as the entire ISO media file and sample storage files are randomly accessible and there are no performance or memory constraints. In practice, additional physical storage constraints are usually required in order to ensure uninterrupted, synchronous playback.

To enable useful file delivery scenarios, such as *progressive download*, and to improve interoperability and minimize device requirements; the CFF places restrictions on the physical storage of elementary streams and their access units. Rather than employ an additional systems layer, the CFF stores a small number of elementary stream access units with each *fragment* of the ISO *track* that references those access units as samples.

Because logical metadata and physical sample storage is grouped together in the CFF, each segment of an ISO track has the necessary metadata and sample data for decryption and decoding that is optimized for random access playback and progressive download.

1.7.3 Track Encryption and DRM support

DECE specifies a standard encryption scheme and key mapping that can be used with multiple DRM systems capable of providing the necessary key management and protection, content usage control, and device authentication and authorization. Standard encryption algorithms are specified for regular, opaque sample data, and for video data with sub-sample level headers exposed to enable reformatting of video streams without decryption. The "Scheme" method specified in [ISO] is required for all encrypted files. This method provides accessible key identification and mapping information that an authorized DRM system can use to create DRM-specific information, such as a license, that can be stored in a reserved area within the file, or delivered separately from the file.

1.7.3.1 DRM Signaling and License Embedding

Each DRM system that embeds DRM-specific information in a DCC file does so by creating a DRM-specific box in the Movie Box (`‘moov’`) as described in Section 3. This box can be used to store DRM-specific information, such as license acquisition objects, rights objects, licenses and other information. This information is used by the specific DRM system to enable content decryption and playback.

Alternatively, DRM-specific boxes can be stored in the Media Package as specified by [DDMP].

1.7.4 Video Elementary Streams

This specification supports the use of NAL Structured Video elementary streams encoded according to the AVC codec specified in [H264] or the HEVC codec specified in [H265] and stored in the Common File Format in accordance with [ISOVIDEO], with some additional requirements and constraints.

1.7.5 Audio Elementary Streams

A wide range of audio coding technologies are supported for inclusion in the Common File Format, including several based on *MPEG-4 AAC* as well as Dolby™ and DTS™ formats. Consistent with MPEG-4 architecture, AAC elementary streams specified in this format only include raw audio samples in the elementary bit-stream. These raw audio samples are mapped to access units at the elementary stream level and samples at the container layer. Other syntax elements typically included for synchronization, packetization, decoding parameters, content format, etc. are mapped either to descriptors at the container layer, or are eliminated because the ISO container already provides comparable functions, such as sample identification and synchronization.

In the case of Dolby and DTS formats, complete elementary streams normally used by decoders are mapped to access units and stored as samples in the container. Some parameters already included in the bit-streams are duplicated at the container level in accordance with ISO media file requirements. During playback, the complete elementary stream, which is present in the stored samples, is sent to the decoder for presentation. The decoder uses the in-band decoding and stream structure parameters specified by each codec.

These codecs use a variety of different methods and structures to map and mix channels, as well as sub- and extension streams to scale from 2.0 channels to 7.1 channels and enable increasing levels of quality. Rather than trying to describe and enable all the decoding features of each stream using ISO tracks and sample group layers, the Common File Format identifies only the maximum capability of each stream at the container level (e.g. "7.1 channel lossless") and allows standard decoders for these codecs to decode using the in-band information (as is typically done in the installed base of these decoders).

1.7.6 Subtitle Elementary Streams

This specification supports the use of both image and text-based subtitles in the Common File Format using the SMPTE TT format defined in [SMPTE-TT]. An extension of the W3C Timed Text Markup Language, subtitles are stored as a series of SMPTE TT documents and, optionally, PNG images. A single Digital CFF Container can contain multiple subtitle tracks, which are composed of fragments, each containing a single sample that maps to a SMPTE TT document and any images it references. The subtitles themselves can be stored in character coding form (e.g. Unicode) or as sub-pictures, or both. Subtitle tracks can address purposes such as normal captions, subtitles for the deaf and hearing impaired, descriptive text, and commentaries, among others.

1.7.7 Media Profiles and Delivery Targets

The Common File Format defines all of the general requirements and constraints for a conformant DCC. In addition, Annex B. of this document defines requirements for specific Media Profiles and Annex C. of this document defines requirements for specific Delivery Targets.

Media Profiles normatively define distinct subsets of the elementary stream formats that can be stored within a DCC in order to ensure interoperability with certain classes of devices. These restrictions include mandatory and optional codecs, picture format restrictions and codec parameter restrictions, among others. In general, each Media Profile defines the maximum set of tools and performance parameters content is permitted to use and still comply with the Media Profile. However, compliant content can use less than the maximum limits, unless otherwise specified. This makes it possible for a device that decodes a higher Media Profile of content to also be able to decode files that conform to lower Media Profiles, though the reverse is not necessarily true.

Delivery Targets normatively define additional restrictions on elementary stream formats and DCC file structure to support particular types of delivery methods such as download or streaming.

A conformant DCC will comply with the general requirements defined in this specification together with at least one Media Profile and at least one Delivery Target. It is possible for a DCC to simultaneously comply with more than one Delivery Target.

Compliant devices are expected to gracefully ignore metadata and format options they do not support.

Over time, additional Media Profiles and Delivery Targets might be added in order to support new features, formats and capabilities.

2 The Common File Format

The Common File Format (CFF) is based on an enhancement of the ISO Base Media File Format defined by [ISO]. The principal enhancements to the ISO Base Media File Format are support for multiple DRM technologies and separate storage of audio, video, and subtitle samples in track fragments to allow flexible delivery methods (including progressive download) and playback.

2.1 Common File Format

The Common File Format is defined by the 'ccff' brand, which is a code point on the ISO Base Media File Format defined by [ISO]. The brand 'ccff' requires support for all features of the 'iso6' brand as defined in [ISO]. In addition, this specification defines boxes, requirements and constraints that are required in addition to those defined by [ISO]; included are constraints on the layout of certain information within the container in order to improve interoperability, random access playback and progressive download.   The following boxes are extensions for the Common File Format:

- 'coin': Content Information Box (see Section 2.2.2)
- 'avcn': AVC NAL Unit Storage Box (not recommended for use – see Section 2.2.3)
- 'senc': Sample Encryption Box (see Section 2.2.4)
- 'trik': Trick Play Box (not recommended for use – see Section 2.2.5)

Table 2-1 shows the box type, structure, nesting level and cross-references for the Common File Format. The nesting in Table 2 1 indicates containment, not necessarily order. Differences and extensions to the ISO Base Media File Format are highlighted. Unless otherwise prohibited in this specification, the DCC and any box within it can contain additional boxes to the extent permitted by [ISO].

Table 2-1 - Box structure of the Common File Format (CFF)

NL 0	NL 1	NL 2	NL 3	NL 4	NL 5	Format Req.	Specification	Description
ftyp						1	Section 2.3.1	File Type and Compatibility
moov						1	[ISO] 8.2.1	Container for functional metadata
	mvhd					1	[ISO] 8.2.2	Movie header
	coin					1	Section 2.2.2	Content Information Box
	meta					0/1	[ISO] 8.11.1	Multi-Track Required Metadata
		hdlr				1	Section 2.3.3	Handler for common file metadata
		xml				1	Section 2.3.4.1	XML for Multi-Track Required Metadata
		iloc				1	[ISO] 8.11.3	Item Location (i.e. for XML references to mandatory images, etc.)
		idat				0/1	[ISO] 8.11.11	Container for Metadata image files
	trak					+	[ISO] 8.3.1	Container for each track
		tkhd				1	[ISO] 8.3.2	Track header
		edts				0/1	[ISO] 8.6.5	Edit Box
			elst			0/1	[ISO] 8.6.6	Edit List Box
		mdia				1	[ISO] 8.4	Track Media Information
			mdhd			1	Section 2.3.6	Media Header
			hdlr			1	[ISO] 8.4.3	Declares the media handler type
			minf			1	[ISO] 8.4.4	Media Information container
				vmhd		0/1	Section 2.3.7	Video Media Header
				smhd		0/1	Section 2.3.8	Sound Media Header
				sthd		0/1	Section 2.3.9	Subtitle Media Header
				dinf		1	[ISO] 8.7.1	Data Information Box
					dref	1	Section 2.3.10	Data Reference Box, declares source of media data in track
				stbl		1	[ISO] 8.5	Sample Table Box, container for the time/space map
					stsd	1	Section 2.3.11	Sample Descriptions (See Table 2-2 for additional detail.)
					stts	1	Section 2.3.12	Decoding, Time to Sample
					stsc	1	Section 2.3.16	Sample-to-Chunk
					stsz / stz2	1	Section 2.3.13	Sample Size Box
					stco	1	Section 2.3.17	Chunk Offset
	mvex					1	[ISO] 8.8.1	Movie Extends Box
		mehd				1	[ISO] 8.8.2	Movie Extends Header
		trex				+ (1 per track)	Section 2.3.17	Track Extends Box

NL 0	NL 1	NL 2	NL 3	NL 4	NL 5	Format Req.	Specification	Description
	pssh					*	[CENC] 8.1	Protection System Specific Header Box
	free					0/1	[ISO] 8.1.2	Free Space Box reserved space for DRM information
sidx						0 / 1	Section 2.3.21	Segment Index Box
emsg						*	[DASH] 5.10.3.3	Event message box
moof						+	[ISO] 8.8.4	Movie Fragment
	mfhd					1	Section 2.3.18	Movie Fragment Header
	traf					1	[ISO] 8.8.6	Track Fragment
		tfhd				1	Section 2.3.19	Track Fragment Header
		tfdt				1	[ISO] 8.8.12	Track Fragment Base Media Decode Time
		trik				0/1 for AVC video which includes (‘avc1’) sample entries, 0 for others	Section 2.2.7	Trick Play Box
		trun				1	Section 2.3.20	Track Fragment Run Box
		avcn				0/1 for AVC video which includes (‘avc1’) sample entries, 0 for others	Section 2.2.2	AVC NAL Unit Storage Box
		senc				0/1	Section 2.2.6	Sample Encryption Box
		saio				+ if encrypted, * if unencrypted	[ISO] 8.7.13	Sample Auxiliary Information Offsets Box
		saiz				+ if encrypted, * if unencrypted	[ISO] 8.7.12	Sample Auxiliary Information Sizes Box
		sbgp				*	[ISO] 8.9.2	Sample to Group Box
		sgpd				*	[ISO] 8.9.3	Sample Group Description Box
mdat						+	Section 2.3.22	Media Data container for media samples
meta						0/1	[ISO] 8.11.1	Multi-Track Optional Metadata
	hdlr					0/1	Section 2.3.3	Handler for common file metadata
	xml					0/1	Section 2.3.4.2	XML for Multi-Track Optional Metadata
	iloc					0/1	[ISO] 8.11.3	Item Location (i.e. for XML references to optional images, etc.)
	idat					0/1	[ISO] 8.11.11	Container for Metadata image files
mfra						0 / 1	[ISO] 8.8.9	Movie Fragment Random Access
	tfra					+ (one per track)	Section 2.3.18	Track Fragment Random Access
	mfro					1	[ISO] 8.8.11	Movie Fragment Random Access Offset

Format Req.: Number of boxes required to be present in the container, where '*' means "zero or more" and '+' means "one or more". A value of "0/1" indicates only that a box might or might not be present but does not stipulate the conditions of its appearance.

Table 2-2 – Additional 'stsd' Detail: Protected Sample Entry Box structure

NL 5	NL 6	NL 7	NL 8	Format Req	Source	Description
stsd				1	Section 2.3.11	Sample Description Box
	sinf			*	ISO 8.12.1	Protection Scheme Information Box
		frma		1	ISO 8.12.2	Original Format Box
		schm		1	[ISO] 8.12.5	Scheme Type Box
		schi		1	[ISO] 8.12.6	Scheme Information Box
			tenc	1	[CENC] 8.2	Track Encryption Box

2.1.1 Digital CFF Container Structure

For the purpose of this specification, the Digital CFF Container (DCC) structure defined by the Common File Format is divided into three sections: DCC Header, DCC Movie Fragments, and DCC Footer, as shown in Figure 2-1.

- A Digital CFF Container starts with a DCC Header, as defined in Section 2.1.2.
- One or more DCC Movie Fragments, as defined in Section 2.1.3 follow the DCC Header. Other boxes MAY exist between the DCC Header and the first DCC Movie Fragment. Other boxes MAY exist between DCC Movie Fragments, as well.
- A Digital CFF Container ends with a DCC Footer, as defined in Section 2.1.4. Other boxes MAY exist between the last DCC Movie Fragment and the DCC Footer. A DCC Footer MAY contain no boxes.

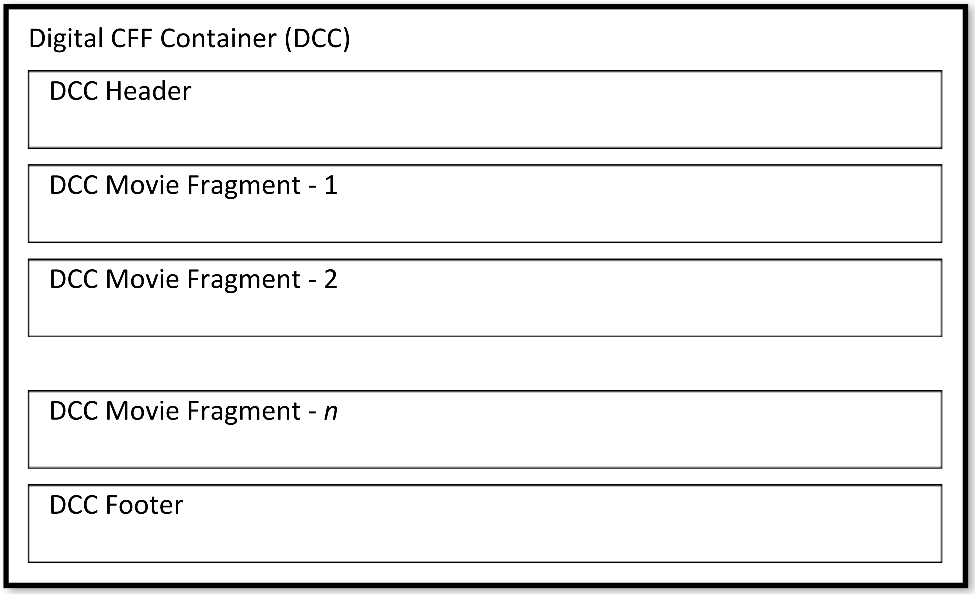


Figure 2-1 – Structure of a Digital CFF Container (DCC)

2.1.2 DCC Header

The DCC Header defines the set of boxes that appear at the beginning of a Digital CFF Container (DCC), as shown in Figure 2-2. The box requirements defined for the DCC Header SHALL apply to the start of a DCC, beginning with the first box in the DCC through to the first box of the first Movie Fragment. These boxes are defined in compliance with [ISO] with the following additional constraints and requirements:

The DCC Header SHALL start with a File Type Box (`‘ftyp’`), as defined in Section 2.3.1.

- The DCC Header SHALL include one Movie Box (`‘moov’`).
- The Movie Box SHALL contain a Movie Header Box (`‘mvhd’`), as defined in Section 2.3.2.
- The Movie Box SHALL contain a Content Information Box (`‘coin’`), as defined in Section 2.2.2. The Content Information Box (`‘coin’`) SHOULD immediately follow the Movie Header Box (`‘mvhd’`) in the DCC Header.
- The Movie Box MAY contain Multi-Track Required Metadata as specified in Section 2.1.2.1. This metadata provides content, file and track information necessary for file identification, track selection, and playback.
- The Movie Box SHALL contain media tracks as specified in Section 2.1.2.2, which defines the Track Box (`‘trak’`) requirements for the Common File Format.
- The Movie Box SHALL contain a Movie Extends Box (`‘mvex’`), as defined in Section 8.8.1 of [ISO], to indicate that the container utilizes Movie Fragment Boxes.
- The Movie Extends Box (`‘mvex’`) SHALL contain a Movie Extends Header Box (`‘mehd’`), as defined in [ISO] Section 8.8.2, to provide the overall duration of a fragmented movie.
- The Movie Box (`‘moov’`) MAY contain one or more Protection System Specific Header Boxes (`‘pssh’`), as specified in [CENC] Section 8.1.
- The Movie Box (`‘moov’`) MAY contain a Free Space Box (`‘free’`) to provide reserved space for adding DRM-specific information. If present in the DCC file, the Free Space Box (`‘free’`) SHALL be the last box in the Movie Box (`‘moov’`)
- The DCC Header MAY contain a Segment Index Box (`‘sidx’`). If present, the Segment Index Box (`‘sidx’`) SHALL appear after the Movie Box (`‘moov’`).

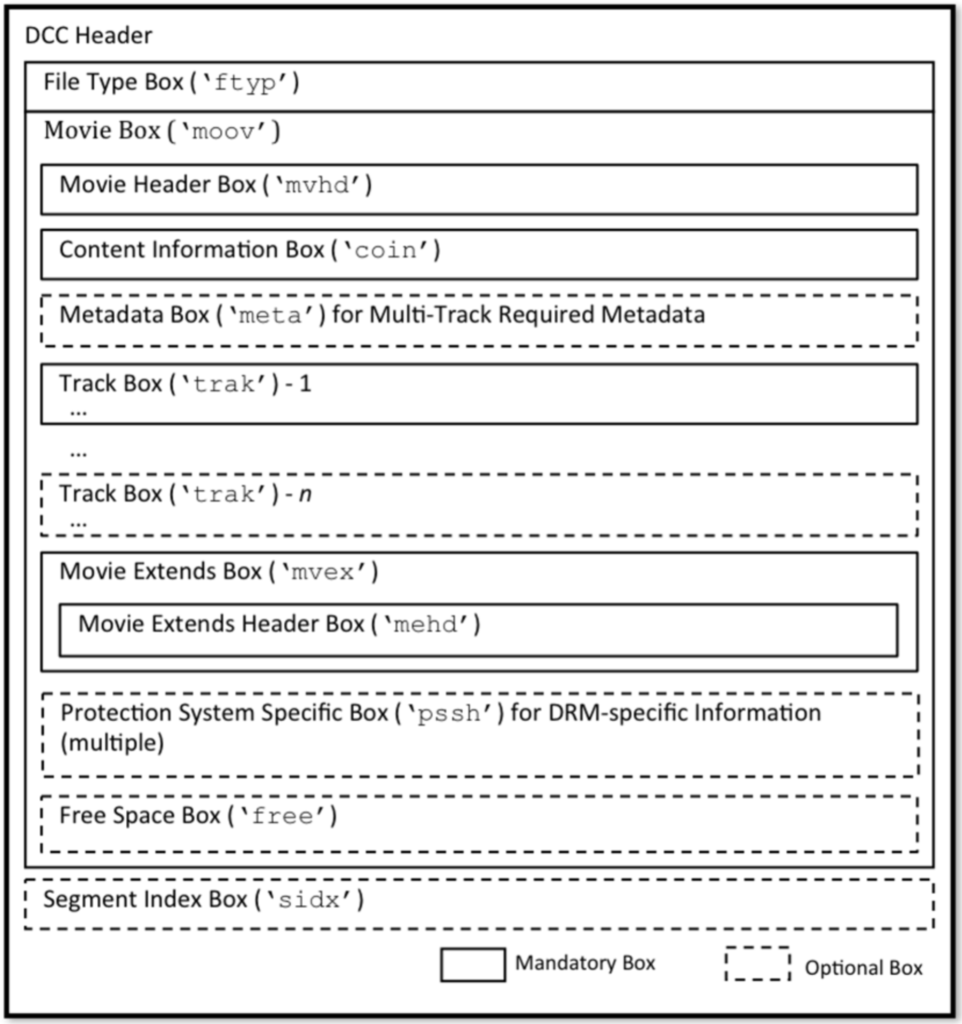

DCC Header
File Type Box ('ftyp')
Movie Box ('moov')
Movie Header Box ('mvhd')
Content Information Box ('coin')
Metadata Box ('meta') for Multi-Track Required Metadata
Track Box ('trak') - 1
...
...
Track Box ('trak') - n
...
Movie Extends Box ('mvex')
Movie Extends Header Box ('mehd')
Protection System Specific Box ('pssh') for DRM-specific Information (multiple)
Free Space Box ('free')
Segment Index Box ('sidx')
Mandatory Box
Optional Box

Figure 2-2 – Structure of a DCC Header

2.1.2.1 Required Multi-Track Metadata

The required multi-track metadata provides movie and track information, such as title, publisher, run length, release date, track types, language support, etc for multi-track Delivery Targets (see Annex C.). The required multi-track metadata is stored according to the following definition:

- A Meta Box (`'meta'`), as defined in [ISO] Section 8.11.1 MAY exist in the Movie Box. This Meta Box SHOULD precede any Track Boxes to enable faster access to the metadata it contains.
- The Meta Box SHALL contain a Handler Reference Box (`'hdlr'`) for Common File Metadata, as defined in Section 2.3.3.
- The Meta Box SHALL contain an XML Box (`'xml'`) for multi-track required metadata, as defined in Section 2.3.4.1.
- The Meta Box (`'meta'`) SHALL contain an Item Location Box (`'iloc'`) to enable XML references to images and any other binary data contained in the file, as defined in [ISO] 8.11.3.
- Images and any other binary data that are referenced by an XML document in the XML Box (`'xml'`) for multi-track required metadata SHALL be stored in one `'idat'` box which SHOULD follow all of

the boxes the ‘meta’ box contains. Each item SHALL have a corresponding entry in the ‘iloc’ box described above, the ‘iloc’construction_method field SHALL be set to ‘1’ and the ‘iloc’ extent_offset field SHALL be relative to the first byte of data[] in the ‘idat’ box containing images and any other binary data that can be referenced by an XML document in the ‘xml’ box (note: the extent_offset field in this case uses a different relative offset approach from other offset fields in other boxes).

2.1.2.2 Media Tracks

Each track of media content (i.e. audio, video, subtitles, etc.) is described by a Track Box (‘trak’) in accordance with [ISO], with the addition of the following constraints:

- Each Track Box (‘trak’) SHALL contain a Track Header Box (‘tkhd’), as defined in Section 2.3.5.
- Each Track Box (‘trak’) MAY contain an Edit Box (‘edts’) as described in Section 2.4.
- The Edit Box in the Track Box MAY contain an Edit List Box (‘elst’) as described in Section 2.4.
  - If Edit List Box (`elst’) is included, entry_count SHALL be 1, and the entry SHALL have fields set to the values described in Section 2.4.
- The track_ID associated with the media track SHALL adhere to the assignment specified in Table 2-3.
- Each Track Box (‘trak’) SHALL NOT reference media samples.
- The Media Box (‘mdia’) in a Track Box (‘trak’) SHALL contain a Media Header Box (‘mdhd’), as defined in Section 2.3.6.
- The Media Box in a Track Box (‘trak’) SHALL contain a Handler Reference Box (‘hdlr’), as defined in [ISO] Section 8.4.3.
- The Media Information Box SHALL contain a header box corresponding to the track’s media type, as follows:
  - Video tracks: Video Media Header Box (‘vmhd’), as defined in Section 2.3.8.
  - Audio tracks: Sound Media Header Box (‘smhd’), as defined in Section 2.3.9.
  - Subtitle tracks: Subtitle Media Header Box (‘sthd’), as defined in Section 2.3.9.
- The Data Information Box in the Media Information Box SHALL contain a Data Reference Box (‘dref’), as defined in Section 2.3.10.
- The Sample Table Box (‘stbl’) in the Media Information Box SHALL contain a Sample Description Box (‘stsd’), as defined in Section 2.3.11.
- For encrypted tracks, the Sample Description Box SHALL contain at least one Protection Scheme Information Box (‘sinf’), as defined in Section 2.3.14, to identify the encryption transform applied and its parameters, as well as to document the original (unencrypted) format of the media. Note: ‘sinf’ is contained in a Sample Entry with a codingname of ‘enca’ or ‘encv’ which is contained within the Sample Description Box (‘stsd’).
- The Sample Table Box SHALL contain a Decoding Time to Sample Box (‘stts’), as defined in Section 2.3.12.

- The Sample Table Box SHALL contain a Sample to Chunk Box (`‘stsc’`), as specified in Section 2.3.16, and a Chunk Offset Box (`‘stco’`), as defined in Section 2.3.17, indicating that chunks are not used.
- Additional constraints for tracks are defined corresponding to the track’s media type, as follows:
  - Video tracks: See Section 4.2 Data Structure for Video Track
  - Audio tracks: See Section 5.2 Data Structure for Audio Track.
  - Subtitle tracks: See Section 6.7 Data Structure for CFF-TT Track.

Table 2-3 – Track ID Assignment

track_ID range	Track Type
1-49	Primary Video
50-99	Secondary Video
100-999	Main Audio
1,000-1,999	Secondary Audio
2,000-9,999	Tertiary Audio
10,000-10,999	Main Subtitle
11,000-11,999	Secondary Subtitle
12,000+	Other (e.g. Metadata)

2.1.3 DCC Movie Fragment

A DCC Movie Fragment contains the metadata and media samples for a limited, but continuous sequence of homogenous content, such as audio, video or subtitles, belonging to a single track, as shown in Figure 2-3. Multiple DCC Movie Fragments containing different media types with parallel decode times are placed in close proximity to one another in the Common File Format in order to facilitate synchronous playback. The box requirements defined for DCC Movie Fragments SHALL apply to the DCC after the box requirements defined for the DCC Header.

DCC Movie Fragments SHALL comply with [ISO] Section 8.8 and the following requirements:

- The DCC Movie Fragment MAY contain one or more Event Message Boxes (`‘emsg’`).
- The DCC Movie Fragment SHALL contain a Movie Fragment Box (`‘moof’`).

The DCC Movie Fragment SHALL contain one or more Media Data Boxes (`‘mdat’`) for media samples (see Figure 2-3).

The Movie Fragment Box (`‘moof’`) in a DCC Movie Fragment SHALL comply with [ISO] Section 8.8.4 and the following requirements:

- The Movie Fragment Box SHALL contain a single Track Fragment Box (`‘traf’`) defined in [ISO] Section 8.8.6.
- All media samples in a DCC SHALL:
  - be referenced from Track Fragment Boxes (`‘traf’`) contained in Movie Fragment Boxes (`‘moof’`); and
  - only utilize effective sample parameters (`sample_index, sample_duration`, `sample_size`, and `sample_flags`) in fields and parameters located in each Movie Fragment, either in the Track Fragment Header Box (`‘tfhd’`) or Track Run Box (`‘trun’`) (see also Section 2.3.17); and
  - only be addressed using byte offsets relative to the first byte of the Movie Fragment Boxes (`‘moof’`) (see also Section 2.3.19 and Section 2.3.20).

- The Track Fragment Box (‘traf’) SHALL contain a Track Fragment Base Media Decode Time Box (‘tfdt’), as defined in [ISO] Section 8.8.12, to provide decode start time of the fragment.
- For AVC Video tracks utilizing (‘avc1’) sample entries as per Section 4.3.1.1, the Track Fragment Box (‘traf’) MAY:
  - contain a Trick Play Box (‘trik’), as defined in Section 2.2.7, to facilitate random access and trick play modes (i.e. fast forward and rewind); and
  - contain an AVC NAL Unit Storage Box (‘avcn’) as defined in Section 2.2.2. If an AVC NAL Unit Storage Box is present in any AVC Video track fragment in the DCC, one SHALL be present in all AVC video track fragments in that file.
- The Track Fragment Box (‘traf’) SHALL contain exactly one Track Fragment Run Box (‘trun’), as defined in Section 2.3.20.
- For DCC Movie Fragments which contain encrypted samples:
  - the Track Fragment Box (‘traf’) SHALL contain exactly one Sample Auxiliary Information Offsets Box (‘saio’) with an aux_info_type value of "cenc" and exactly one Sample Auxiliary Information Sizes Box (‘saiz’) with an aux_info_type value of "cenc", as specified in Section 2.2.7; and
  - the Track Fragment Box (‘traf’) SHALL contain exactly one Sample Encryption Box (‘senc’), as specified in Section 2.2.4; and
  - if the DCC Movie Fragment contains samples with different encryption keys, the DCC Movie Fragments SHALL contain a sample group and sample group description (‘seig’) as specified in [CENC].

The Media Data Box (‘mdat’) in a DCC Movie Fragment SHALL comply with Section 2.3.22 of this Specification, and the following requirements:

- The Media Data Box in the DCC Movie Fragment SHALL contain all of the media samples (i.e. audio, video or subtitles) referred to by the Track Fragment Box that falls within the same DCC Movie Fragment.
- To ensure DCC Movie Fragments containing different media types with parallel decode times are placed in close proximity to one another in a DCC, DCC Movie Fragments SHALL be ordered in sequence based on the decode time of the first sample in each DCC Movie Fragment (i.e. the movie fragment start time). When DCC Movie Fragments share the same start times, smaller size fragments SHOULD be stored first.

Note: In the case of subtitle tracks, the movie fragment start time might not equal the actual time of the first appearance of text or images in the CFF-TT document stored in the first and only sample in DCC Movie Fragment.

- Additional constraints for tracks are defined corresponding to the track’s media type, as follows:
  - Video tracks: See Section 4.2 Data Structure for Video Track.
  - Audio tracks: See Section 5.2 Data Structure for Audio Track.
    Subtitle tracks: See Section 6.7 Data Structure for CFF-TT Track.

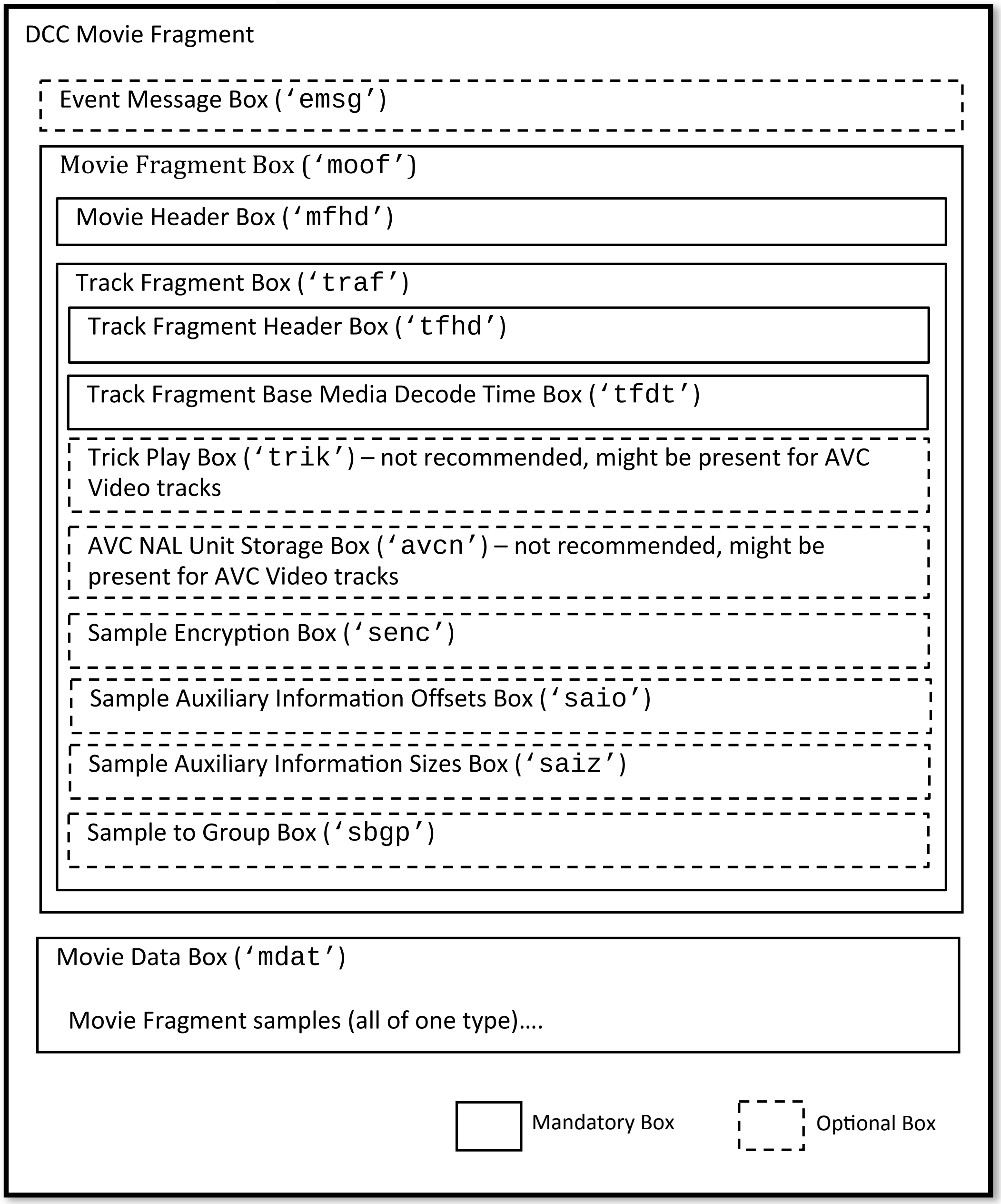

DCC Movie Fragment
Event Message Box (‘emsg’)
Movie Fragment Box (‘moof’)
Movie Header Box (‘mfhd’)
Track Fragment Box (‘traf’)
Track Fragment Header Box (‘tfhd’)
Track Fragment Base Media Decode Time Box (‘tfdt’)
Trick Play Box (‘trik’) – not recommended, might be present for AVC Video tracks
AVC NAL Unit Storage Box (‘avcn’) – not recommended, might be present for AVC Video tracks
Sample Encryption Box (‘senc’)
Sample Auxiliary Information Offsets Box (‘saio’)
Sample Auxiliary Information Sizes Box (‘saiz’)
Sample to Group Box (‘sbgp’)
Movie Data Box (‘mdat’)
Movie Fragment samples (all of one type)….
Mandatory Box
Optional Box

Figure 2-3 – DCC Movie Fragment Structure

2.1.4 DCC Footer

The DCC Footer contains Optional Multi-Track Metadata and optional information for supporting random access into the audio-visual contents of the file, as shown in Figure 2-4. The box requirements defined for DCC Footer SHALL apply to the DCC after the last box in the last DCC Movie Fragment through to the end of the DCC. These boxes are defined in compliance with [ISO] with the following additional constraints and requirements:

- The DCC Footer MAY contain a Meta Box (‘meta’), as defined in [ISO] Section 8.11.1. If the Meta Box (‘meta’), is present in the DCC Footer:

  - ➢ The Meta Box (`‘meta’`) SHALL contain a Handler Reference Box (`‘hdlr’`) for Common File Metadata, as defined in Section 2.3.3.
  - ➢ The Handler Reference Box for Common File Metadata SHALL be followed by an XML Box (`‘xml’`) for Optional Metadata, as defined in Section 2.3.4.2.
  - ➢ The Meta Box (`‘meta’`) MAY contain an Item Location Box (`‘iloc’`) to enable XML references to images and any other binary data contained in the file, as defined in [ISO] Section 8.11.3. If any such reference exists, then the Item Location Box SHALL exist.
  - ➢ Images and any other binary data referred to by the contents of the XML Box for Optional Metadata SHALL be stored in one `‘idat’` Box which SHOULD follow all of the boxes the Meta Box contains. Each item SHALL have a corresponding entry in the `‘iloc’` described above and the `‘iloc’construction_method` field SHALL be set to ‘1’.
- The DCC Footer MAY contain a Movie Fragment Random Access Box (`‘mfra’`), as defined in [ISO]Section 8.8.9. If the Movie Fragment Random Access Box (`‘mfra’`) is present in the DCC Footer:
  - ➢ Movie Fragment Random Access Box (`‘mfra’`) SHALL be the last file-level box in the DCC Footer.
  - ➢ The Movie Fragment Random Access Box (`‘mfra’`) SHALL contain one Track Fragment Random Access Box (`‘tfra’`), as defined in Section 2.3.18, for each track in the file.
  - ➢ The last box contained within the Movie Fragment Random Access Box SHALL be a Movie Fragment Random Access Offset Box (`‘mfro’`), as defined in [ISO] Section 8.8.11.

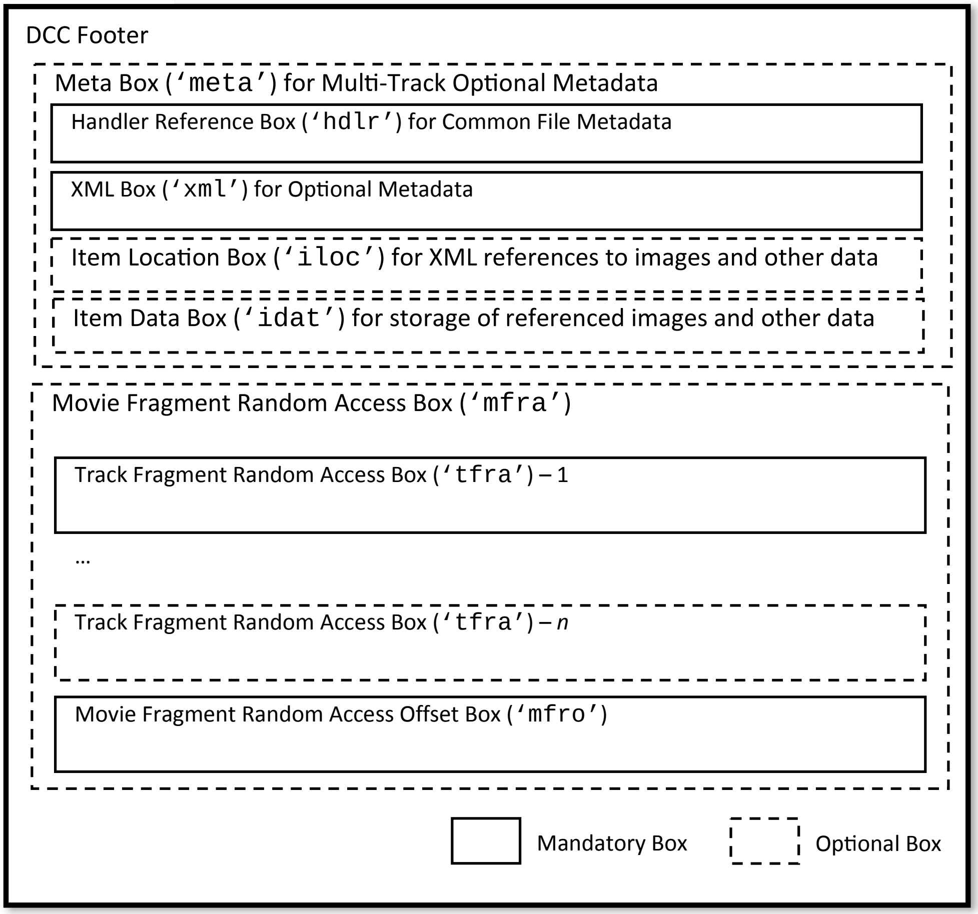

DCC Footer
Meta Box (‘meta’) for Multi-Track Optional Metadata
Handler Reference Box (‘hdlr’) for Common File Metadata
XML Box (‘xml’) for Optional Metadata
Item Location Box (‘iloc’) for XML references to images and other data
Item Data Box (‘idat’) for storage of referenced images and other data
Movie Fragment Random Access Box (‘mfra’)
Track Fragment Random Access Box (‘tfra’) – 1
…
Track Fragment Random Access Box (‘tfra’) – n
Movie Fragment Random Access Offset Box (‘mfro’)
Mandatory Box
Optional Box

Figure 2-4 – Structure of a DCC Footer

2.2 Extensions to ISO Base Media File Format

2.2.1 Standards and Conventions

2.2.1.1 Extension Box Registration

The extension boxes defined in Section 2.2 are not part of the original [ISO] specification but have been registered with [MP4RA].

2.2.1.2 Notation

To be consistent with [ISO], this section uses a class-based notation with inheritance. The classes are consistently represented as structures in the file as follows: The fields of a class appear in the file structure in the same order they are specified, and all fields in a parent class appear before fields for derived classes.

For example, an object specified as:

```
aligned(8) class Parent (
      unsigned int(32) p1_value, ..., unsigned int(32) pN_value)
{
   unsigned int(32) p1 = p1_value;
   ...
   unsigned int(32) pN = pN_value;
}

aligned(8) class Child (
      unsigned int(32) p1_value, ... , unsigned int(32) pN_value,
      unsigned int(32) c1_value, ... , unsigned int(32) cN_value)
   extends Parent (p1_value, ..., pN_value)
{
   unsigned int(32) c1 = c1_value;
   ...
   unsigned int(32) cN = cN_value;
}
```

Maps to:

```
aligned(8) struct
{
   unsigned int(32) p1 = p1_value;
   ...
   unsigned int(32) pN = pN_value;
   unsigned int(32) c1 = c1_value;
   ...
   unsigned int(32) cN = cN_value;
}
```

This section uses `string` syntax elements. These fields SHALL be encoded as a string of UTF-8 bytes as defined in [UNICODE], followed by a single null byte (0x00). When an empty string value is provided, the field SHALL be encoded as a single null byte (0x00)."

When a box contains other boxes as children, child boxes always appear after any explicitly specified fields, and can appear in any order (i.e. sibling boxes can always be re-ordered without breaking compliance to the specification).

2.2.2 Content Information Box (‘coin’)

Box Type	‘coin’
Container	Movie Box (‘moov’)
Mandatory	Yes
Quantity	One

The Content Information Box (‘coin’) contains the file metadata necessary to identify, license and play content within an Ecosystem.

2.2.2.1 Syntax

```
aligned(8) class ContentInformationBox
   extends FullBox(‘coin’, version=0, flags=0)
{



   string            mime_subtype_name;
   string            profile-level-idc;
   string            codecs;
   string            protection;
   string            languages;
   unsigned int(8)   brand_entry_count;
   for( int i=0; i < brand_entry_count; i++)
   {
      unsigned int(32)   iso_brand;
      unsigned int(32)   version
   }
   unsigned int(8)   id_entry_count;
   for( int i=0; i < id_entry_count; i++)
   {
      string   namespace;
      string   asset_id;
   }
}
```

2.2.2.2 Semantics

- `mime_subtype_name` - identifies the Media Type associated with the DCC and SHALL be set in accordance with the Media Type Template requirements defined in Annex D.
- `profile-level-idc` - SHALL be set in accordance with the `profile-level-idc` parameter requirements defined in Annex D.
- `codecs` - SHALL be set in accordance with the `codecs` parameter requirements defined in Annex D.
- `protection` - SHALL be set in accordance with the `protected` parameter requirements defined in Annex D.
- `languages` - SHALL be set in accordance with the `languages` parameter requirements defined in Annex D.
- `brand_entry_count` - SHALL be set to the number of ISO brand entries contained in the Content Information Box (`‘coin’`).
- `iso_brand` - SHALL be set to a [MP4RA] four character code associated with an ISO brand with which the DCC is compatible. One brand entry SHALL exist for each such ISO brand. All brands signaled in the File Type Box (`'ftyp'`), including both `major_brand` and `compatible_brand` values, SHALL be signaled here. The DCC SHALL conform to the requirements of any ISO brand signaled here.

- `version` - SHALL be set to the version of the specification associated with the ISO brand as defined by the relevant specification. For brands defined by [ISO], the field value will be set to the integer representation of '4' (to represent "fourth edition").
  - Any ISO brands defined in this specification SHALL be set to the integer representation of DMEDIA_VERSION_NOPOINTS (defined in Annex A.).
- `id_entry_count` – SHALL be set to the number of asset ID entries contained in the Content Information Box (`'coin'`).
- `namespace` - SHALL be set to the namespace associated with the asset identifier. The namespace indicated SHOULD be a namespace that is registered or otherwise managed e.g. for [RFC2141] the `namespace` field could be set to `"urn:<NID>"` or for [RFC4151] the `namespace` field could be `"tag:<taggingEntity>"`.
- `asset-id` - SHALL identify the content rendition within the defined namespace including the namespace. The `asset-id` field SHALL be globally unique.

2.2.3 AVC NAL Unit Storage Box (`'avcn'`)

The AVC NAL Unit Storage Box (`'avcn'`) SHOULD NOT be used. Instead sole use of the (`'avc3'`) in-band sample entries as per Section 4.3.1.1 is recommended.
The AVC NAL Unit Storage Box (`'avcn'`) SHALL NOT appear in AVC Video tracks which include the (`'avc3'`) in-band sample entries (defined in Section 4.3.1.1) and SHALL NOT appear in HEVC Video tracks.

Box Type `'avcn'`
Container Track Fragment Box (`'traf'`)
Mandatory No
Quantity Zero, or one in an AVC track fragment in a file
An AVC NAL Unit Storage Box SHALL contain an `AVCDecoderConfigurationRecord`, as defined in section 5.3.3.1 of [ISOVIDEO].

2.2.3.1 Syntax

```
aligned(8) class AVCNALBox
   extends Box('avcn')
{
   AVCDecoderConfigurationRecord()  AVCConfig;
}
```

2.2.3.2 Semantics

- `AVCConfig` – SHALL contain sufficient `sequenceParameterSetNALUnit` and `pictureParameterSetNALUnit` entries to describe the configurations of all samples referenced by the current track fragment.

Note: `AVCDecoderConfigurationRecord` contains a table of each unique Sequence Parameter Set NAL unit and Picture Parameter Set NAL unit referenced by AVC Slice NAL Units contained in samples in this track fragment. As defined in [ISOVIDEO] Section 5.2.4.1.2 semantics:

- sequenceParameterSetNALUnit contains a SPS NAL Unit, as specified in [H264]. SPSs shall occur in order of ascending parameter set identifier with gaps being allowed.
- pictureParameterSetNALUnit contains a PPS NAL Unit, as specified in [H264]. PPSs shall occur in order of ascending parameter set identifier with gaps being allowed.

2.2.4 Sample Encryption Box (‘senc’)

Box Type ‘senc’
Container Track Fragment Box (‘traf’)
Mandatory No (Yes, if track fragment is encrypted)
Quantity Zero or one

The Sample Encryption Box contains the sample specific encryption data, including the initialization vectors needed for decryption and, optionally, alternative decryption parameters. It is used when the sample data in the fragment might be encrypted.

2.2.4.1 Syntax

```
aligned(8) class SampleEncryptionBox
   extends FullBox(‘senc’, version=0, flags)
{
   unsigned int(32)  sample_count;
   {
      unsigned int(IV_size*8)  InitializationVector;
      if (flags & 0x000002)
      {
         unsigned int(16)  subsample_count;
         {
            unsigned int(16)  BytesOfClearData;
            unsigned int(32)  BytesOfEncryptedData;
         } [ subsample_count ]
      }
   }[ sample_count ]
}
```

2.2.4.2 Semantics

- flags is inherited from the FullBox structure. The SampleEncryptionBox currently supports the following bit values:
- 0x2 – UseSubSampleEncryption
    - If the UseSubSampleEncryption flag is set, then the track fragment that contains this Sample Encryption Box SHALL use the sub-sample encryption as described in Section 3.2. When this flag is set, sub-sample mapping data follows each InitilizationVector. The sub-sample mapping data consists of the number of sub-samples for each sample, followed by an array of values describing the number of bytes of clear data and the number of bytes of encrypted data for each sub-sample.

- `sample_count` is the number of encrypted samples in this track fragment. This value SHALL be either zero ('0') or the total number of samples in the track fragment.
- `InitializationVector` SHALL conform to the definition specified in [CENC] Section 9.2. Only one `IV_size` SHALL be used within a file, or zero ('0') when a sample is unencrypted. Selection of `InitializationVector` values SHOULD follow the recommendations of [CENC] Section 9.3.
  - ➢ See Section 3.2 for further details on how encryption is applied.
- `subsample_count` SHALL conform to the definition specified in [CENC] Section 9.2.
- `BytesOfClearData` SHALL conform to the definition specified in [CENC] Section 9.2.
- `BytesOfEncryptedData` SHALL conform to the definition specified in [CENC] Section 9.2.

2.2.5 Trick Play Box ('trik')

The Trick Play Box ('trik') SHOULD NOT be used.
The Trick Play Box ('trik') SHALL NOT appear in AVC Video tracks which utilize the ('avc3') in-band sample entries (defined in Section 4.3.1.1) and SHALL NOT appear in HEVC Video tracks.

Box Type	'trik'
Container	Track Fragment Box ('traf')
Mandatory	No (May be used for AVC Video tracks which include ('avc1') sample entries as per Section 4.3.1.1)
Quantity	Zero or one

This box answers three questions about AVC Video sample dependency:

1. Is this sample independently decodable (i.e. does this sample NOT depend on others)?
2. Can normal-speed playback be started from this sample with full reconstruction of all subsequent pictures in output order?
3. Can this sample be discarded without interfering with the decoding of a known set of other samples?

When performing random access (i.e. starting normal playback at a location within the track), beginning decoding at samples of picture type 1 and 2 ensures that all subsequent pictures in output order will be fully reconstructable.

Note: Pictures of type 3 (unconstrained I-picture) can be followed in output order by samples that reference pictures prior to the entry point in decoding order, preventing those pictures following the I-picture from being fully reconstructed if decoding begins at the unconstrained I-picture.

When performing "trick" mode playback, such as fast forward or reverse, it is possible to use the dependency level information to locate independently decodable samples (i.e. I-pictures), as well as pictures that can be discarded without interfering with the decoding of subsets of pictures with lower `dependency_level` values.

If the Trick Play Box ('trik') is present in an AVC Video track, it SHALL be present in the Track Fragment Box ('traf') for all AVC Video track fragments.

As this box appears in a Track Fragment Box, `sample_count` SHALL be taken from the `sample_count` in the corresponding Track Fragment Run Box ('trun').

All independently decodable samples in the video track fragment (i.e. I-frames) SHALL have a correct `pic_type` value set (value 1, 2 or 3); and all other samples SHOULD have the correct `pic_type` and dependency_level set for all pictures contained in the video track fragment.

2.2.5.1 Syntax

```
aligned(8) class TrickPlayBox
   extends FullBox('trik', version=0, flags=0)
{
   for (i=0; I < sample_count; i++) {
      unsigned int(2)  pic_type;
      unsigned int(6)  dependency_level;
   }
}
```

2.2.5.2 Semantics

- pic_type takes one of the following values for AVC video:
  - 0 – The type of this sample is unknown.
  - 1 – This sample is an IDR picture.
  - 2 – This sample is a Random Access (RA) I-picture, as defined in Section 4.2.6.
  - 3 – This sample is an unconstrained I-picture.
- dependency_level indicates the level of dependency of this sample, as follows:
  - 0x00 – The dependency level of this sample is unknown.
  - 0x01 to 0x3E – This sample does not depend on samples with a greater dependency_level values than this one.
  - 0x3F – Reserved.

2.2.6 Clear Samples within an Encrypted Track

"Encrypted tracks" MAY contain unencrypted samples. An "Encrypted track" is a track whose Sample Entry has the codingname of either 'encv' or 'enca' and has Track Encryption Box ('tenc') with IsEncrypted value of 0x1.

If samples in a DCC Movie Fragment for an "encrypted track" are not encrypted, the Track Fragment Box ('traf') of the Movie Fragment Box ('moof') in that DCC Movie Fragment SHALL contain a Sample to Group Box ('sbgp') and a Sample Group Description Box ('sgpd'). The entry in the Sample to Group Box ('sbgp') describing the unencrypted samples SHALL have a group_description_index that points to a CencSampleEncryptionInformationVideoGroupEntry or CencSampleEncryptionInformationAudioGroupEntry structure that has an IsEncrypted of '0x0' (Not encrypted) and a KID of zero (16 bytes of '0'). The CencSampleEncryptionInformationVideoGroupEntry or CencSampleEncryptionInformationAudioGroupEntry referenced by the Sample to Group Box ('sbgp') in a Track Fragment Box ('traf') SHALL be present at the referenced group description index location in the Sample Group Description Box ('sgpd') in the same Track Fragment Box ('traf').

Note: The group description indexes start at 0x10001 as specified in [ISO].

Track fragments SHALL NOT have a mix of encrypted and unencrypted samples. For clarity, this does not constrain subsample encryption as defined in [CENC] for NAL Structured Video tracks. If a track fragment is not encrypted, then the Sample Encryption Box (`‘senc’`), and related Sample Auxiliary Information Offsets Box (`‘saio’`) and Sample Auxiliary Information Sizes Box (`‘saiz’`) SHALL be omitted.

2.2.7 Storing Sample Auxiliary Information in a Sample Encryption Box

For encrypted track fragments, the Track Fragment Box (`‘traf’`) SHALL contain a Sample Auxiliary Information Offsets Box (`‘saio’`) with an `aux_info_type` value of `"cenc"` as defined in [CENC] Section 7 to provide sample-specific encryption data. This Sample Auxiliary Information Offsets Box (`‘saio’`) is constrained as follows:

- The `offset` field SHALL point to the first byte of the first initialization vector in the Sample Encryption Box (`'senc'`); and
- The `entry_count` field SHALL be 1 as the data in the Sample Encryption Box (`‘senc’`) is contiguous for all of the samples in the movie fragment (the `CencSampleAuxiliaryDataFormat` structure has the same format as the data in the Sample Encryption Box (`'senc'`), by design); and
- the `offset` field of the entry SHALL be calculated as the difference between the first byte of the containing Movie Fragment Box (`‘moof’`) and the first byte of the first `InitializationVector` in the Sample Encryption Box (assuming movie fragment relative addressing where no base data offset is provided in the track fragment header).

The size of this sample auxiliary data SHALL be specified in a Sample Auxiliary Information Sizes Box (`‘saiz’`) with an `aux_info_type` value of `"cenc"`, as defined in [CENC] Section 7. This Sample Auxiliary Information Sizes Box (`‘saiz’`) is constrained as follows:

- the sample_count field SHALL match the `sample_count` in the Sample Encryption Box (`'senc'`); and
- the `default_sample_info_size` SHALL be zero (‘0’) if the size of the per-sample information is not the same for all of the samples in the Sample Encryption Box (`'senc'`).
  Note that sample encryption information, such as initialization vectors, referenced by the Sample Auxiliary Information Offsets Box (`‘saio’`) takes precedence over sample encryption information stored in the Sample Encryption Box (`‘senc’`) - this specification defines storage in a Sample Encryption Box (`‘senc’`) in each movie fragment, but operations such as defragmentation that can occur in players or other systems rely on Sample Auxiliary Information Offsets Box (`‘saio’`) offset pointers that can refer to any storage location.

2.3 Constraints on ISO Base Media File Format Boxes

2.3.1 File Type Box (`‘ftyp’`)

Files conforming to the Common File Format SHALL include a File Type Box (`‘ftyp’`) as specified by [ISO] Section 4.3 with the following constraints:

- The `‘ccff’` ISO brand SHALL be set as a `compatible_brand` in the File Type Box (`‘ftyp’`). Note: Signaling of the `‘ccff’` brand indicates that the file fully complies with Sections 1 through to 6 of this specification.

- If the major_brand field is set to 'ccff', the minor_version field SHALL be set to the integer representation of DMEDIA_VERSION_NOPOINTS (defined in Annex A.).
- Note: compatible_brands might include additional brands that the file conforms to such as 'iso6'.

2.3.2 Movie Header Box ('mvhd')

The Movie Header Box in a DCC SHALL conform to [ISO] Section 8.2.2 with the following additional constraints:

- The value of the duration field SHALL be set to a value of zero ('0').
- The following fields SHALL be set to their default values as defined in [ISO]:
  - rate, volume and matrix.

2.3.3 Handler Reference Box ('hdlr') for Common File Metadata

The Handler Reference Box ('hdlr') for Common File Metadata SHALL conform to [ISO] Section 8.4.3 with the following additional constraints:

- The value of the handler_type field SHALL be 'cfmd', indicating the Common File Metadata handler for parsing required and optional metadata defined in Section 4 of [DMeta].
- For Multi-Track Required Metadata, the value of the name field SHOULD be "Required Metadata".
- For Multi-Track Optional Metadata, the value of the name field SHOULD be "Optional Metadata".

2.3.4 XML Box ('xml') for Common File Metadata

Two types of XML Boxes are defined in this specification. One contains Multi-Track Required Metadata, and the other contains Multi-Track Optional Metadata. Other types of XML Boxes not defined here MAY exist within a DCC.

2.3.4.1 XML Box ('xml') for Multi-Track Required Metadata

The XML Box for Multi-Track Required Metadata SHALL conform to [ISO] Section 8.11.2 with the following additional constraints:

- The xml field SHALL contain a well-formed XML document with contents that conform to Section 4.1 of [DMeta].

2.3.4.2 XML Box ('xml') for Multi-Track Optional Metadata

The XML Box for Multi-Track Optional Metadata SHALL conform to [ISO] Section 8.11.2 with the following additional constraints:

- The xml field SHALL contain a well-formed XML document with contents that conform to Section 4.2 of [DMeta].

2.3.5 Track Header Box (`‘tkhd’`)

Track Header Boxes in a DCC SHALL conform to [ISO] Section 8.3.1 with the following additional constraints:

- The value of the duration field SHALL be set to a value of zero (‘0’).
- The following fields SHALL be set to their default values as defined in [ISO]:
  - ➢ `matrix`
- The following field SHALL be set to its default value as defined in [ISO], unless specified otherwise in this specification:
  - ➢ `layer`

  Note: Section 6.7.1.1 specifies the `layer` field value for subtitle tracks.
- The `width` and `height` fields for a non-visual track (i.e. audio) SHALL be 0.
- The `width` and `height` fields for a visual track SHALL specify the track’s visual presentation size as fixed-point 16.16 values expressed in square pixels after decoder cropping parameters have been applied, without cropping of video samples in “overscan” regions of the image and after scaling has been applied to compensate for differences in video sample sizes and shapes; e.g. NTSC and PAL non-square video samples, and sub-sampling of horizontal or vertical dimensions. Track video data is normalized to these dimensions (logically) before any transformation or displacement caused by a composition system or adaptation to a particular physical display system. Track and movie matrices, if used, also operate in this uniformly scaled space.
  - ➢ Note: Section 4.2.1 specifies additional constraints for Video tracks and Section 6.7.2 specifies additional constraints for Subtitle tracks.

2.3.6 Media Header Box (`‘mdhd’`)

Media Header Boxes in a DCC SHALL conform to [ISO] Section 8.4.2 with the following additional constraints:

- The value of the `duration` field SHALL be set to a value of zero (‘0’);

  Note: The `duration` field in the Media Header Box (`‘mdhd’`) applies to the Track Box (`‘trak’`), which contains no media samples in DCCs. The duration of an entire fragmented movie can optionally be stored in the `fragment_duration` field of the Movie Extends Header Box (`‘mehd’`), which is equal to the sum of all track fragment durations in the longest track in the movie.
- The `timescale` field SHOULD contain a value that is exactly divisible by sample duration (this allows a fixed default sample duration rather than a table with different values per sample, which would be required if the timebase causes rounding or sampling errors on fixed duration audio and video samples);
- The `language` field SHOULD represent the original release language of the content.

  Note: Required Metadata (as defined in Section 2.1.2.1) provides normative language definitions for CFF.

2.3.7 Video Media Header (‘vmhd’)

Video Media Header Boxes in a DCC SHALL conform to [ISO] Section 8.4.5 with the following additional constraints:

- The following fields SHALL be set to their default values as defined in [ISO] Section 8.4.5:
  - ➢ `version=0`
  - ➢ `graphicsmode=0`
  - ➢ `opcolor={0, 0, 0}`

2.3.8 Sound Media Header (‘smhd’)

Sound Media Header Boxes in a DCC SHALL conform to [ISO] Section 8.4.5 with the following additional constraints:

- The following fields SHALL be set to their default values as defined in [ISO] Section 8.4.5:
  - ➢ `version=0`

2.3.9 Subtitle Media Header Box (‘sthd’)

Subtitle Media Header Boxes in a DCC SHALL conform to [ISO] Section 8.4.5 with the following additional constraints:

- The following fields SHALL be set to their default values as defined in [ISO] Section 8.4.5:
  - ➢ `version=0`

2.3.10Data Reference Box (‘dref’)

Data Reference Boxes in a DCC SHALL conform to [ISO] Section 8.7.2 with the following additional constraints:

- The Data Reference Box (‘dref’) SHALL contain a single entry with the `entry_flags` field set to `0x000001` (which means that the media data is in the same file as the Movie Box containing this data reference).

2.3.11Sample Description Box (‘stsd’)

Sample Description Boxes in a DCC SHALL conform to version 0 as defined in [ISO] Section 8.5.2 with the following additional constraints:

- Sample entries for encrypted tracks (those containing any encrypted sample data) SHALL encapsulate the existing sample entry with a Protection Scheme Information Box (‘sinf’) that conforms to Section 2.3.14.
- For video tracks, a visual sample entry SHALL be used. Design rules are specified in Section 4.2.2.
- For audio tracks, an audio Sample entry SHALL be used. Design rules are specified in Section 5.2.6.
- For subtitle tracks a subtitle sample entry SHALL be used. Design rules are specified in Section 6.7.1.5.

2.3.12 Protection Scheme Information Box (‘sinf’)

The DCC SHALL use Common Encryption as defined in [CENC] and follow Scheme Signaling as defined in [CENC] Section 4. The DCC MAY include more than one 'sinf' box.

2.3.13 Decoding Time to Sample Box (‘stts’)

Decoding Time to Sample Boxes in a DCC SHALL conform to [ISO] Section 8.6.1.2 with the following additional constraints:

- The entry_count field SHOULD have a value of zero (‘0’).

2.3.14 Sample to Chunk Box (‘stsc’)

Sample to Chunk Boxes in a DCC SHALL conform to [ISO] Section 8.7.4 with the following additional constraints:

- The entry_count field SHALL be set to a value of zero (‘0’).

2.3.15 Sample Size Boxes (‘stsz’ or ‘stz2’)

Both the sample_size and sample_count fields of the ‘stsz’ box SHALL be set to zero (‘0’). The sample_count field of the ‘stz2’ box SHALL be set to zero (‘0’). The actual sample size information can be found in the Track Fragment Run Box (‘trun’) for the track. Note: this is because the Movie Box (‘moov’) contains no media samples.

2.3.16 Chunk Offset Box (‘stco’)

Chunk Offset Boxes in a DCC SHALL conform to [ISO] Section 8.7.5 with the following additional constraints:

- The entry_count field SHALL be set to a value of zero (‘0’).

2.3.17 Track Extends Box (‘trex’)

Track Extends Boxes (‘trex’) in a DCC SHALL conform to [ISO] Section 8.8.3. DCCs SHALL NOT rely on default flags or parameters in the Track Extends Box (‘trex’) located in the Movie Box (‘moov’), including:

- default_sample_description_index
- default_sample_duration
- default_sample_size
- default_sample_flags

Note: Default sample parameters in the Track Extends Box (‘trex’) are stored in the DCC Header, and may not be available for each Media Segment during decoding. Values in the Track Fragment Run Box (‘trun’) override default values in the Track Fragment Header Box (‘tfhd’) which override default values in the Track Extends Box (‘trex’); so default values can be set in the Track Extends Box (‘trex’) as long as they are duplicated or overridden by values in the Track Fragment Header Box (‘tfhd’) or

Track Fragment Run Box (‘trun’) so that the effective values are stored in each movie fragment e.g. a default duration can be set for audio or video samples in the Track Extends Box (‘trex’) and every Track Fragment Header Box (‘tfhd’), and that duration can be inherited by default by all samples in Track Fragment Run Boxes (‘trun’).

2.3.18Movie Fragment Header Box (‘mfhd’)

Movie Fragment Header Boxes (‘mfhd’) in a DCC SHALL conform to [ISO] Section 8.8.5 with the following additional constraints:

- Movie Fragment Header Boxes (‘mfhd’) SHALL contain sequence_number values that are sequentially numbered starting with the number 1 and incrementing by +1, sequenced by movie fragment storage and presentation order.

2.3.19Track Fragment Header Box (‘tfhd’)

Track Fragment Header Boxes (‘tfhd’) in a DCC SHALL conform to [ISO] Section 8.8.7 with the following additional constraints:

- the base-data-offset-present flag (in the tf_flags field) SHALL be set to false in order to indicate that media samples are addressed using byte offsets relative to the first byte of the Movie Fragment Box (‘moof’); and
- the default-base-is-moof flag (in the tf_flags field) SHALL be set to true in order to indicate that the data_offset field in the Track Fragment Run Box (‘trun’) is always calculated relative to the first byte of the enclosing Movie Fragment Box (‘moof’).

2.3.20Track Fragment Run Box (‘trun’)

Track Fragment Run Boxes (‘trun’) in a DCC SHALL conform to [ISO] Section 8.8.8 with the following additional constraints:

- the version field SHALL be set to ‘1’; and
- the data-offset-present flag (in the tf_flags field) SHALL be set to true in order to indicate that the data_offset field is present and contains the byte offset from the start of this fragment’s Movie Fragment Box (‘moof’) to the first sample of media data in the following Media Data Box (‘mdat’).

2.3.21Segment Index Box (‘sidx’)

The Segment Index Box (‘sidx’) SHALL comply with [ISO] Section 8.16.3 with the following additional constraints:

- The timescale field SHALL have the same value as the timescale field in the Media Header Box (‘mdhd’) within the same track; and
- ➢the reference_ID field SHALL be set to the track_ID of the ISO Media track as defined in the Track Header Box (‘tkhd’).

2.3.22 Media Data Box (‘mdat’)

Each DCC Movie Fragment contains an instance of a Media Data box for media samples. The definition of this box complies with the Media Data Box (‘mdat’) definition in [ISO] Section 8.1.1 with the following additional constraints:

- Each instance of this box SHALL contain only media samples for a single track fragment of media content (i.e. audio, video, or subtitles from one track). In other words, all samples within an instance of this box belong to the same DCC Movie Fragment.

2.3.23 Track Fragment Random Access Box (‘tfra’)

Track Fragment Random Access Boxes in a DCC SHALL conform to [ISO] Section 8.8.10 with the following additional constraint:

- At least one entry SHALL exist for each fragment in the track that refers to the first random accessible sample in the fragment.

2.4 Inter-track Synchronization

There are two techniques available to shift decoding and composition timelines to guarantee accurate inter-track synchronization: 1) use negative composition offsets; or 2) use edit lists. These techniques are used when there is reordering of video frames, and/or misalignment of initial video and audio frame boundaries to achieve accurate inter-track synchronization. A combination of these techniques can be used; negative composition offsets to adjust for reordering of video frames without introducing presentation delay, and edit lists for an audio track to adjust for initial audio sync frame boundary misalignment. This section describes how to use these techniques.

2.4.1 Mapping media timeline to presentation timeline

Negative composition offsets in the Track Run Box (‘trun’) SHALL be used to adjust the composition time of the first presented video sample in a video track fragment to equal the decode time of the first decoded sample in that fragment. Composition offsets in subsequent samples SHALL reorder samples for continuous presentation at the intended frame rate. Note: This results in video tracks being synchronized to movie presentation time, and to audio and subtitle tracks whose sample presentation times also equal their decode times.

The Edit List Box (‘elst’) SHALL NOT be used to map the specified Media-Time in the media timeline to the start of the presentation timeline in video tracks.

2.4.2 Adjusting audio sync frame boundary misalignments

To adjust for misalignment between the start of the first audio sync frame boundary and the start of the movie timeline, an edit list timeline mapping edit entry MAY be used to define an initial offset in the audio track. This might be necessary to correct for a non-zero presentation time of the first audio sync frame - for example audio encoding began earlier and was then trimmed to the nearest sync frame to align with the start of video presentation. Unless audio and video encoders start simultaneously, audio and video frame

boundaries will usually not be aligned because they have different durations. Audio track fragments SHALL start with the first sync frame that overlaps or starts at presentation time zero.
When there is a sync frame boundary mismatch and accurate inter-track synchronization is required:

- The audio timeline mapping edit entry values are set as follows:
  `Segment-duration` = 0
  `Media-Time` = initial offset
  `Media-Rate` = 1

3 Encryption of Track Level Data

3.1 Multiple DRM Support (Informative)

Support for multiple DRM systems in the Common File Format is accomplished by using the Common Encryption mechanism defined in [CENC], along with additional methods for storing DRM-specific information. The standard encryption method utilizes AES 128-bit in Counter mode (AES-CTR) as defined in [AES]. Encryption metadata is described using track level defaults in the Track Encryption Box (`‘tenc’`) that can be overridden using sample groups. Protected tracks are signaled using the Scheme method specified in [ISO]. DRM-specific information can be stored in the new *Protection System Specific Header Box* (`‘pssh’`).

Initialization vectors are specified on a sample basis to facilitate features such as fast forward and reverse playback. Key Identifiers (KID) are used to indicate what encryption key was used to encrypt the samples in each track or fragment. Each of the Media Profiles (see Annex B.) define constraints on the number and selection of encryption keys for each track, but any fragment in an encrypted track can be unencrypted if identified as such by the `IsEncrypted` field in the fragment metadata.

By standardizing the encryption algorithm in this way, the same file can be used by multiple DRM systems, and multiple DRM systems can grant access to the same file thereby enabling playback of a single media file on multiple DRM systems. The differences between DRM systems are reduced to how they acquire the decryption key, and how they represent the usage rights associated with the file.

The data objects used by the DRM-specific methods for retrieving the decryption key and rights object or license associated with the file are stored in the Protection System Specific Header Box (`‘pssh’`) as specified in [CENC]. Players are required to be capable of parsing the files that include this DRM signaling mechanism. Any number of Protection System Specific Header Boxes (`‘pssh’`) can be contained in the Movie Box (`‘moov’`) or in the Media Package specified by [DDMP]; each box corresponding to a different DRM system. The boxes and DRM system are identified by a `SystemID`. The data objects used for retrieving the decryption key and rights object are stored in an opaque data object of variable size within the Protection System Specific Header Box. A DCC Header requires that a Free Space Box (`‘free’`), if present, be the last box in the Movie Box, following any Protection System Specific Header Boxes (`‘pssh’`) that it can contain. When DRM-specific information is added into a DCC it is required that the total size of the DRM-specific information and Free Space Box remains constant, in order to avoid changing the file size and invalidating byte offset pointers used throughout the media file.

Decryption is initiated when a device determines that the file has been protected by a stream type of `‘encv’` (encrypted video) or `‘enca’` (encrypted audio) – this is part of the ISO standard. The ISO parser examines the Scheme Information box within the Protection Scheme Information Box and determines that the track is encrypted via the DECE scheme. The parser then looks for a Protection System Specific Header Box (`‘pssh’`) that corresponds to a DRM, which it supports. A device uses the opaque data in the selected Protection System Specific Header Box to accomplish everything required by the particular DRM system to obtain a decryption key, obtain rights objects or licenses, authenticate the content, and authorize the playback system. Using the key it obtains and a key identifier in the Track Encryption Box (`‘tenc’`) or a sample group description with grouping type of `‘seig’`, which is shared by all the DRM systems, it can then decrypt audio and video samples.

3.2 Track Encryption

Encrypted track level data in a DCC SHALL use the encryption scheme defined in [CENC] Section 9. Encrypted NAL Structured Video tracks SHALL follow the scheme outlined in [CENC] Section 9.6.2, which defines a NAL unit based encryption scheme to allow access to NALs and unencrypted NAL headers in an encrypted NAL Structured Video elementary stream. All other types of tracks SHALL follow the scheme outlined in [CENC] Section 9.5, which defines a simple sample-based encryption scheme.

The following additional constraints SHALL be applied to all encrypted tracks:

- All key identifier values SHALL be a UUID conforming to [RFC4122] and binary encoded in the KID field according to [RFC4122] section 4.1.2.
- Correspondence of keys and `KID` values SHOULD be 1:1; i.e. if two tracks have the same key, then they will have the same `KID` value, and vice versa.

The following additional constraints SHALL be applied to the encryption of NAL Structured Video tracks:

- The first 96 or more bytes of each NAL Unit SHALL be left unencrypted.
  - Note: this byte range includes the NAL length and `nal_unit_type` fields in a NAL Unit.
- The `BytesOfEncryptedData` in all NAL Unit subsamples SHALL be either zero (‘0’) or a multiple of 16.

4 Video Elementary Streams

4.1 Introduction

This chapter describes the video track in relation to the ISO Base Media File and the constraints on each video format. The mapping of video sequences and parameters to samples and descriptors in a DCC is defined in Section 4.2, specifying which methods allowed in [ISO] and [ISOVIDEO] SHALL be used.

4.2 Data Structure for Video Track

Common File Format for video track SHALL comply with [ISO] and [ISOVIDEO]. In this section, the operational rules for boxes and their contents of Common File Format for video track are described.

4.2.1 Track Header Box ('tkhd')

For video tracks, the fields of the Track Header Box ('tkhd') SHALL be set to the values specified below. There are some "template" fields declared to use; see [ISO].

- flags = 0x000007, except for the case where the track belongs to an alternate group
- width and height must correspond as closely as possible to the active picture area of the video content and SHALL be set to the Normalized Display Height and Width of the encoded video.
  - ➢ The Normalized Display Height SHALL be the vertical sample count after cropping of encoded line pairs to the active image height indicated by cropping parameters.
  - ➢ The Normalized Display Width SHALL be the equivalent number of horizontal square pixels after cropping and sample aspect ratio scaling to the active image width indicated by cropping parameters.
  - ➢ Note: Normalized Display Size indicates the video track display size after decoding and compensation for sample shape and size. A player can apply additional scaling and adaptation such as cropping or padding to match images to the shape and resolution of the display currently in use.

4.2.2 Sample Description Box ('stsd')

The Sample Description Box ('stsd') in a video track SHALL include a NAL Structured Video Parameter Set that contains:

- the maximum width and height values corresponding to the maximum cropped horizontal and vertical sample counts indicated in any Sequence Parameter Set in the track; and
- a Decoder Configuration Record which:
  - ➢ signals the highest Profile, Level, and other parameters in the video track as specified in [ISOVIDEO] Section 8.3.3.1 (HEVC) and [ISOVIDEO] Section 5.3.3.1 (AVC); and
  - ➢ contains an array of initialization NAL units (SPS and PPS NALs for AVC Video, or VPS, SPS, and PPS NALs for HEVC Video) containing parameters matching the maximum values indicated in the Decoder Configuration Record.

If the Sample Description Box in a video track contains more than one visual sample entry, then the visual sample entry which satisfies the requirements above SHALL be stored as the first visual sample entry.

4.2.3 Track Fragment Run Box (‘`trun`’)

The syntax and values for Track Fragment Run Box (‘`trun`’) for video tracks SHALL conform to Section 2.3.20 with the following additional constraints:

- For samples in which presentation time stamp (PTS) and decode time stamp (DTS) differ, the `sample-composition-time-offsets-present` flag SHALL be set and corresponding values provided.
- The `data-offset-present` and `sample-size-present` flags SHALL be set and corresponding values provided.
- The `sample-duration-present` flag SHOULD be set and corresponding values provided.
- For `AVCSampleEntry` (‘`avc3`’) and `HEVCSampleEntry` (‘`hev1`’) NAL Structured Video tracks, the `'first_sample_flags'` SHALL signal the picture type of the first sample in each DCC Movie Fragment as specified below.
    - `sample_is_non_sync_sample=0`: If the first sample is a sync sample.
    - `sample_is_non_sync_sample=1`: If the first sample is not a sync sample.
    - `sample_depends_on=2`: If the first sample is an I frame.

4.2.4 Movie Fragment Box (‘`moof`’)

Movie Fragments in video tracks are required to conform to the following constrains:

- Every video track Movie Fragment except the last Movie Fragment of a video track SHALL have a duration of at least one second. The last Movie Fragment of a video track MAY have a duration of less than one second; and
- Every video track Movie Fragment SHALL have a duration no greater than 10.01 seconds.

4.2.5 Access Unit

The structure of an Access Unit for pictures in the video track SHALL comply with the data structure defined in Table 4-1.

Table 4-1 – Access Unit structure for pictures

Syntax Elements	Mandatory/Optional
Access Unit Delimiter NAL	Mandatory
Slice data	Mandatory

As specified in [ISOVIDEO], timing information provided within a video elementary stream SHOULD be ignored - rather, timing information provided at the file format level SHALL be used.

4.2.6 Random Access Points

A sync sample or AVC RA-I sample (see below) SHALL occur every 3.003 seconds or less within a NAL Structured Video Track.

For AVC elementary streams, an AVC Random Access (RA) I-picture is defined in this specification as an I-picture that is followed in output order by pictures that do not reference pictures that precede the AVC RA I-picture in decoding order, as shown in Figure 4-1 below.

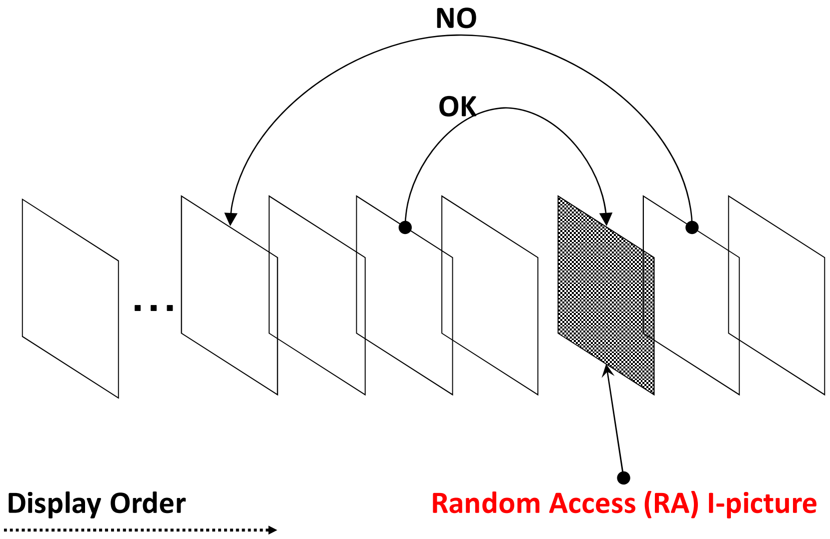


Figure 4-1 – Example of an AVC Random Access (RA) I picture

4.3 AVC

4.3.1 Storage of AVC Elementary Streams

AVC video tracks SHALL comply with Section 5 of [ISOVIDEO].

4.3.1.1 Visual Sample Entry

The syntax and values for visual sample entry:

- SHALL conform to `AVCSampleEntry (‘avc1’)` or `AVCSampleEntry (‘avc3’)` as defined in [ISOVIDEO].; and
- SHOULD conform to `AVCSampleEntry (‘avc3’)`.

If AVCSampleEntry (`‘avc3’`) is used the following requirements apply:

- If the sample is a Sync Sample, all parameter sets needed for decoding that sample SHALL be included in the sample itself.
- If the sample is a random access point that is not a Sync Sample, all parameter sets needed for decoding that sample SHALL be included in the sample itself.

4.3.1.2 AVCDecoderConfigurationRecord

Sequence Parameter Set NAL Units and Picture Parameter Set NAL Units MAY be mapped to `AVCDecoderConfigurationRecord` as specified in [ISOVIDEO] Section 5.3.3 “Decoder configuration information" and Section 5.4 "Derivation from ISO Base Media File Format", with the following additional constraints:

- All Sequence Parameter Set NAL Units mapped to `AVCDecoderConfigurationRecord` SHALL conform to the constraints defined in Section 4.3.2.2.
- All Picture Parameter Set NAL Units mapped to `AVCDecoderConfigurationRecord` SHALL conform to the constraints defined in Section 4.3.2.3.

4.3.2 Constraints on [H264] Elementary Streams

4.3.2.1 Picture type

All pictures SHALL be encoded as coded frames, and SHALL NOT be encoded as coded fields.

4.3.2.2 Sequence Parameter Sets (SPS)

Sequence Parameter Set NAL Units that occur within a DCC AVC video track SHALL conform to [H264] with the following additional constraints:

- The following fields SHALL have pre-determined values as follows:
  - ➢ `gaps_in_frame_num_value_allowed_flag` SHALL be set to 0
  - ➢ `vui_parameters_present_flag` SHALL be set to 1
- The following fields SHOULD have pre-determined values as follows:
  - ➢ `frame_mbs_only_flag` SHOULD be set to 1
- The values of the following fields SHALL NOT change throughout an [H264] elementary stream:
  - ➢ `profile_idc`
  - ➢ `level_idc`
- The values of the following fields SHOULD NOT change throughout an [H264] elementary stream:
  - ➢ `direct_8x8_inference_flag`
- If the area defined by the `width` and `height` fields of the Track Header Box of a video track (see Section 2.3.5) sub-sampled to the sample aspect ratio of the encoded picture format, does not completely fill all encoded macroblocks, then the following additional constraints apply:
  - ➢ `frame_cropping_flag` SHALL be set to 1 to indicate that AVC cropping parameters are present
  - ➢ `frame_crop_left_offset` and `frame_crop_right_offset` SHALL be set such as to crop the horizontal encoded picture to the nearest even integer width (i.e. 2, 4, 6, …) that is equal to or larger than the sub-sampled width of the track
  - ➢ `frame_crop_top_offset` and `frame_crop_bottom_offset` SHALL be set such as to crop the vertical picture to the nearest even integer height that is equal to or larger than the sub-sampled height of the track

Note: Given the definition above, for Media Profiles that support dynamic sub-sampling, if the sample aspect ratio of the encoded picture format changes within the video stream (i.e. due to a change in sub-sampling), then the values of the corresponding cropping parameters are required to change accordingly. Thus, it is possible for AVC cropping parameters to be present in one portion of an [H264] elementary stream (i.e. where cropping is necessary) and not another. As specified in [H264], when frame_cropping_flag is equal to 0, the values of frame_crop_left_offset, frame_crop_right_offset, frame_crop_top_offset, and frame_crop_bottom_offset are inferred to be equal to 0.

4.3.2.2.1 Visual Usability Information (VUI) Parameters

VUI parameters that occur within a DCC AVC video track SHALL conform to [H264] with the following additional constraints:

- The following fields SHALL have pre-determined values as follows:
  - aspect_ratio_info_present_flag SHALL be set to 1
  - chroma_loc_info_present_flag SHALL be set to 0
  - timing_info_present_flag SHALL be set to 1
  - fixed_frame_rate_flag SHALL be set to 1
  - pic_struct_present_flag SHALL be set to 1, when frame_mbs_only_flag is set to 0
- The following fields SHOULD have pre-determined values as follows:
  - pic_struct_present_flag SHOULD be set to 1
  - colour_description_present_flag SHOULD be set to 1
    Note: Per [H264], if the colour_description_present_flag is set to 1, the colour_primaries, transfer_characteristics and matrix_coefficients fields must be defined in the [H264] elementary stream
  - overscan_appropriate, if present, SHOULD be set to 0
- The values of the following fields SHALL NOT change throughout an [H264] elementary stream:
  - video_full_range_flag
  - low_delay_hrd_flagcolour_primaries, when present
  - transfer_characteristics, when present
  - matrix_coefficients, when present
- The condition of the following SHALL NOT change throughout an [H264] elementary stream:
  - time_scale/num_units_in_tick/2
    Note: The requirement that fixed_frame_rate_flag be set to 1 and the condition that time_scale/num_units_in_tick/2 does not change throughout a stream ensures a fixed frame rate throughout the [H264] elementary stream.
- The values of the following fields SHOULD NOT change throughout an [H264] elementary stream:
  - max_dec_frame_buffering,
  - colour_description_present_flag
  - overscan_info_present_flag
  - overscan_appropriate

4.3.2.3 Picture Parameter Sets (PPS)

Picture Parameter Set NAL Units that occur within a DCC SHALL conform to [H264] with the following additional constraints:

- The condition of the following fields SHALL NOT change throughout an [H264] elementary stream:
  - entropy_coding_mode_flag

4.3.2.4 Maximum Bitrate

The maximum bitrate of [H264] elementary streams SHALL be calculated by implementation of the buffer and timing model defined in [H264] Annex C.

4.3.2.5 Frame rate

The frame rate of [H264] elementary streams SHALL be calculated as follows:

- `frame rate = time_scale ÷ (2 * num_units_in_tick)`
  Note: `time_scale` and `num_units_in_tick` are [H264] coding parameters. Based on the restrictions defined in Section 4.3.2.2.1, this equation applies to all [H264] elementary conforming to this Specification.

4.4 HEVC

4.4.1 Storage of HEVC Elementary Streams

HEVC video tracks SHALL comply with Section 8 of [ISOVIDEO].

4.4.1.1 Visual Sample Entry

The syntax and values for visual sample entry:

- SHALL conform to `HEVCSampleEntry` (`'hev1'`) sample entries as defined in [ISOVIDEO].

The following requirements apply to `HEVCSampleEntry` (`'hev1'`):

- If the sample is a Sync Sample, all parameter sets needed for decoding that sample SHALL be included in the sample itself.
- If the sample is a random access point that is not a Sync Sample, all parameter sets needed for decoding that sample SHALL be included in the sample itself.

4.4.1.2 HEVCDecoderConfigurationRecord

Video Parameter Set NAL Units (`NAL_unit_type` = `"VPS_NUT"`), Sequence Parameter Set NAL Units (`NAL_unit_type` = `"SPS_NUT"`) and Picture Parameter Set NAL Units (`NAL_unit_type` = `"PPS_NUT"`) MAY be mapped to `HEVCDecoderConfigurationRecord` as specified in [ISOVIDEO] Section 8.3.3 "Decoder configuration information" and Section 8.4 "Derivation from ISO Base Media File Format", with the following additional constraints:

- All Video Parameter Set NAL Units mapped to `HEVCDecoderConfigurationRecord` SHALL conform to the constraints defined in Section 4.4.2.2.
- All Sequence Parameter Set NAL Units mapped to `HEVCDecoderConfigurationRecord` SHALL conform to the constraints defined in Section 4.4.2.3.

4.4.2 Constraints on [H265] Elementary Streams

4.4.2.1 Picture type

All pictures SHALL be encoded as coded frames, and SHALL NOT be encoded as coded fields.

4.4.2.2 Video Parameter Sets (VPS)

Video Parameter Set NAL Units that occur within a DCC HEVC track SHALL conform to [H265] with the following additional constraints:

- The following fields SHALL have pre-determined values as follows:
  - ➢ `fixed_pic_rate_general_flag` SHOULD be set to 1
  - ➢ `general_interlaced_source_flag` SHALL be set to 0
- The condition of the following fields SHALL NOT change throughout an [H265] elementary stream:
  - ➢ `general_profile_space`
  - ➢ `general_profile_idc`
  - ➢ `general_tier_flag`
  - ➢ `general_level_idc`

4.4.2.3 Sequence Parameter Sets (SPS)

Sequence Parameter Set NAL Units that occur within a DCC HEVC track SHALL conform to [H265] with the following additional constraints:

- The following fields SHALL have pre-determined values as follows:
  - ➢ `general_frame_only_constraint_flag` SHOULD be set to 1
  - ➢ `vui_parameters_present_flag` SHALL be set to 1
- If the area defined by the `width` and `height` fields of the Track Header Box of a video track (see Section 2.3.5) sub-sampled to the sample aspect ratio of the encoded picture format does not completely fill all encoded coding tree units, the following additional constraints SHALL apply:
  - ➢ `conformance_window_flag` SHALL be set to 1 to indicate that HEVC cropping parameters are present
  - ➢ `conf_win_left_offset` and `conf_win_right_offset` SHALL be set such as to crop the horizontal encoded picture to the nearest even integer width (i.e. 2, 4, 6, …) that is equal to or larger than the sub-sampled width of the track
  - ➢ `conf_win_top_offset` and `conf_win_bottom_offset` SHALL be set such as to crop the vertical picture to the nearest even integer height that is equal to or larger than the sub-sampled height of the track

Note: Given the definition above, for Media Profiles that support dynamic sub-sampling, if the sample aspect ratio of the encoded picture format changes within the video stream (i.e. due to a change in sub-sampling), then the values of the corresponding cropping parameters are required to change accordingly. Thus, it is possible for HEVC cropping parameters to be present in one portion of a [H265] elementary stream (i.e. where cropping is necessary) and not another. As specified in [H265], when `conformance_window_flag` is equal to 0, the values of `conf_win_left_offset`, `conf_win_right`, `conf_win_top_offset`, and `conf_win_bottom_offset` are inferred to be equal to 0.

4.4.2.3.1 Visual Usability Information (VUI) Parameters

VUI parameters that occur within a DCC HEVC track SHALL conform to [H265] with the following additional constraints:

- The following fields SHALL have pre-determined values as defined:
    - ➢ `aspect_ratio_info_present_flag` SHALL be set to 1
    - ➢ `chroma_loc_info_present_flag` SHALL be set to 0
    - ➢ `vui_hrd_parameters_present_flag` SHALL be set to 1
    - ➢ `vui_timing_info_present_flag` SHALL be set to 1
    - ➢ `nal_hrd_parameters_present_flag` SHALL be set to 1
    - ➢ The following fields SHOULD have pre-determined values as follows:
    - ➢ `colour_description_present_flag` SHOULD be set to 1.
      Note: Per [H265], if the `colour_description_present_flag` is set to 1, the `colour_primaries`, `transfer_characteristics` and `matrix_coefficients` fields must be defined in the [H265] elementary stream.
    - ➢ `overscan_appropriate`, if present, SHOULD be set to 0.
- The values of the following fields SHALL NOT change throughout an [H265] elementary stream:
    - ➢ `video_full_range_flag`
    - ➢ `low_delay_hrd_flag`
    - ➢ `colour_primaries`, when present
    - ➢ `transfer_characteristics`, when present
    - ➢ `matrix_coeffs`, when present
    - ➢ `vui_time_scale`
    - ➢ `vui_num_units_in_tick`
      Note: The requirement that `fixed_pic_rate_general` be set to 1 and the values of `vui_num_units_in_tick` and `vui_time_scale` not change throughout a stream ensures a fixed frame rate throughout the [H265] elementary stream.
- The values of the following fields SHOULD NOT change throughout an [H265] elementary stream:
    - ➢ `colour_description_present_flag`
    - ➢ `overscan_info_present_flag`
    - ➢ `overscan_appropriate`

4.4.2.4 Maximum Delay

The (maximum) delay is the ratio of the CPB size / max-rate. The delay SHALL be less than or equal to 8 seconds

4.4.2.5 Maximum Bitrate

The maximum bitrate of [H265] elementary streams SHALL be calculated by implementation of the buffer and timing model defined in [H265] Annex C.

4.4.2.6 Frame Rate

The frame rate of [H265] elementary streams SHALL be calculated as follows:

- `frame rate = vui_time_scale ÷ vui_num_units_in_tick`
  Note: `viu_time_scale` and `vui_num_units_in_tick` are [H265] coding parameters. Based on the restrictions defined in Section 4.4.2.3.1, this equation applies to all [H264] elementary conforming to this Specification.

4.5 Sub-sampling and Cropping

In order to promote the efficient encoding and display of video content, cropping and sub-sampling is supported. However, the extent to which each is supported is specified in each Media Profile definition (see the Media Profile Annexes of this specification).

4.5.1 Sub-sampling

Spatial sub-sampling can be a helpful tool for improving coding efficiency of a video elementary stream. It is achieved by reducing the resolution of the coded picture relative to the source picture, while adjusting the sample aspect ratio to compensate for the change in presentation. For example, by reducing the horizontal resolution of the coded picture by 50% while increasing the sample aspect ratio from 1:1 to 2:1, the coded picture size is reduced by half. While this does not necessarily correspond to a 50% decrease in the amount of coded picture data, the decrease can nonetheless be significant.

The extent to which a Coded Video Sequence is sub-sampled is primarily specified by the combination of the following sequence parameter set fields in the video elementary stream:

- [H264]:
  - `pic_width_in_mbs_minus1` which defines the number of horizontal samples
  - `pic_height_in_map_units_minus1`, which defines the number of vertical samples
  - `aspect_ratio_idc`, which defines the aspect ratio of each sample
- [H265]:
  - `pic_width_in_luma_samples` which defines the number of horizontal samples
  - `pic_height_in_luma_samples` ([H265]) which defines the number of vertical samples
  - `aspect_ratio_idc`, which defines the aspect ratio of each sample

The display dimensions of a video track are defined in terms of square pixels (i.e. 1:1 sample aspect ratio) in the `width` and `height` fields of the Track Header Box (`‘tkhd’`) of the video track (see Section 2.3.5 and Section 4.2.1). These values are used to determine the appropriate processing to apply when displaying the content.

Each Delivery Target in this specification (see Annex C) defines constraints on the amount and nature of spatial sub-sampling that is permitted by the Delivery Target.

4.5.1.1 Sub-sample Factor

For the purpose of this specification, the extent of sub-sampling applied is characterized by a *sub-sample factor* in each of the horizontal and vertical dimensions, defined as follows:

- The *horizontal sub-sample factor* is defined as the ratio of the number of columns of the *luma* sample array in a full encoded frame absent of cropping over the number of columns of the *luma* sample array in a picture format’s frame as specified with SAR 1:1.

- The *vertical sub-sample factor* is defined as the ratio of the number of rows of the *luma* sample array in a full encoded frame absent of cropping over the number of rows of the *luma* sample array in a picture format's frame as specified with SAR 1:1.

The sub-sample factor is specifically used for selecting appropriate `width` and `height` values for the Track Header Box for video tracks, as specified in Section 2.3.5. The Media Profile definitions in the Annexes of this document specify the picture formats and the corresponding sub-sample factors and sample aspect ratios of the encoded picture that are supported for each Profile.

4.5.1.1.1 Examples of Single Dimension Sub-sampling

If a 1920 x 1080 square pixel (SAR 1:1) source picture is horizontally sub-sampled and encoded at a resolution of 1440 x 1080 (SAR 4:3), which corresponds to a 1920 x 1080 square pixel (SAR 1:1) picture format, then the horizontal sub-sample factor is 1440 ÷ 1920 = 0.75, while the vertical sub-sample factor is 1.0 since there is no change in the vertical dimension.

Similarly, if a 1280 x 720 (SAR 1:1) source picture is vertically sub-sampled and encoded at a resolution of 1280 x 540 (SAR 3:4), which corresponds to a 1280 x 720 (SAR 1:1) picture format frame size, then the horizontal sub-sample factor is 1.0 since the is no change in the horizontal dimension, and the vertical sub-sample factor is 540 ÷ 720 = 0.75.

4.5.1.1.2 Example of Mixed Sub-sampling

If a 1280 x 1080 (SAR 3:2) source picture is vertically sub-sampled and encoded at a resolution of 1280 x 540 (SAR 3:4), corresponding to a 1920 x 1080 square pixel (SAR 1:1) picture format frame size, then the horizontal sub-sample factor is 1280 ÷ 1920 = $^2/_3$, and the vertical sub-sample factor is 540 ÷ 1080 = 0.5. To understand how this is an example of mixed sub-sampling, it is helpful to remember that the initial source picture resolution of 1280 x 1080 (SAR 3:2) can itself be thought of as having been horizontally sub-sampled from a higher resolution picture.

4.5.2 Cropping to Active Picture Area

Another helpful tool for improving coding efficiency in a video elementary stream is the use of cropping. This specification defines a set of rules for defining encoding parameters to reduce or eliminate the need to encode non-essential picture data such as black matting (i.e. "letterboxing" or "black padding") that fall outside of the active picture area of the original source content.

The dimensions of the active picture area of a video track are specified by the `width` and `height` fields of the Track Header Box (`'tkhd'`), as described in Section 2.3.5 and Section 4.2.1. These values are specified in square pixels, and track video data is normalized to these dimensions before any transformation or displacement caused by a composition system or adaptation to a particular physical display system.

When sub-sampling is applied, as described above, the number of coded samples is scaled in one or both dimensions. However, since the sub-sampled picture area might not always fall exactly on the sample coding unit boundary employed by the video elementary stream, additional cropping parameters are used to further define the dimensions of the coded picture.

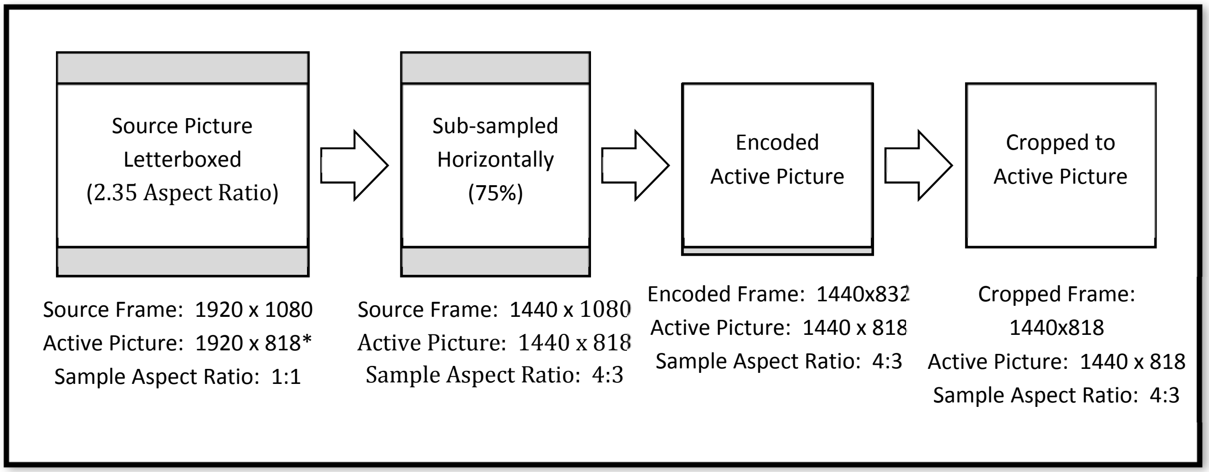
- [H264]:
  - "Macroblocks" define the sample coding unit boundary (and are 16x16 blocks)
  - See the process defined in Section 4.3.2.2

- [H265]:
  - “Coding Tree Units” define the sample coding unit boundary (and are 64×64, 32×32, or 16×16 blocks)
  - See the process defined in Section 4.4.2.3

4.5.3 Relationship of Cropping and Sub-sampling

When spatial sub-sampling is applied, additional cropping parameters are often needed to compensate for the mismatch between the coded picture size and the macroblock ([H264]) / coding tree unit ([H265]) boundaries. The specific relationship between theses mechanisms is defined as follows:

- Each picture is decoded using the coding parameters, including decoded picture size and cropping fields, defined in the sequence parameter set corresponding to that picture’s Coded Video Sequence.
- The dimensions defined by the `width` and `height` fields in the Track Header Box are used to determine which, if any, scaling or other composition operations are necessary for display. For example, to output the video to an HDTV, the decoded image might need to be scaled to the resolution defined by `width` and `height` and then additional matting applied (if necessary) in order to form a valid television video signal.



* AVC cropping can only operate on even numbers of lines, requiring that the selected height be rounded up

Figure 4-2 – Example of Encoding Process of Letterboxed Source Content

Figure 4-1 shows an example of the encoding process that can be applied. Table 4-2 shows the parameter values that could be used.

Table 4-2 – Example Sub-sample and Cropping Values for Figure 4-1

Object	Field	Value
Picture Format	width	1920
Frame Size	height	1080
Sub-sample Factor	horizontal	0.75
	vertical	1.0
Track Header Box	width	1920
	height	818
[H264] Parameter Values	chroma_format_idc	1 (4:2:0)
	aspect_ratio_idc	14 (4:3)

	pic_width_in_mbs_minus1	89
	pic_height_in_map_units_minus1	51
	frame_cropping_flag	1
	frame_crop_left_offset	0
	frame_crop_right_offset	0
	frame_crop_top_offset	0
	frame_crop_bottom_offset	7
[H265] Parameter Values	chroma_format_idc	1 (4:2:0)
	MinCbSizeY	16
	log2_min_luma_coding_block_size_minus3	1
	aspect_ratio_idc	14 (4:3)
	pic_width_in_luma_samples	1440
	pic_height_in_luma_samples	832
	conformance_window_flag	1
	conf_win_left_offset	0
	conf_win_right_offset	0
	conf_win_top_offset	0
	conf_win_bottom_offset	7

Notes:

- as `chroma_format_idc` is 1, `SubWidthC` and `SubWidthC` are set to 2 per [H264] and [H265]. This results in a doubling of frame crop parameters (so `frame_crop_bottom_offset` and `conf_win_bottom_offset` both equate to 14 pixels in the above example).
- As [H265] `MinCbSizeY` is 16 and `log2_min_luma_coding_block_size_minus3` is 1, the Coding Tree Unit size is 16x16 (matching the [H264] macroblock size of 16x16).

The decoding and display process for this content is illustrated in Figure 4-2, below. In this example, the decoded picture dimensions are 1440 x 818, one line larger than the original active picture area. This is due to a limitation in the cropping parameters to crop only even pairs of lines.

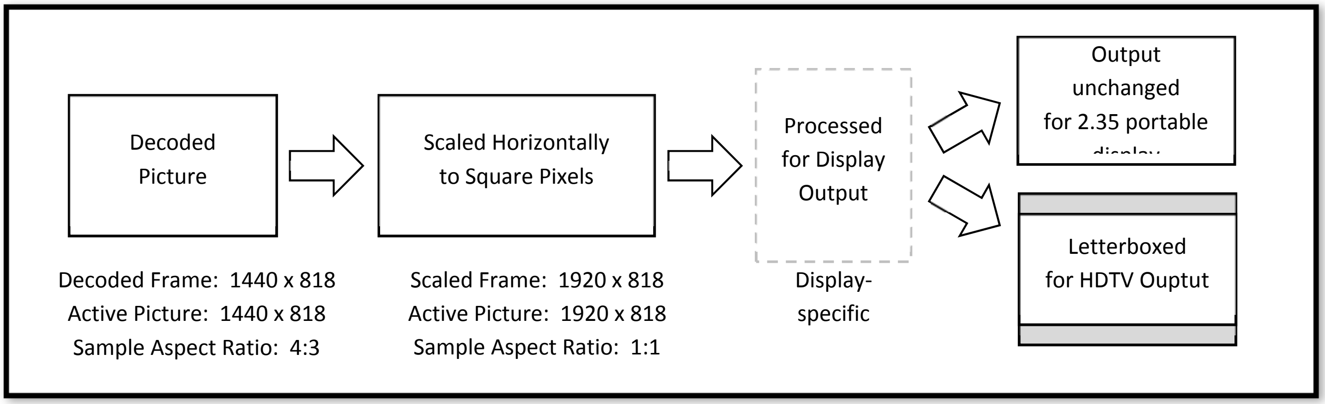

Decoded Picture
Decoded Frame: 1440 x 818
Active Picture: 1440 x 818
Sample Aspect Ratio: 4:3
Scaled Horizontally to Square Pixels
Scaled Frame: 1920 x 818
Active Picture: 1920 x 818
Sample Aspect Ratio: 1:1
Processed for Display Output
Display-specific
Output unchanged for 2.35 portable
Letterboxed for HDTV Ouptut

Figure 4-3 – Example of Display Process for Letterboxed Source Content

Figure 4-3, below, illustrates what might happen when both sub-sampling and cropping are working in the same horizontal dimension. The original source picture content is first sub-sampled horizontally from a 1:1 sample aspect ratio at 1920 x 1080 to a sample aspect ratio of 4:3 at 1440 x 1080, then the 1080 x 1080 pixel active picture area of the sub-sampled image is encoded. However, the actual coded picture has a resolution of 1088 x 1088 pixels due to the coding unit boundaries falling on even multiples of 16 pixels in this example - therefore, additional cropping parameters must be provided in both horizontal and vertical dimensions.

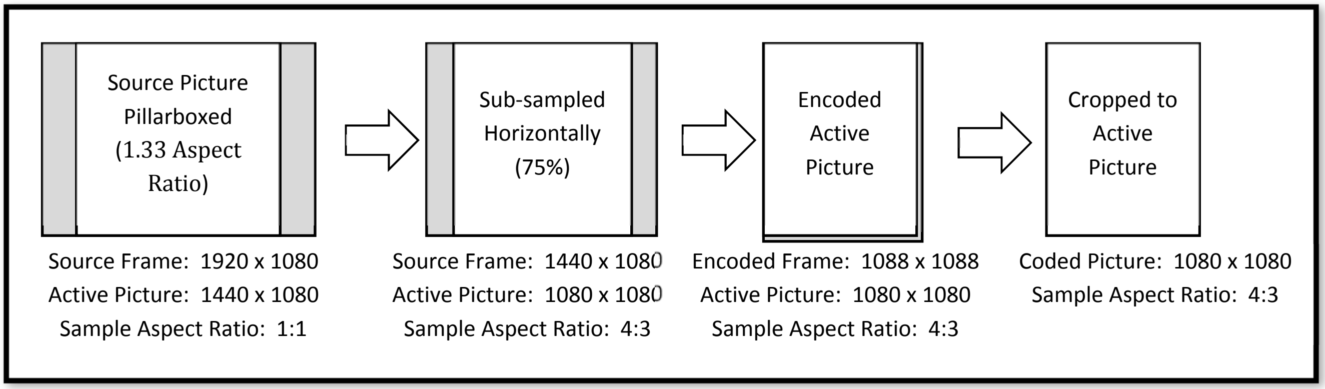

Source Picture Pillarboxed (1.33 Aspect Ratio)
Source Frame: 1920 x 1080
Active Picture: 1440 x 1080
Sample Aspect Ratio: 1:1
Sub-sampled Horizontally (75%)
Source Frame: 1440 x 1080
Active Picture: 1080 x 1080
Sample Aspect Ratio: 4:3
Encoded Active Picture
Encoded Frame: 1088 x 1088
Active Picture: 1080 x 1080
Sample Aspect Ratio: 4:3
Cropped to Active Picture
Coded Picture: 1080 x 1080
Sample Aspect Ratio: 4:3

Figure 4-4 – Example of Encoding Process for Pillarboxed Source Content

Table 4-3 lists the various parameters that might appear in the resulting file for this sample content.

Table 4-3 – Example Sub-sample and Cropping Values for Figure 4-3

Object	Field	Value
Picture Format	width	1920
Frame Size	height	1080
Sub-sample Factor	horizontal	0.75
	vertical	1.0
Track Header Box	width	1440
	height	1080
[H264] Parameter Values	chroma_format_idc	1 (4:2:0)
	aspect_ratio_idc	14 (4:3)
	pic_width_in_mbs_minus1	67
	pic_height_in_map_units_minus1	67
	frame_cropping_flag	1
	frame_crop_left_offset	0
	frame_crop_right_offset	4
	frame_crop_top_offset	0
	frame_crop_bottom_offset	4
[H265] Parameter Values	chroma_format_idc	1 (4:2:0)
	MinCbSizeY	16
	log2_min_luma_coding_block_size_minus3	1
	aspect_ratio_idc	14 (4:3)
	pic_width_in_luma_samples	1088
	pic_height_in_luma_samples	1088
	conformance_window_flag	1
	conf_win_left_offset	0
	conf_win_right_offset	4
	conf_win_top_offset	0
	conf_win_bottom_offset	4

Notes:

- as `chroma_format_idc` is 1, `SubWidthC` and `SubWidthC` are set to 2 per [H264] and [H265]. This results in a doubling of frame crop parameters (so `frame_crop_bottom_offset` and `conf_win_bottom_offset` both equate to 14 pixels in the above example).
- As [H265] `MinCbSizeY` is 16 and `log2_min_luma_coding_block_size_minus3` is 1, the Coding Tree Unit size is 16x16 (matching the [H264] macroblock size of 16x16).

The process for reconstructing the video for display is shown in Figure 4-4. As in the previous example, the decoded picture is required to be scaled back up to the original 1:1 sample aspect ratio.

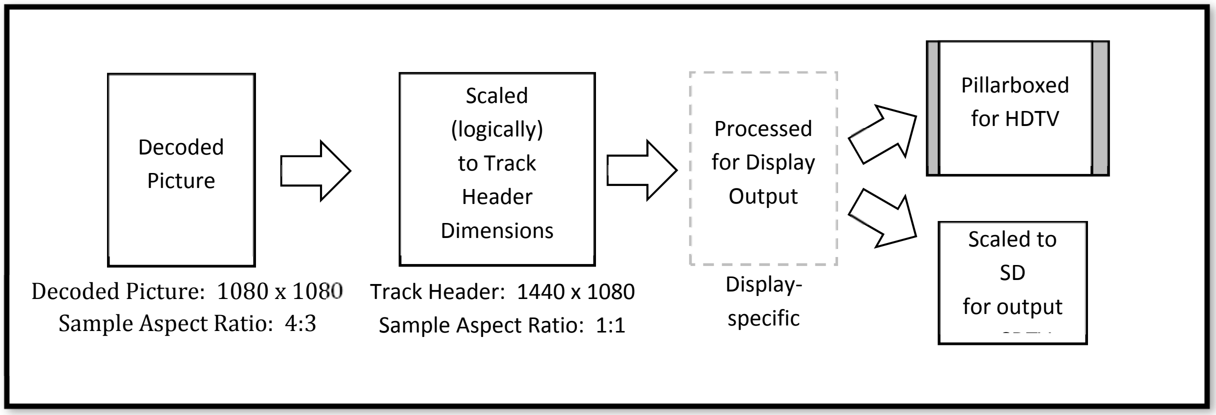

Decoded Picture
Decoded Picture: 1080 x 1080
Sample Aspect Ratio: 4:3
Scaled (logically) to Track Header Dimensions
Track Header: 1440 x 1080
Sample Aspect Ratio: 1:1
Processed for Display Output
Display-specific
Pillarboxed for HDTV
Scaled to SD for output

Figure 4-5 – Example of Display Process for Pillarboxed Source Content

If this content was to be displayed on a standard 4:3 television, no further processing of the image would be necessary. However, if this content was to be displayed on a 16:9 HDTV, it might be necessary for it to apply additional matting on the left and right sides to reconstruct the original pillarboxes in order to ensure the video image displays properly.

4.5.4 Dynamic Sub-sampling

For Media Profiles that support dynamic sub-sampling, the spatial sub-sampling of the content can be changed periodically throughout the duration of the file. Changes to the sub-sampling values are implemented by changing the elementary stream parameter values identified in Section 4.5.1. Dynamic sub-sampling is supported by Media Profiles that do not specifically prohibit these values from changing within a video track.

4.5.4.1 Constraints on [H264] Elementary Streams

- the `pic_width_in_mbs_minus1`, `pic_height_in_map_units_minus1` and `aspect_ratio_idc` sequence parameter set field values SHALL only be changed at the start of a fragment.
- When sub-sampling parameters are changed within the file, the `frame_cropping_flag`, `frame_crop_left_offset`, `frame_crop_right_offset`, `frame_crop_top_offset`, `frame_crop_bottom_offset` cropping parameters SHALL also be changed to match, as specified in Section 4.3.2.2.
- Note: If `pic_width_in_mbs_minus1` or `pic_height_in_map_units_minus1` changes from the previous Coded Video Sequence, this SHALL NOT imply `no_output_of_prior_pics_flag` is equal to one – in this case video presentation and output of all video frames SHOULD continue without interruption in presentation, i.e. no pictures SHOULD be discarded.

4.5.4.2 Constraints on [H265] Elementary Streams

- the `pic_width_in_luma_samples`, `pic_height_in_luma_samples` and `aspect_ratio_idc` sequence parameter set field values SHALL only be changed at the start of a fragment.

- When sub-sampling parameters are changed within the file, the `conformance_window_flag`, `conf_win_left_offset, conf_win_right_offset, conf_win_top_offset, conf_win_bottom_offset` cropping parameters SHALL also be changed to match, as specified in Section 4.4.2.3.
- Note: If `pic_width_in_luma_samples` or `pic_height_in_luma_samples` changes from the previous Coded Video Sequence, this SHALL NOT imply `no_output_of_prior_pics_flag` is equal to one – in this case video presentation and output of all video frames SHOULD continue without interruption in presentation, i.e. no pictures SHOULD be discarded.

5 Audio Elementary Streams

5.1 Introduction

This chapter describes the audio track in relation to the ISO Base Media File, the required vs. optional audio formats and the constraints on each audio format.
In general, the system layer definition described in [MPEG4S] is used to embed the audio. This is described in detail in Section 5.2.

5.2 Data Structure for Audio Track

The common data structure for storing audio tracks in a DCC is described here. All required and optional audio formats comply with these conventions.

5.2.1 Track Header Box (‘tkhd’)

For audio tracks, the fields of the Track Header Box SHALL be set to the values specified below. There are some “template” fields declared to use; see [ISO].

- flags = 0x000007, except for the case where the track belongs to an alternate group
- layer = 0
- volume = 0x0100
- matrix = {0x00010000, 0, 0, 0, 0x00010000, 0, 0, 0, 0x40000000}
- width = 0
- height = 0

5.2.2 Movie Fragment Box (‘moof’)

Movie Fragments in audio tracks are required to conform to the following constrains:

- Every audio track Movie Fragment except the last Movie Fragment of an audio track SHALL have a duration of at least one second. The last Movie Fragment of an audio track MAY have a duration of less than one second; and
- Every audio track Movie Fragment SHALL have a duration no greater than 6 seconds.

5.2.3 Sync Sample Box (‘stss’)

The Sync Sample Box (‘stss’) SHALL NOT be used.
Note: “sync sample” in movie fragments cannot be signaled by the absence of the Sync Sample box (‘stss’) or by the presence of the Sync Sample box (‘stss’), since this box is not designed to list sync samples in movie fragments.

- For audio formats in which every audio access unit is a random access point (sync sample), signaling can be achieved by other means such as setting the 'sample_is_non_sync_sample' flag to "0" in the 'default_sample_flags' field in the Track Extends box (‘trex’).

- For audio formats in which some audio access units are not sync samples, sync samples can be signaled using `sample_flags` in the Track Run box (`'trun'`).

5.2.4 Handler Reference Box (`'hdlr'`)

The syntax and values for the Handler Reference Box (`'hdlr'`) for audio tracks SHALL conform to [ISO] with the following additional constraints:

- The `handler_type` field SHALL be set to `"soun"`

5.2.5 Sound Media Header Box (`'smhd'`)

The syntax and values for the Sound Media Header Box SHALL conform to [ISO] with the following additional constraints:

- The following fields SHALL be set as defined:
  - `balance` = 0

5.2.6 Sample Description Box (`'stsd'`)

The contents of the Sample Description Box (`'stsd'`) are determined by value of the `handler_type` parameter in the Handler Reference Box (`'hdlr'`). For audio tracks, the `handler_type` parameter is set to "soun", and the Sample Description Box contains a audio sample entry that describes the configuration of the audio track.

For each of the audio formats supported by the Common File Format, a specific audio sample entry box that is derived from the `AudioSampleEntry` box defined in [ISO] is used. Each codec-specific `SampleEntry` box is identified by a unique `codingname` value, and specifies the audio format used to encode the audio track, and describes the configuration of the audio elementary stream. Table 5-1 lists the audio formats that are supported by the Common File Format, and the corresponding `SampleEntry` that is present in the Sample Description Box for each format.

Table 5-1 – Defined Audio Formats

codingname	Audio Format	SampleEntry Type	Section Reference
mp4a	MPEG-4 AAC [2-channel]	MP4AudioSampleEntry	Section 5.3.2
	MPEG-4 HE AAC V2 [5.1, 7.1-channel]		Section 5.3.3
	MPEG-4 HE AAC v2		Section 5.3.4
	MPEG-4 HE AAC v2 with MPEG Surround		Section 5.3.5
ac-3	AC-3 (Dolby Digital)	AC3SampleEntry	Section 5.5.1
ec-3	Enhanced AC-3 (Dolby Digital Plus)	EC3SampleEntry	Section 5.5.2
mlpa	MLP	MLPSampleEntry	Section 5.5.3
dtsc	DTS	DTSSampleEntry	Section 5.6
dtsh	DTS-HD with core substream	DTSSampleEntry	Section 5.6
dtsl	DTS-HD Master Audio	DTSSampleEntry	Section 5.6
dtse	DTS-HD low bit rate	DTSSampleEntry	Section 5.6

5.2.7 Shared elements of AudioSampleEntry

For all audio formats supported by the Common File Format, the following elements of the AudioSampleEntry box defined in [ISO] are shared:

```
class AudioSampleEntry(codingname)
   extends SampleEntry(codingname)
{
   const unsigned int(32)     reserved[2] = 0;
   template unsigned int(16)  channelcount;
   template unsigned int(16)  samplesize = 16;
   unsigned int(16)           pre_defined = 0;
   const unsigned int(16)     reserved = 0;
   template unsigned int(32)  sampleRate;
   (codingnamespecific)Box
}
```

For all audio tracks within a DCC, the value of the samplesize parameter SHALL be set to 16.

Each of the audio formats supported by the Common File Format extends the AudioSampleEntry box through the addition of a box (shown above as "(codingnamespecific)Box") containing codec-specific information that is placed within the AudioSampleEntry. This information is described in the following codec-specific sections.

5.3 MPEG-4 AAC Formats

5.3.1 General Consideration for Encoding

Since the AAC codec is based on overlap transform, and it does not establish a one-to-one relationship between input/output audio frames and audio decoding units (AUs) in bit-streams, it is necessary to be careful in handling timestamps in a track. Figure 5-1 shows an example of an AAC bit-stream in the track.

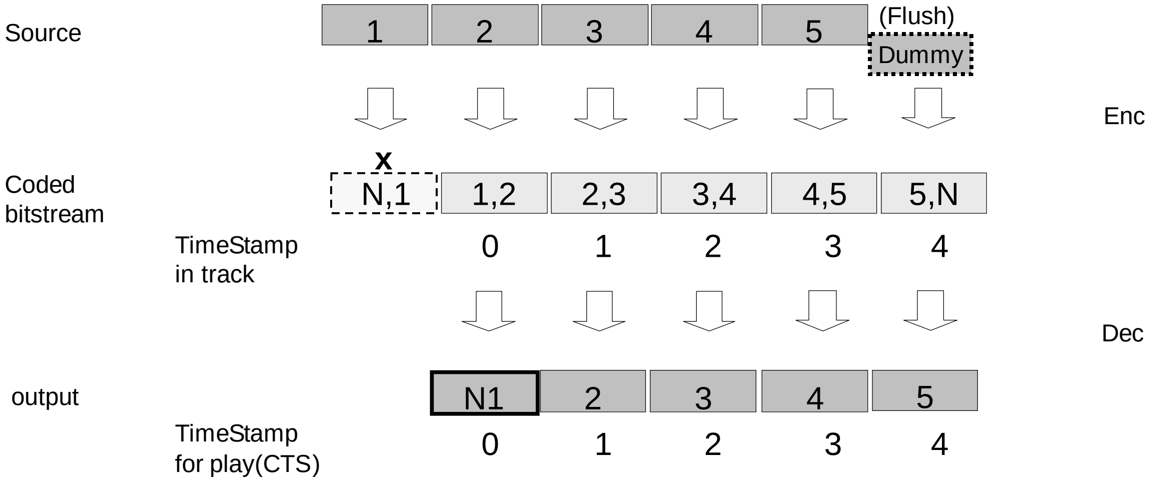


Figure 5-1 – Example of AAC bit-stream

In this figure, the first block of the bit-stream is AU [1, 2], which is created from input audio frames [1] and [2]. Depending on the encoder implementation, the first block might be AU [N, 1] (where N indicates a silent interval inserted by the encoder), but this type of AU could cause failure in synchronization and therefore SHALL NOT be included in the file.

To include the last input audio frame (i.e., [5] of source in the figure) into the bit-stream for encoding, it is necessary to terminate it with a silent interval and include AU [5, N] into the bit-stream. This produces the same number of input audio frames, AUs, and output audio frames, eliminating time difference.

When a bit-stream is created using the method described above, the decoding result of the first AU does not necessarily correspond to the first input audio frame. This is because of the lack of the first part of the bit-stream in overlap transform. Thus, the first audio frame (21 ms per frame when sampled at 48 kHz, for example) is not guaranteed to play correctly. In this case, it is up to decoder implementations to decide whether the decoded output audio frame [N1] is to be played or muted.

Taking this into consideration, the content SHOULD be created by making the first input audio frame a silent interval.

5.3.2 MPEG-4 AAC LC [2-Channel]

5.3.2.1 Storage of MPEG-4 AAC LC [2-Channel] Elementary Streams

Storage of MPEG-4 AAC LC [2-channel] elementary streams within a DCC SHALL be according to [MP4]. The following additional constraints apply when storing 2-channel MPEG-4 AAC LC elementary streams in a DCC:

- An audio sample SHALL consist of a single AAC audio access unit.
- The parameter values of `AudioSampleEntry`, `DecoderConfigDescriptor`, and `DecoderSpecificInfo` SHALL be consistent with the configuration of the AAC audio stream.

5.3.2.1.1 Audio Sample Entry Box for MPEG-4 AAC LC [2-Channel]

The syntax and values of the `AudioSampleEntry` SHALL conform to `MP4AudioSampleEntry` (`‘mp4a’`) as defined in [MP4], and the following fields SHALL be set as defined:

- `channelcount` = 1 (for mono) or 2 (for stereo)

For MPEG-4 AAC, the `(codingnamespecific)Box` that extends the `MP4AudioSampleEntry` is the `ESDBox` defined in [MP4], which contains an `ES_Descriptor`.

5.3.2.1.2 ESDBox

The syntax and values for `ES_Descriptor` SHALL conform to [MPEG4S], and the fields of the `ES_Descriptor` SHALL be set to the following specified values. Descriptors other than those specified below SHALL NOT be used.

- `ES_ID` = 0
- `streamDependenceFlag` = 0
- `URL_Flag` = 0;
- `OCRstreamFlag` = 0
- `streamPriority` = 0
- `decConfigDescr` = `DecoderConfigDescriptor` (see Section 5.3.2.1.3)
- `slConfigDescr` = `SLConfigDescriptor`, predefined type 2

5.3.2.1.3 DecoderConfigDescriptor

The syntax and values for DecoderConfigDescriptor SHALL conform to [MPEG4S], and the fields of this descriptor SHALL be set to the following specified values. In this descriptor, decoderSpecificInfo SHALL be used, and ProfileLevelIndicationIndexDescriptor SHALL NOT be used.

- objectTypeIndication = 0x40 (Audio)
- streamType = 0x05 (Audio Stream)
- upStream = 0
- decSpecificInfo = AudioSpecificConfig (see Section 5.3.2.1.4)

5.3.2.1.4 AudioSpecificConfig

The syntax and values for AudioSpecificConfig SHALL conform to [AAC], and the fields of AudioSpecificConfig SHALL be set to the following specified values:

- audioObjectType = 2 (AAC LC)
- channelConfiguration = 1 (for single mono) or 2 (for stereo)
- GASpecificConfig (see Section 5.3.2.1.5)

Channel assignment SHALL NOT be changed within the audio stream that makes up a track.

5.3.2.1.5 GASpecificConfig

The syntax and values for GASpecificConfig SHALL conform to [AAC], and the fields of GASpecificConfig SHALL be set to the following specified values:

- frameLengthFlag = 0 (1024 lines IMDCT)
- dependsOnCoreCoder = 0
- extensionFlag = 0

5.3.2.2 MPEG-4 AAC LC [2-Channel] Elementary Stream Constraints

5.3.2.2.1 General Encoding Constraints

MPEG-4 AAC [2-Channel] elementary streams SHALL conform to the requirements of the MPEG-4 AAC profile at Level 2 as specified in [AAC] with the following restrictions:

- Only the MPEG-4 AAC LC object type SHALL be used.
- The elementary stream SHALL be a Raw Data stream. ADTS and ADIF SHALL NOT be used.
- The transform length of the IMDCT for AAC SHALL be 1024 samples for long and 128 for short blocks.
- The following parameters SHALL NOT change within the elementary stream
    - ➢ Audio Object Type
    - ➢ Sampling Frequency
    - ➢ Channel Configuration
    - ➢ Bit Rate

5.3.2.2.2 Syntactic Elements

- The syntax and values for syntactic elements SHALL conform to [AAC]. The following elements SHALL NOT be present in an MPEG-4 AAC elementary stream:

- `coupling_channel_element` (CCE)

5.3.2.2.2.1 Arrangement of Syntactic Elements

- Syntactic elements SHALL be arranged in the following order for the channel configurations below.
  - <SCE><FIL><TERM>... for mono
  - <CPE><FIL><TERM>... for stereo

Note: Angled brackets (<>) are delimiters for syntactic elements.

5.3.2.2.2.2 individual_channel_stream

- The syntax and values for `individual_channel_stream` SHALL conform to [AAC]. The following fields SHALL be set as defined:
  - `gain_control_data_present` = 0

5.3.2.2.2.3 ics_info

- The syntax and values for `ics_info` SHALL conform to [AAC]. The following fields SHALL be set as defined:
  - `predictor_data_present` = 0

5.3.2.2.2.4 Maximum Bitrate

The maximum bitrate of MPEG-4 AAC LC [2-Channel] elementary streams SHALL be calculated in accordance with the AAC buffer requirements as defined in ISO/IEC 14496-3:2009, section 4.5.3. Only the raw data stream SHALL be considered in determining the maximum bitrate (system-layer descriptors are excluded).

5.3.3 MPEG-4 HE AAC V2 [5.1, 7.1-Channel]

Note that content providers encoding content according to the HE AAC V2 Profile can use any of AAC-LC, HE-AAC and HE AAC V2 profiles. Clients supporting the HE AAC V2 Profile will be able to play AAC-LC, HE-AAC and HE AAC V2 encoded content.

5.3.3.1 Storage of MPEG-4 HE AAC V2 [5.1, 7.1-Channel] Elementary Streams

Storage of MPEG-4 HE AAC V2 [5.1, 7.1-Channel] elementary streams within a DCC SHALL be according to [MP4]. The following additional constraints apply when storing MPEG-4 AAC elementary streams in a DCC.

- An audio sample SHALL consist of a single AAC audio access unit.
- The parameter values of `AudioSampleEntry`, `DecoderConfigDescriptor`, `DecoderSpecificInfo` and `program_config_element` (if present) SHALL be consistent with the configuration of the AAC audio stream.

5.3.3.1.1 Audio Sample Entry Box for MPEG-4 HE AAC V2 [5.1, 7.1-Channel]

- The syntax and values of the `AudioSampleEntry` box SHALL conform to `MP4AudioSampleEntry` (`‘mp4a’`) as defined in [MP4], and the following fields SHALL be set as defined:

- ➢ channelcount SHALL match the number of audio channels, including the LFE, in the stream

For MPEG-4 HE AAC V2 [5.1, 7.1-Channel], the (codingnamespecific)Box that extends the MP4AudioSampleEntry is the ESDBox defined in [MP4] that contains an ES_Descriptor

5.3.3.1.2 ESDBox

- The syntax and values for ES_Descriptor SHALL conform to [MPEG4S], and the fields of the ES_Descriptor SHALL be set to the following specified values. Descriptors other than those specified below SHALL NOT be used.
  - ➢ ES_ID = 0
  - ➢ streamDependenceFlag = 0
  - ➢ URL_Flag = 0
  - ➢ OCRstreamFlag = 0
  - ➢ streamPriority = 0
  - ➢ decConfigDescr = DecoderConfigDescriptor (see Section 5.3.3.1.3)
  - ➢ slConfigDescr = SLConfigDescriptor, predefined type 2

5.3.3.1.3 DecoderConfigDescriptor

- The syntax and values for DecoderConfigDescriptor SHALL conform to [MPEG4S], and the fields of this descriptor SHALL be set to the following specified values. In this descriptor, DecoderSpecificInfo SHALL always be used, and ProfileLevelIndicationIndexDescriptor SHALL NOT be used.
  - ➢ objectTypeIndication = 0x40 (Audio)
  - ➢ streamType = 0x05 (Audio Stream)
  - ➢ upStream = 0
  - ➢ decSpecificInfo = AudioSpecificConfig (see Section 5.3.3.1.4)

5.3.3.1.4 AudioSpecificConfig

- The syntax and values for AudioSpecificConfig SHALL conform to [AAC] and [AACC], and the following fields of AudioSpecificConfig SHALL be set to the specified values:
  - ➢ audioObjectType = 2 (AAC LC)
  - ➢ extensionAudioObjectType = 5 (SBR) if SBR Tool is used
  - ➢ channelConfiguration = 0 or 5 or 6 or 7 or 11 or 12 or 14
  - ➢ GASpecificConfig (see Section 5.3.3.1.5)

5.3.3.1.4.1 channelConfiguration 0

- The value of 0 for channelConfiguration is allowed for 5.1-channel streams only in this case, a program_config_element that contains program configuration data SHALL be used to specify the composition of channel elements. See Section 5.3.3.1.6 for details on the program_config_element. Channel assignment SHALL NOT be changed within the audio stream that makes up a track.

5.3.3.1.5 GASpecificConfig

- The syntax and values for `GASpecificConfig` SHALL conform to [AAC] and [AACC], and the following fields of `GASpecificConfig` SHALL be set to the specified values:
  - ➢ `frameLengthFlag` = 0 (1024 lines IMDCT)
  - ➢ `dependsOnCoreCoder` = 0
  - ➢ `extensionFlag` = 0
  - ➢ `program_config_element` (see Section 5.3.3.1.6)

5.3.3.1.6 program_config_element

- The syntax and values for `program_config_element()` (PCE) SHALL conform to [AAC], and the following fields SHALL be set as defined:
  - ➢ `element_instance_tag` = 0
  - ➢ `object_type` = 1 (AAC LC)
  - ➢ `num_front_channel_elements` = 2
  - ➢ `num_side_channel_elements` = 0
  - ➢ `num_back_channel_elements` = 1
  - ➢ `num_lfe_channel_elements` = 1
  - ➢ `num_assoc_data_elements` = 0 or 1
  - ➢ `num_valid_cc_elements` = 0
  - ➢ `mono_mixdown_present` = 0
  - ➢ `stereo_mixdown_present` = 0
  - ➢ `matrix_mixdown_idx_present` = 0 or 1
  - ➢ `if (matrix_mixdown_idx_present = = 1) {`

    `matrix_mixdown_idx` = 0 to 3

    `pseudo_surround_enable` = 0 or 1

    `}`
  - ➢ `front_element_is_cpe[0]` = 0
  - ➢ `front_element_is_cpe[1]` = 1
  - ➢ `back_element_is_cpe[0]` = 1

5.3.3.2 MPEG-4 HE AAC V2[5.1, 7.1-channel] Elementary Stream Constraints

5.3.3.2.1 General Encoding Constraints

MPEG-4 HE AAC V2 [5.1, 7.1-channel] elementary streams SHALL conform to the requirements of the MPEG-4 AAC profile at Level 6 as specified in [AAC] with the following restrictions:

- The elementary stream SHALL be a Raw Data stream. ADTS and ADIF SHALL NOT be used.
- The transform frame length of an AAC frame (access unit) SHALL be 1024 samples (1024 IMDCT lines for long and 8 times128 for short blocks.
- The following parameters SHALL NOT change within the elementary stream:
  - ➢ Audio Object Type
  - ➢ Sampling Frequency
  - ➢ Channel Configuration

5.3.3.2.2 Syntactic Elements

- The syntax and values for syntactic elements SHALL conform to [AAC] and [AACC]. The following elements SHALL NOT be present in an MPEG-4 AAC elementary stream:
  - `coupling_channel_element` (CCE)
- The syntax and values for syntactic elements SHALL conform to [AAC] and [AACC]. The following elements SHALL be present in an MPEG-4 AAC elementary stream:
  - `dynamic_range_info()`

5.3.3.2.2.1 6.1 and 7.1 Channel Configurations

In addition to the above, for 6.1 and 7.1 Channel Configuration following elements SHALL be present in an MPEG-4 AAC elementary stream:

  - `MPEG4_ancillary_data()`

5.3.3.2.2.2 Arrangement of Syntactic Elements

- Syntactic elements SHALL be arranged in the following [AAC] i.e. ordering of channel elements SHALL follow Table 1.19 – Channel Configuration of [AAC], for example:
  - <SCE><FIL-<EXT-SBR>><CPE><FIL-<EXT-SBR>><CPE><FIL-<EXT-SBR>><LFE>>…<FIL-<EXT-DRC>><FIL><TERM>… for channelConfiguration 6 (5.1-channels)

Note: Angled brackets (<>) are delimiters for syntactic elements.

5.3.3.2.2.3 individual_channel_stream

- The syntax and values for `individual_channel_stream` SHALL conform to [AAC] and [AACC]. The following fields SHALL be set as defined:
  - `gain_control_data_present` = 0;

5.3.3.2.2.4 ics_info

- The syntax and values for `ics_info` SHALL conform to [AAC] and [AACC]. The following fields SHALL be set as defined:
  - `predictor_data_present` = 0;

5.3.3.2.2.5 Maximum Bitrate

The maximum bitrate of MPEG-4 HE AAC V2 [5.1, 7.1-Channel] elementary streams SHALL be calculated in accordance with the AAC buffer requirements as defined in ISO/IEC 14496-3:2009, section 4.5.3 (288 kbps per full audio channel at 48 kHz AAC core sampling rate). Only the raw data stream SHALL be considered in determining the maximum bitrate (system-layer descriptors are excluded).

5.3.4 MPEG-4 HE AAC v2

5.3.4.1 Storage of MPEG-4 HE AAC v2 Elementary Streams

Storage of MPEG-4 HE AAC v2 elementary streams within a DCC SHALL be according to [MP4]. The following requirements SHALL be met when storing MPEG-4 HE AAC v2 elementary streams in a DCC.

- An audio sample SHALL consist of a single HE AAC v2 audio access unit.

- The parameter values of AudioSampleEntry, DecoderConfigDescriptor, and DecoderSpecificInfo SHALL be consistent with the configuration of the MPEG-4 HE AAC v2 audio stream.

5.3.4.1.1 Audio Sample Entry Box for MPEG-4 HE AAC v2

- The syntax and values of the AudioSampleEntry box SHALL conform to MP4AudioSampleEntry (‘mp4a’) defined in [MP4], and the following fields SHALL be set as defined:
  - channelcount = 1 (for mono or parametric stereo) or 2 (for stereo)

For MPEG-4 AAC, the (codingnamespecific)Box that extends the MP4AudioSampleEntry is the ESDBox defined in ISO 14496-14 [14], which contains an ES_Descriptor.

5.3.4.1.2 ESDBox

- The ESDBox contains an ES_Descriptor. The syntax and values for ES_Descriptor SHALL conform to [MPEG4S], and the fields of the ES_Descriptor SHALL be set to the following specified values. Descriptors other than those specified below SHALL NOT be used.
  - ES_ID = 0
  - streamDependenceFlag = 0
  - URL_Flag = 0
  - OCRstreamFlag = 0 (false)
  - streamPriority = 0
  - decConfigDescr = DecoderConfigDescriptor (see Section 5.3.4.1.3)
  - slConfigDescr = SLConfigDescriptor, predefined type 2

5.3.4.1.3 DecoderConfigDescriptor

- The syntax and values for DecoderConfigDescriptor SHALL conform to [MPEG4S], and the fields of this descriptor SHALL be set to the following specified values. In this descriptor, DecoderSpecificInfo SHALL be used, and ProfileLevelIndicationIndexDescriptor SHALL NOT be used.
  - objectTypeIndication = 0x40 (Audio)
  - streamType = 0x05 (Audio Stream)
  - upStream = 0
  - decSpecificInfo = AudioSpecificConfig (see Section 5.3.4.1.4)

5.3.4.1.4 AudioSpecificConfig

- The syntax and values for AudioSpecificConfig SHALL conform to [AAC] and the fields of AudioSpecificConfig SHALL be set to the following specified values:
  - audioObjectType = 5 (SBR)
  - channelConfiguration = 1 (for mono or parametric stereo) or 2 (for stereo)
  - underlying audio object type = 2 (AAC LC)
  - GASpecificConfig (see Section 5.3.4.1.5)

This configuration uses explicit hierarchical signaling to indicate the use of the SBR coding tool, and implicit signaling to indicate the use of the PS coding tool.

5.3.4.1.5 GASpecificConfig

- The syntax and values for `GASpecificConfig` SHALL conform to [AAC], and the fields of `GASpecificConfig` SHALL be set to the following specified values.
  - `frameLengthFlag` = 0 (1024 lines IMDCT)
  - `dependsOnCoreCoder` = 0
  - `extensionFlag` = 0

5.3.4.2 MPEG-4 HE AAC v2 Elementary Stream Constraints

Note: MPEG-4 HE AAC v2 is the superset of MPEG-4 AAC, MPEG-4 HE AAC and MPEG-4 HE AAC v2.

5.3.4.2.1 General Encoding Constraints

The MPEG-4 HE AAC v2 elementary stream as defined in [AAC] SHALL conform to the requirements of the MPEG-4 HE AAC v2 Profile at Level 2, except as follows:

- The elementary stream MAY be encoded according to the MPEG-4 AAC, HE AAC or HE AAC v2 Profile. Use of the MPEG-4 HE AAC v2 profile is recommended.
- The audio SHALL be encoded in mono, parametric stereo or 2-channel stereo.
- The transform length of the IMDCT for AAC SHALL be 1024 samples for long and 128 for short blocks.
- The elementary stream SHALL be a Raw Data stream. ADTS and ADIF SHALL NOT be used.
- The following parameters SHALL NOT change within the elementary stream:
  - Audio Object Type
  - Sampling Frequency
  - Channel Configuration
  - Bit Rate

5.3.4.2.2 Syntactic Elements

- The syntax and values for syntactic elements SHALL conform to [AAC]. The following elements SHALL NOT be present in an MPEG-4 HE AAC v2 elementary stream:
  - `coupling_channel_element` (CCE)
  - `program_config_element` (PCE).

5.3.4.2.2.1 Arrangement of Syntactic Elements

- Syntactic elements SHALL be arranged in the following order for the channel configurations below.
  - <SCE><FIL><TERM>… for mono and parametric stereo
  - <CPE><FIL><TERM>... for stereo

5.3.4.2.2.2 ics_info

- The syntax and values for `ics_info` SHALL conform to [AAC]. The following fields SHALL be set as defined:
  - `predictor_data_present` = 0

5.3.4.2.2.3 Maximum Bitrate

The maximum bitrate of MPEG-4 HE AAC v2 elementary streams in a DCC SHALL be calculated in accordance with the AAC buffer requirements as defined in ISO/IEC 14496-3:2009, section 4.5.3. Only the raw data stream SHALL be considered in determining the maximum bitrate (system-layer descriptors are excluded).

5.3.5 MPEG-4 HE AAC v2 with MPEG Surround

Note: MPEG-4 HE AAC v2 is the superset of MPEG-4 AAC, MPEG-4 HE AAC and MPEG-4 HE AAC v2.

5.3.5.1 Storage of MPEG-4 HE AAC v2 Elementary Streams with MPEG Surround

Storage of MPEG-4 HE AAC v2 elementary streams that contain MPEG Surround spatial audio data within a DCC SHALL be according to [MP4] and [AAC]. The requirements defined in Section 5.3.4.1 SHALL be met when storing MPEG-4 HE AAC v2 elementary streams containing MPEG Surround spatial audio data in a DCC. Additionally:

- The presence of MPEG Surround spatial audio data within an MPEG-4 AAC, HE AAC or HE AAC v2 elementary stream SHALL be indicated using explicit backward compatible signaling as specified in [AAC].
    - The `mpsPresentFlag` within the `AudioSpecificConfig` SHALL be set to 1.

5.3.5.2 MPEG-4 HE AAC v2 with MPEG Surround Elementary Stream Constraints

5.3.5.2.1 General Encoding Constraints

The elementary stream as defined in [AAC] and [MPS] SHALL be encoded according to the functionality defined in the MPEG-4 AAC, HE AAC or HE AAC v2 Profile at Level 2, in combination with the functionality defined in MPEG Surround Baseline Profile Level 4, with the following additional constraints:

- The audio SHALL be encoded in mono, parametric stereo or 2-channel stereo.
- The transform length of the IMDCT for AAC SHALL be 1024 samples for long and 128 for short blocks.
- The elementary stream SHALL be a Raw Data stream. ADTS and ADIF SHALL NOT be used.
- The following parameters SHALL NOT change within the elementary stream:
    - Audio Object Type
    - Sampling Frequency
    - Channel Configuration
    - Bit Rate
- The MPEG Surround payload data SHALL be embedded within the core elementary stream, as specified in [AAC] and SHALL NOT be carried in a separate audio track.
- The sampling frequency of the MPEG Surround payload data SHALL be equal to the sampling frequency of the core elementary stream.
- Separate fill elements SHALL be employed to embed the SBR/PS extension data elements `sbr_extension_data()` and the MPEG Surround spatial audio data `SpatialFrame()`.

- The value of `bsFrameLength` SHALL be set to 15, 31 or 63, resulting in effective MPEG Surround frame lengths of 1024, 2048 or 4096 time domain samples respectively.
- All audio access units SHALL contain an extension payload of type `EXT_SAC_DATA`.
- The interval between occurrences of `SpatialSpecificConfig` in the bit-stream SHALL NOT exceed 500 ms. Within the corresponding `SpatialFrame()` the value of `bsIndependencyFlag` SHALL be set to one.
- To ensure consistent decoder behavior during trick play operations, the first `AudioSample` of each fragment SHALL contain the `SpatialSpecificConfig` structure. Within the corresponding `SpatialFrame()` the value of `bsIndependencyFlag` SHALL be set to one.

5.3.5.2.2 Syntactic Elements

- The syntax and values for syntactic elements SHALL conform to [AAC] and [MPS]. The following elements SHALL NOT be present in an MPEG-4 HE AAC v2 elementary stream that contains MPEG Surround data:
  - ➢ `coupling_channel_element` (CCE)
  - ➢ `program_config_element` (PCE).

5.3.5.2.2.1 Arrangement of Syntactic Elements

- Syntactic elements SHALL be arranged in the following order for the channel configurations below:
  - ➢ <SCE><FIL><FIL><TERM>… for mono and parametric stereo core audio streams
  - ➢ <CPE><FIL><FIL><TERM>... for stereo core audio streams

5.3.5.2.2.2 ics_info

- The syntax and values for `ics_info` SHALL conform to [AAC]. The following fields SHALL be set as defined:
  - ➢ `predictor_data_present` = 0

5.3.5.2.2.3 Maximum Bitrate

The maximum bitrate of MPEG-4 HE AAC v2 elementary streams that contain MPEG Surround spatial audio data SHALL be calculated in accordance with the AAC buffer requirements as defined in ISO/IEC 14496-3:2009, section 4.5.3. Only the raw data stream SHALL be considered in determining the maximum bitrate (system-layer descriptors are excluded).

5.4 AC-3, Enhanced AC-3, MLP and DTS Format Timing Structure

Unlike the MPEG-4 audio formats, the DTS and Dolby formats do not overlap between frames. Synchronized frames represent a contiguous audio stream where each audio frame represents an equal size block of samples at a given sampling frequency. See Figure 5-2 for illustration.

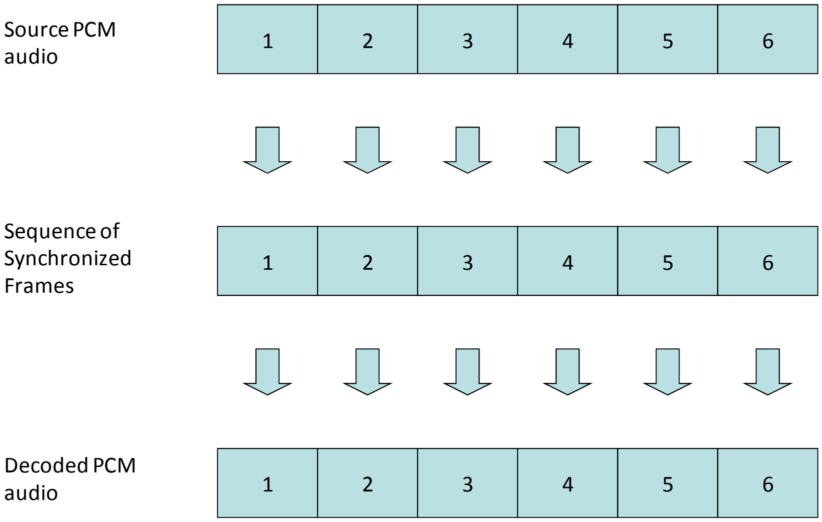


Figure 5-2 – Non-AAC bit-stream example

Additionally, unlike AAC audio formats, the DTS and Dolby formats do not require external metadata to set up the decoder, as they are fully contained in that regard. Descriptor data is provided, however, to provide information to the system without requiring access to the elementary stream, as the ES is typically encrypted in the DCC.

5.5 Dolby Formats

5.5.1 AC-3 (Dolby Digital)

5.5.1.1 Storage of AC-3 Elementary Streams

Storage of AC-3 elementary streams within a DCC SHALL be according to Annex F of [EAC3].

- An audio sample SHALL consist of a single AC-3 frame.
- Note that per Annex F of [EAC3] the audio stream can be encoded either "big endian" or "little endian" byte order. Big endian SHOULD be used.

5.5.1.1.1 Audio Sample Entry Box for AC-3

The syntax and values of the `AudioSampleEntry` box SHALL conform to `AC3SampleEntry` (`‘ac-3’`) as defined in Annex F of [EAC3]. The configuration of the AC-3 elementary stream is described in the `AC3SpecificBox` (`‘dac3’`) within `AC3SampleEntry`, as defined in Annex F of [EAC3]. For convenience the syntax and semantics of the `AC3SpecificBox` are replicated in Section 5.5.1.1.2.

5.5.1.1.2 AC3Specific Box

The syntax of the `AC3SpecificBox` is shown below:

```
Class AC3SpecificBox
{
   unsigned int(2)  fscod;
   unsigned int(5)  bsid;
   unsigned int(3)  bsmod;
   unsigned int(3)  acmod;
   unsigned int(1)  lfeon;
   unsigned int(5)  bit_rate_code;
   unsigned int(5)  reserved = 0;
}
```

5.5.1.1.2.1 Semantics

The `fscod`, `bsid`, `bsmod`, `acmod` and `lfeon` fields have the same meaning and are set to the same value as the equivalent parameters in the AC-3 elementary stream. The `bit_rate_code` field is derived from the value of `frmsizcod` in the AC-3 bit-stream according to Table 5-2.

Table 5-2 – bit_rate_code

bit_rate_code	Nominal bit rate (kbit/s)
00000	32
00001	40
00010	48
00011	56
00100	64
00101	80
00110	96
00111	112
01000	128
01001	160
01010	192
01011	224
01100	256
01101	320
01110	384
01111	448
10000	512
10001	576
10010	640

The contents of the `AC3SpecificBox` SHALL NOT be used to configure or control the operation of an AC-3 audio decoder.

5.5.1.2 AC-3 Elementary Stream Constraints

AC-3 elementary streams SHALL comply with the syntax and semantics as specified in [EAC3], not including Annex E. Additional constraints on AC-3 audio streams are specified in this section.

5.5.1.2.1 General Encoding Constraints

AC-3 elementary streams SHALL be constrained as follows:

- The minimum bit rate of an AC-3 elementary stream SHALL be 64x10^3 bits/second.

- The following bit-stream parameters SHALL remain constant within an AC-3 elementary stream for the duration of an AC-3 audio track:
    - ➢ `bsid`
    - ➢ `bsmod`
    - ➢ `acmod`
    - ➢ `lfeon`
    - ➢ `fscod`
    - ➢ `frmsizcod`

5.5.1.2.2 AC-3 synchronization frame constraints

- AC-3 synchronization frames SHALL comply with the following constraints:
    - ➢ `bsid` – bit-stream identification: This field SHALL be set to 1000b (8), or 110b (6) when the alternate bit-stream syntax described in Annex D of [EAC3] is used.
    - ➢ `frmsizecod` – frame size code: This field SHALL be set to a value between 001000b to 100101b (64Kbps to 640Kbps).
    - ➢ `acmod` – audio coding mode: All audio coding modes except dual mono (`acmod` = 000b) defined in Table 4-3 of [EAC3] are permitted.

5.5.1.2.3 Maximum Bitrate

The maximum bitrate of AC-3 elementary streams SHALL be calculated as the sample size divided by the duration.

Note: The minimum sample size for AC-3 is 256 bytes (64 Kbps). There will only be one size value for the whole track as the stream is CBR. The duration of the sample is 0.032 seconds.

5.5.2 Enhanced AC-3 (Dolby Digital Plus)

5.5.2.1 Storage of Enhanced AC-3 Elementary Streams

Storage of Enhanced AC-3 elementary streams within a DCC SHALL be according to Annex F of [EAC3].

- An audio sample SHALL consist of the number of syncframes required to deliver six blocks of audio data from each substream in the Enhanced AC-3 elementary stream (defined as an Enhanced AC-3 Access Unit).
- The first syncframe of an audio sample SHALL be the syncframe that has a stream type value of 0 (independent) and a substream ID value of 0.
- For Enhanced AC-3 elementary streams that consist of syncframes containing fewer than 6 blocks of audio, the first syncframe of an audio sample SHALL be the syncframe that has a stream type value of 0 (independent), a substream ID value of 0, and has the “convsync” flag set to “1”.
- Note that per Annex F of [EAC3] the audio stream can be encoded either "big endian" or "little endian" byte order. Big endian SHOULD be used.

5.5.2.1.1 Audio Sample Entry Box for Enhanced AC-3

The syntax and values of the `AudioSampleEntry` box SHALL conform to EC3SampleEntry (`‘ec-3’`) defined in Annex F of [EAC3]. The configuration of the Enhanced AC-3 elementary stream is described in the EC3SpecificBox ('dec3'), within EC3SampleEntry, as defined in Annex F of [EAC3]. For convenience the syntax and semantics of the EC3SpecificBox are replicated in Section 5.5.2.1.2.

5.5.2.1.2 EC3SpecificBox

The syntax and semantics of the EC3SpecificBox are shown below. The syntax shown is a simplified version of the full syntax defined in Annex F of [EAC3], as the Enhanced AC-3 encoding constraints specified in Section 5.5.2.2 restrict the number of independent substreams to 1, so only a single set of independent substream parameters is included in the EC3SpecificBox.

```
class EC3SpecificBox
{
   unsigned int(13)  data_rate;
   unsigned int(3)   num_ind_sub;
   unsigned int(2)   fscod;
   unsigned int(5)   bsid;
   unsigned int(5)   bsmod;
   unsigned int(3)   acmod;
   unsigned int(1)   lfeon;
   unsigned int(3)   reserved = 0;
   unsigned int(4)   num_dep_sub;
   if (num_dep_sub > 0)
   {
      unsigned int(9)  chan_loc;
   }
   else
   {
      unsigned int(1)  reserved = 0;
   }
}
```

5.5.2.1.2.1 Semantics

- data_rate – this field indicates the bit rate of the Enhanced AC-3 elementary stream in kbit/s. For Enhanced AC-3 elementary streams within a DCC, the minimum value of this field is 32.
- num_ind_sub – This field indicates the number of independent substreams that are present in the Enhanced AC-3 bit-stream. The value of this field is one less than the number of independent substreams present. For Enhanced AC-3 elementary streams within a DCC, this field is always set to 0 (indicating that the Enhanced AC-3 elementary stream contains a single independent substream).
- fscod – This field has the same meaning and is set to the same value as the fscod field in independent substream 0.
- bsid – This field has the same meaning and is set to the same value as the bsid field in independent substream 0.
- bsmod – This field has the same meaning and is set to the same value as the bsmod field in independent substream 0. If the bsmod field is not present in independent substream 0, this field SHALL be set to 0.
- acmod – This field has the same meaning and is set to the same value as the acmod field in independent substream 0.

- lfeon – This field has the same meaning and is set to the same value as the lfeon field in independent substream 0.
- num_dep_sub – This field indicates the number of dependent substreams that are associated with independent substream 0. For Enhanced AC-3 elementary streams within a DCC, this field MAY be set to 0 or 1.
- chan_loc – If there is a dependent substream associated with independent substream, this bit field is used to identify channel locations beyond those identified using the acmod field that are present in the bit-stream. For each channel location or pair of channel locations present, the corresponding bit in the chan_loc bit field is set to "1", according to Table 5-3. This information is extracted from the chanmap field of the dependent substream.

Table 5-3 – chan_loc field bit assignments

Bit	Location
0	Lc/Rc pair
1	Lrs/Rrs pair
2	Cs
3	Ts
4	Lsd/Rsd pair
5	Lw/Rw pair
6	Lvh/Rvh pair
7	Cvh
8	LFE2

The contents of the EC3SpecificBox SHALL NOT be used to control the configuration or operation of an Enhanced AC-3 audio decoder.

5.5.2.2 Enhanced AC-3 Elementary Stream Constraints

Enhanced AC-3 elementary streams SHALL comply with the syntax and semantics as specified in [EAC3], including Annex E. Additional constraints on Enhanced AC-3 audio streams are specified in this section.

5.5.2.2.1 General Encoding Constraints

Enhanced AC-3 elementary streams SHALL be constrained as follows:

- The minimum bit rate of an Enhanced AC-3 elementary stream SHALL be 32x10^3 bits/second.
- An Enhanced AC-3 elementary stream SHALL always contain at least one independent substream (stream type 0) with a substream ID of 0. An Enhanced AC-3 elementary stream MAY also additionally contain one dependent substream (stream type 1).
- The following bit-stream parameters SHALL remain constant within an Enhanced AC-3 elementary stream for the duration of an Enhanced AC-3 track:
    - ➢ Number of independent substreams
    - ➢ Number of dependent substreams
    - ➢ Within independent substream 0:
        - bsid
        - bsmod
        - acmod
        - lfeon
        - fscod

- Within dependent substream 0:
  - `bsid`
  - `acmod`
  - `lfeon`
  - `fscod`
  - `chanmap`

5.5.2.2.2 Independent substream 0 constraints

Independent substream 0 consists of a sequence of Enhanced AC-3 synchronization frames. These synchronization frames SHALL comply with the following constraints:

- `bsid` – bit-stream identification: If independent substream 0 is the only substream in the Enhanced AC-3 elementary stream, this field SHALL be set to 10000b (16). If the Enhanced AC-3 elementary stream contains both independent substream 0 and dependent substream 0, this field SHALL be set to 00110 (6), 01000 (8) or 10000 (16).
- When `bsid=10000b` (16), then:
  - `strmtyp` – stream type: This field SHALL be set to 00b (Stream Type 0 – independent substream); and
  - `substreamid` – substream identification: This field SHALL be set to 000b (substream ID = 0).
- `acmod` – audio coding mode: All audio coding modes except dual mono (acmod=000b) defined in Table 4-3 of [EAC3] are permitted. Audio coding mode dual mono (acmod=000b) SHALL NOT be used.

5.5.2.2.3 Dependent substream constraints

Dependent substream 0 consists of a sequence of Enhanced AC-3 synchronization frames. These synchronization frames SHALL comply with the following constraints:

- `bsid` – bit-stream identification: This field SHALL be set to 10000b (16).
- `strmtyp` – stream type: This field SHALL be set to 01b (Stream Type 1 – dependent substream).
- `substreamid` – substream identification: This field SHALL be set to 000b (substream ID = 0).
- `acmod` – audio coding mode: All audio coding modes except dual mono (acmod=000b) defined in Table 4-3 of [EAC3] are permitted. Audio coding mode dual mono (acmod=000b) SHALL NOT be used.

5.5.2.2.4 Substream configuration for delivery of more than 5.1 channels of audio

To deliver more than 5.1 channels of audio, both independent (Stream Type 0) and dependent (Stream Type 1) substreams are included in the Enhanced AC-3 elementary stream. The channel configuration of the complete elementary stream is defined by the `acmod` parameter carried in the independent substream, and the `acmod` and `chanmap` parameters carried in the dependent substream. The loudspeaker locations supported by Enhanced AC-3 are defined in [SMPTE428].

The following rules apply to channel numbers and substream use:

- When more than 5.1 channels of audio are to be delivered, independent substream 0 of an Enhanced AC-3 elementary stream SHALL be configured as a downmix of the complete program.

- Additional channels necessary to deliver up to 7.1 channels of audio SHALL be carried in dependent substream 0.

5.5.2.2.5 Maximum Bitrate

The maximum bitrate of Enhanced AC-3 elementary streams SHALL be calculated as the sample size divided by the duration.
Note: The minimum sample size of Enhanced AC-3 is 128 bytes (32 Kbps). As there are always six blocks of audio data from every substream present in the sample, the duration of each sample is the same as AC-3 – 0.032 seconds.

5.5.3 MLP (Dolby TrueHD)

5.5.3.1 Storage of MLP elementary streams

Storage of MLP elementary streams within a DCC SHALL be according to [MLPISO].

- An audio sample SHALL consist of a single MLP access unit as defined in [MLP].

5.5.3.1.1 Audio Sample Entry Box for MLP

The syntax and values of the `AudioSampleEntry` box SHALL conform to `MLPSampleEntry` (`‘mlpa’`) defined in [MLPISO].
Within `MLPSampleEntry`, the `sampleRate` field has been redefined as a single 32-bit integer value, rather than the 16.16 fixed-point field defined in the ISO base media file format. This enables explicit support for sampling frequencies greater than 48 kHz.
The configuration of the MLP elementary stream is described in the `MLPSpecificBox` (`‘dmlp’`), within `MLPSampleEntry`, as described in [MLPISO]. For convenience the syntax and semantics of the `MLPSpecificBox` are replicated in Section 5.5.3.1.2.

5.5.3.1.2 MLPSpecificBox

The syntax and semantics of the `MLPSpecificBox` are shown below:

```
Class MLPSpecificBox
{
   unsigned int(32)  format_info;
   unsigned int(15)  peak_data_rate;
   unsigned int(1)   reserved = 0;
   unsigned int(32)  reserved = 0;
}
```

5.5.3.1.2.1 Semantics

- `format_info` – This field has the same meaning and is set to the same value as the `format_info` field in the MLP bit-stream.
- `peak_data_rate` – This field has the same meaning and is set to the same value as the `peak_data_rate` field in the MLP bit-stream.

The contents of the `MLPSpecificBox` SHALL NOT be used to control the configuration or operation of an MLP audio decoder.

5.5.3.2 MLP Elementary Stream Constraints

MLP elementary streams SHALL comply with the syntax and semantics as specified in [MLP]. Additional constraints on MLP audio streams are specified in this section.

5.5.3.2.1 General Encoding Constraints

MLP elementary streams SHALL be constrained as follows:

- All MLP elementary streams SHALL comply with MLP Form B syntax, and the stream type SHALL be FBA streams.
- The sample rate of all substreams within the MLP bit-stream SHALL be identical.
- The following parameters SHALL remain constant within an MLP elementary stream for the duration of an MLP audio track.
  - `audio_sampling_frequency` – sampling frequency
  - `substreams` – number of MLP substreams
  - `min_chan` and `max_chan` in each substream – number of channels
  - `6ch_source_format` and `8ch_source_format` – audio channel assignment
  - `substream_info` – substream configuration

5.5.3.2.2 MLP access unit constraints

- Sample rate – The sample rate SHALL be identical on all channels.
- Sampling phase – The sampling phase SHALL be simultaneous for all channels.
- Wordsize – The quantization of source data and of coded data MAY be different. The quantization of coded data is always 24 bits. When the quantization of source data is fewer than 24 bits, the source data is padded to 24 bits by adding bits of zero ('0') as the least significant bit(s).
- 2-ch decoder support – The stream SHALL include support for a 2-ch decoder.
- 6-ch decoder support – The stream SHALL include support for a 6-ch decoder when the total stream contains more than 6 channels.
- 8-ch decoder support – The stream SHALL include support for an 8-ch decoder.

5.5.3.2.3 Loudspeaker Assignments

The MLP elementary stream supports 2-channel, 6-channel and 8-channel presentations. Loudspeaker layout options are described for each presentation in the stream. Please refer to Appendix E of "Meridian Lossless Packing - Technical Reference for FBA and FBB streams" Version 1.0. The loudspeaker locations supported by MLP are defined in [SMPTE428].

5.5.3.2.4 Maximum Bitrate

The maximum bitrate of MLP elementary streams SHALL be calculated according to MLP Tech Ref [MLP] Section 8.8.1.

5.6 DTS Formats

5.6.1 Storage of DTS elementary streams

Storage of DTS formats within a DCC SHALL be according to this specification.

- An audio sample SHALL consist of a single DTS audio frame, as defined in [DTS].

5.6.1.1 Audio Sample Entry Box for DTS Formats

The syntax and values of the AudioSampleEntry Box SHALL conform to DTSSampleEntry.
The parameter sampleRate SHALL be set to either the sampling frequency indicated by SFREQ in the core substream or to the frequency represented by the parameter nuRefClockCode in the extension substream.
The configuration of the DTS elementary stream is described in the DTSSpecificBox (‘ddts’), within DTSSampleEntry. The syntax and semantics of the DTSSpecificBox are defined in the following section.
The parameter channelcount SHALL be set to the number of decodable output channels in basic playback, as described in the (‘ddts’) configuration box.

5.6.1.2 DTSSpecificBox

The syntax and semantics of the DTSSpecificBox are shown below.

```
class DTSSpecificBox
{
   unsigned int(32)  size;          //Box.size
   unsigned char[4]  type=‘ddts’;   //Box.type
   unsigned int(32)  DTSSamplingFrequency;
   unsigned int(32)  maxBitrate;
   unsigned int(32)  avgBitrate;
   unsigned char     reserved = 0;
   bit(2)   FrameDuration;          // 0=512, 1=1024, 2=2048, 3=4096
   bit(5)   StreamConstruction;     // Table 5-4
   bit(1)   CoreLFEPresent;         // 0=none; 1=LFE exists
   bit(6)   CoreLayout;             // Table 5-5
   bit(14)  CoreSize;               // FSIZE, Not to exceed 4064 bytes
   bit(1)   StereoDownmix           // 0=none; 1=emb. downmix present
   bit(3)   RepresentationType;     // Table 5-6
   bit(16)  ChannelLayout;          // Table 5-7
   bit(8)  reserved = 0;
}
```

5.6.1.2.1 Semantics

- DTSSamplingFrequency – The maximum sampling frequency stored in the compressed audio stream.
- maxBitrate – The peak bit rate, in bits per second, of the audio elementary stream for the duration of the track. The calculated value will be rounded up to the nearest integer.
- avgBitrate – The average bit rate, in bits per second, of the audio elementary stream for the duration of the track. The calculated value will be rounded up to the nearest integer.

- `FrameDuration` – This code represents the number of audio samples decoded in a complete audio access unit at `DTSSamplingFrequency`.
- `CoreLayout` – This parameter is identical to the DTS Core substream header parameter `AMODE` [DTS] and represents the channel layout of the core substream prior to applying any information stored in any extension substream. See Table 5-5. If no core substream exists, this parameter SHALL be ignored.
- `CoreLFEPresent` – Indicates the presence of an LFE channel in the core. If no core exists, this value SHALL be ignored.
- `StreamConstructon` – Provides complete information on the existence and of location of extensions in any synchronized frame. See Table 5-4.
- `ChannelLayout` – This parameter is identical to `nuSpkrActivitymask` defined in the extension substream header [DTS]. This 16-bit parameter that provides complete information on channels coded in the audio stream including core and extensions. See Table 5-7. The binary masks of the channels present in the stream are added together to create `ChannelLayout`.
- `StereoDownmix` – Indicates the presence of an embedded stereo downmix in the stream. This parameter is not valid for stereo or mono streams.
- `CoreSize` – This parameter is derived from FSIZE in the core substream header [DTS] and it represents a core frame payload in bytes. In the case where an extension substream exists in an access unit, this represents the size of the core frame payload only. This simplifies extraction of just the core substream for decoding or exporting on interfaces such as S/PDIF. The value of `CoreSize` will always be less than or equal to 4064 bytes.

  In the case when `CoreSize`=0, `CoreLayout` and `CoreLFEPresent` SHALL be ignored. `ChannelLayout` will be used to determine channel configuration.
- `RepresentationType` – This parameter is derived from the value for `nuRepresentationtype` in the substream header [DTS]. This indicates special properties of the audio presentation. See Table 5-6. This parameter is only valid when all flags in `ChannelLayout` are set to 0. If `ChannelLayout` ≠ 0, this value SHALL be ignored.

Table 5-4 – StreamConstruction

StreamConstruction	Core substream				Extension substream				
	Core	XCH	X96	XXCH	XXCH	X96	XBR	XLL	LBR

1	✓								
2	✓	✓							
3	✓			✓					
4	✓		✓						
5	✓				✓				
6	✓						✓		
7	✓	✓					✓		
8	✓			✓			✓		
9	✓				✓		✓		
10	✓					✓			
11	✓	✓				✓			
12	✓			✓		✓			
13	✓				✓	✓			
14	✓							✓	
15	✓	✓						✓	
16	✓		✓					✓	
17								✓	
18									✓

Table 5-5 – CoreLayout

CoreLayout	Description
0	Mono (1/0)
2	Stereo (2/0)
4	LT, RT (2/0)
5	L, C, R (3/0)
7	L, C, R, S (3/1)
6	L, R, S (2/1)
8	L, R. LS, RS (2/2)
9	L, C, R, LS, RS (3/2)

Table 5-6 – RepresentationType

RepresentationType	Description
000b	Audio asset designated for mixing with another audio asset
001b	Reserved
010b	Lt/Rt Encoded for matrix surround decoding; it implies that total number of encoded channels is 2
011b	Audio processed for headphone playback; it implies that total number of encoded channels is 2
100b	Not Applicable
101b– 111b	Reserved

Table 5-7 – ChannelLayout

Notation	*Loudspeaker Location Description*	*Bit Masks*	*Number of Channels*
C	Center in front of listener	0x0001	1
LR	Left/Right in front	0x0002	2
LsRs	Left/Right surround on side in rear	0x0004	2
LFE1	Low frequency effects subwoofer	0x0008	1
Cs	Center surround in rear	0x0010	1
LhRh	Left/Right height in front	0x0020	2
LsrRsr	Left/Right surround in rear	0x0040	2
Ch	Center Height in front	0x0080	1
Oh	Over the listener’s head	0x0100	1
LcRc	Between left/right and center in front	0x0200	2
LwRw	Left/Right on side in front	0x0400	2
LssRss	Left/Right surround on side	0x0800	2
LFE2	Second low frequency effects subwoofer	0x1000	1
LhsRhs	Left/Right height on side	0x2000	2
Chr	Center height in rear	0x4000	1
LhrRhr	Left/Right height in rear	0x8000	2

5.6.2 Restrictions on DTS Formats

This section describes the restrictions that SHALL be applied to the DTS formats encapsulated in a DCC.

5.6.2.1 General constraints

The following conditions SHALL NOT change in a DTS audio stream or a Core substream:

- Duration of Synchronized Frame
- Bit Rate
- Sampling Frequency
- Audio Channel Arrangement
- Low Frequency Effects flag
- Extension assignment

The following conditions SHALL NOT change in an Extension substream:

- Duration of Synchronized Frame
- Sampling Frequency
- Audio Channel Arrangement
- Low Frequency Effects flag
- Embedded stereo flag
- Extensions assignment defined in `StreamConstruction`

5.6.2.2 Maximum Bitrate

The maximum bitrate of DTS elementary streams SHALL be calculated from a single audio frame (one sample), by dividing the size in bits of the largest sample by the time duration of that sample.

Note: maximum bitrate is represented in the `DTSSampleEntry` as `maxBitrate`. This is a 32-bit integer value represented in bits/second and is calculated only from the audio elementary stream, excluding any and all other ISOBMFF constructions. The value is calculated using floating point arithmetic and any fractional remainder in the calculation is rounded up to the integer portion of the result and that integer is used to represent the value.

6 Subtitle Elementary Streams

6.1 Overview

This chapter defines the CFF subtitle elementary stream format, how it is stored in a DCC as a track, and how it is synchronized and presented in combination with video.

The term "subtitle" in this document is used to mean a visual presentation that is provided synchronized with video and audio tracks. Subtitles are presented for various purposes including dialog language translation, content description, "closed captions" for deaf and hard of hearing, and other purposes.

Subtitle tracks are defined with a new media type and media handler, comparable to audio and video media types and handlers. Subtitle tracks use a similar method to store and access timed "samples" that span durations on the Movie timeline and synchronize with other tracks selected for presentation on that timeline using the basic media track synchronization method of ISO Base Media File Format.

CFF subtitles are defined using the Timed Text Markup Language (TTML), as defined by the [SMPTE-TT] standard, which is derived from the W3C "Timed Text Markup Language" [W3C-TT] standard. With this approach, [SMPTE-TT] XML documents control the presentation of subtitles during their sample duration, analogous to the way an ISO media file audio sample contains a sync frame or access unit of audio samples and presentation information specific to each audio codec that control the decoding and presentation of the contained audio samples during the longer duration of the ISO media file sample.

The [W3C-TT] standard is an XML markup language-primarily designed for the presentation and interchange of character coded text using font sets (text subtitles). The [SMPTE-TT] standard extends the [W3C-TT] standard to support the presentation of stored bitmapped images (image subtitles) and to support the storage of data streams for legacy subtitle and caption formats (e.g. CEA-608).

Text and image subtitles each have advantages for subtitle storage and presentation, so it is useful to have one common subtitling format that is capable of providing either a text subtitle stream or an image subtitle stream.

Advantages of text subtitling include:

- Text subtitles require minimal size and bandwidth
- Devices can present text subtitles with different styles, sizes, and layouts for different displays, viewing conditions and user preferences
- Text subtitles can be converted to speech and tactile readouts (for visually impaired)\
- Text subtitles are searchable

Advantages of image subtitling include:

- Image subtitles enable publishers to fully control presentation of characters (including glyphs, character layout, size, overlay etc.)
- Image subtitles enable publishers to add graphical elements and effects to presentation
- Image subtitles provide a consistent subtitling presentation across all playback environments

CFF subtitle tracks can be either text subtitle tracks or image subtitle tracks, i.e. the mixing of text and image subtitles within one track is not supported.

In order to optimize streaming, progressive playback, and random access user navigation of video and subtitles, [ISOTEXT] and this specification define how [SMPTE-TT] documents are stored as multiple documents in an ISO Base Media Track and how, in the case of an image subtitle track, associated image files are stored as multiple files in an ISO Base Media Track. Image files are stored separately as Items in each sample and referenced from an adjacent [SMPTE-TT] document in order to limit the maximum size of each document, which will decrease download time and player memory requirements.

6.2 CFF-TT Document Format

6.2.1 Definition

CFF-TT documents SHALL conform to the SMPTE Timed Text specification [SMPTE-TT], with the additional constraints defined in this specification.

6.2.2 CFF-TT Text Encoding

CFF-TT documents SHALL use UTF-8 character encoding as specified in [UNICODE]. All Unicode Code Points contained within CFF-TT documents SHALL be interpreted as defined in [UNICODE].

6.2.3 CFF Timed Text Profiles

The [SMPTE-TT] format provides a means for specifying a collection of mandatory and optional features and extensions that must or might be supported. This collection is referred to as a Timed Text Profile. In order to facilitate interoperability, this specification defines the CFF Timed Text Profiles derived from the SMPTE TT Profile defined in [SMPTE-TT].
Two Timed Text Profiles are defined by this specification – text and image. CFF-TT documents SHALL conform to either the text profile (see Section 6.2.2.4) or image profile (see Section 6.2.2.5). Note that the mixing of text and image subtitles within one CFF-TT document is not supported.

6.2.3.1 CFF TTML Extension – forcedDisplayMode

6.2.3.1.1 Definition

The `forcedDisplayMode` TTML extension is defined to support the signaling of a block of subtitle content that is identified as "Forced" subtitle content. "Forced" subtitle content is subtitle content that represents audio (e.g. foreign language) or text (e.g. a sign) that is not translated in the audio/video presentation.

6.2.3.1.2 XML Namespace

http://www.decellc.org/schema/2012/01/cff-tt-meta
The recommended prefix for this namespace is "`cff:`".

6.2.3.1.3 XML Definition

Values:	false | true

Initial:	false
Applies to:	body, div, p, region, span
Inherited:	yes
Percentages:	N/A
Animatable:	discrete

Note: Although the `forcedDisplayMode` attribute, like all the TTML style attributes, has no defined semantics on a <br> content element, `forcedDisplayMode` will apply to a <br> content element if it is either defined on an ancestor content element of the <br> content element or it is applied to a region element corresponding to a region that the <br> content element is being flowed into.

The `forcedDisplayMode` TTML extension is an `xs:Boolean` datatype attribute.

6.2.3.1.4 XML Schema Document

URI reference: `cff-tt-meta-{DMEDIA_VERSION_POINTS}.xsd`

Notes:

- {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex A.
- In any case where the XML schema document conflicts with this specification, this specification is authoritative.
- URI is defined in [RFC3986]

6.2.3.1.5 XML Example Snippet

```
<div>
   <p region="subtitle1" begin="00:05:00" end="00:05:15"
cff:forcedDisplayMode="true">
        This subtitle is forced.
   </p>
</div>
```

6.2.3.1.6 Layout and Flow

When only “Forced” subtitle content is displayed, content not signaled `forcedDisplayMode=”true”` SHALL be hidden but used for layout and flow i.e. this content will be equivalent to `tts:visibility=”hidden”`. Note that setting `tts:visibility=”hidden”` on all elements within a region will leave the region visible if it has `tts:showBackground=”always”` (the default).

6.2.3.2 CFF TTML Extension – progressivelyDecodable

6.2.3.2.1 Definition

The `progressivelyDecodable` TTML extension is defined to support the signaling of whether the document can be progressively decoded. When set to true, this extension signals that the document has been designed to be progressively decodable by a decoder. A document that includes a `ttp:progressivelyDecodable` attribute with value `"true"` on the `<tt>` element SHALL conform to the following:

1. no attribute or element of the TTML timing vocabulary SHALL be present within the `<head>` element; and
2. given two Intermediate Synchronic Documents A and B with presentation times TA and TB, respectively, TA SHALL NOT be greater than TB if A maps to a `<p>` element that occurs earlier in the document than any `<p>` element to which B maps; and
3. child elements of <p> SHALL NOT have an attribute of the TTML timing vocabulary; and
4. no element SHALL reference another element that occurs after it in the document.

Notes:

- Elements with identical resolved begin times need to be in the order desired for flow.

6.2.3.2.2 XML Namespace

http://www.decellc.org/schema/2012/01/cff-tt-meta

The recommended prefix for this namespace is “cff:”.

6.2.3.2.3 XML Definition

Values:	false | true
Initial:	false
Applies to:	tt
Inherited:	no
Percentages:	N/A
Animatable:	N/A

The `progressivelyDecodable` TTML extension is an `xs:Boolean` datatype attribute.

6.2.3.2.4 XML Schema Document

URI reference: `cff-tt-meta-{DMEDIA_VERSION_POINTS}.xsd`

Notes:

- {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.
- In any case where the XML schema document conflicts with this specification, this specification is authoritative.
- URI is defined in [RFC3986]

6.2.3.2.5 XML Example Snippet

```
<tt
    xmlns="http://www.w3.org/ns/ttml"
    xmlns:ttm="http://www.w3.org/ns/ttml#metadata"
    xmlns:tts="http://www.w3.org/ns/ttml#styling"
    xmlns:ttp="http://www.w3.org/ns/ttml#parameter"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cff="http://www.decellc.org/schema/2012/01/cff-tt-meta"
    xsi:schemaLocation="http://www.w3.org/ns/ttml cff-tt-text-ttaf1-dfxp-
{DMEDIA_VERSION_POINTS}.xsd http://www.decellc.org/schema/2012/01/cff-tt-meta cff-
tt-meta-{DMEDIA_VERSION_POINTS}.xsd"
    xml:lang="es"
    ttp:frameRate="24"
    ttp:frameRateMultiplier="1000 1001"
    ttp:timeBase="media"
    cff:progressivelyDecodable=”true”
>
```

Note:

- {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.

6.2.3.3 General Profile Restrictions

6.2.3.3.1 Feature Restrictions

The following TTML restrictions SHALL apply to all CFF Timed Text documents.

Table 6-1 – CFF General TTML Feature Restrictions

FEATURE	CONSTRAINT
`#cellResolution`	SHALL NOT be used.
`#clockMode`	SHALL NOT be used.
`#color`	Note: As required in [SMPTE-TT], the initial value of `tts:color` is "white".
`#dropMode`	SHALL NOT be used.
`#extent-region`	• The maximum size SHALL be specified and SHALL be smaller than or equal to the root container. • regions presented in the same Subtitle Event SHALL NOT overlap (see Section 6.6 for a definition of “Subtitle Event” and Section 6.2.2.3.2 for a definition of when a region is “presented”).
`#extent-root`	• If present on the `<tt>` element, `tts:extent` SHALL be equal to the width and height of the associated video track. •If not specified, the spatial extent of the root container is as defined in Section 6.2.3. • If unit of measure px (pixel) length values are used in the CFF-TT document, then `tts:extent` SHALL be present on the `<tt>` element and comply with the constraint defined above.
`#frameRate`	• If specified, `ttp:frameRate` and `ttp:frameRateMultiplier` attributes SHALL collectively match the frame rate of the associated video track. • If not specified, the frame rate SHALL be the frame rate of the associated video track.

#frameRateMultiplier	• If specified, ttp:frameRate and ttp:frameRateMultiplier attributes SHALL collectively match the frame rate of the associated video track. • If not specified, the frame rate SHALL be the frame rate of the associated video track.
#length	The unit of measure px (pixel) SHALL be the same unit of measure as that used for the associated video track.
#length-cell	SHALL NOT be used.
#length-negative	SHALL NOT be used.
#length-percentage	The "pixel" value equated with any "percentage" length value SHALL be a pixel on the CFF-TT Coordinate System. To calculate the pixel, the "round to nearest" rounding algorithm SHALL be utilized with the "round half-up" tie break rule applied. Note: see Section 6.2.3 for more details on the CFF-TT Coordinate System.
#markerMode	SHALL NOT be used.
#origin	• regions SHALL be contained within the root container. • regions presented in the same Subtitle Event SHALL NOT overlap (see Section 6.6 for a definition of "Subtitle Event" and Section 6.2.2.3.2 for a definition of when a region is "presented"). Note: per the #length-cell restriction defined above, it is prohibited to use "c" (cell) scalar unit representations.
#overflow	SHALL NOT be used.
#pixelAspectRatio	SHALL NOT be used
#subFrameRate	SHALL NOT be used.
#tickRate	•if specified, ttp:tickRate SHALL be set to the same value as that of the timescale parameter in the subtitle track's Media Header Box ('mdhd'). • if #time-offset-with-ticks expressions timeExpression values are used in the CFF-TT document, ttp:tickRate SHALL be present on the <tt> element and comply with the constraint defined above.
#timeBase-clock	SHALL NOT be used.
#timeBase-media	timeBase SHALL be "media" where time zero is the start of the subtitle track decode time on the media timeline. Note that time zero does not reset with every subtitle fragment and media time is accumulated across subtitle fragments.
#timeBase-smpte	SHALL NOT be used.

#timing	• The same syntax (clock-time or offset-time) SHOULD be used throughout the CFF-TT document. • Offset time expressions using the tick metric SHOULD NOT use fractional ticks. • Explicitly defined timing SHALL NOT extend beyond the time span of the CFF-TT document's subtitle sample on the ISO media timeline. •Note: #time-offset-with-frames expressions are translated to media time with the following equation (where M is the media time in seconds): $\mathrm{M} = 60^2 \times \text{hours} + 60 \times \text{minutes} + \text{seconds} + (\text{frames} \div (\text{ttp:frameRateMultiplier} \times \text{ttp:frameRate}))$ • Note: #time-offset-with-ticks expressions are calculated from media time with the following equation (where M is the media time in seconds): $\text{Tick} = \text{ceiling}(\mathrm{M} \times \text{ttp:tickRate})$

6.2.3.3.2 Element Restrictions

The following TTML restrictions SHALL apply to all CFF Timed Text documents.

Table 6-2 – CFF General TTML Element Restrictions

ELEMENT	CONSTRAINT
body	All content presented in a Subtitle Event SHALL be associated with a Document Instance content region i.e. such content SHALL NOT be directly placed in the Root Container Region.
region	Number of regions presented in the same Subtitle Event SHALL be <=4 (see Section 6.6 for a definition of Subtitle Event). A region SHALL be considered “presented” if all four of the following are true: 1) The region does not have a tts:opacity="0.0" (note that “1.0” is the default value of the tts:opacity attribute); and 2) The region does not have a tts:visibility="hidden" (note that “visible” is the default value of the tts:visibility attribute); and 3) The region does not have a tts:display="none" (note that "auto" is the default value of the tts:display attribute); and 4) content is selected into the region at the time of the Subtitle Event, or the region has a tts:showBackground=“always” and a tts:backgroundColor with non-transparent alpha (note that “always” is the default value of the tts:showBackground attribute).
tt	The <tt> element SHALL include an xmlns attribute with or without a prefix) with “http://www.w3.org/ns/ttml”

6.2.3.3.3 Attribute Restrictions

The following TTML restrictions SHALL apply to all CFF Timed Text documents.

Table 6-3 – General TTML Attribute Restrictions

ELEMENT	CONSTRAINT
`xml:lang`	If specified, the `xml:lang` attribute SHALL match the Subtitle/Language Multi-Track Required Metadata (see Section 2.1.2.1) if Multi-Track Required Metadata is present in the DCC. Note: `xml:lang` MAY be set to an empty string.

6.2.3.4 Text Subtitle Profile

6.2.3.4.1 XML Schema Document

URI reference: “`cff-tt-text-ttaf1-dfxp-{DMEDIA_VERSION_POINTS}.xsd`”
Notes:

- {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.
  Note: In any case where the XML schema document conflicts with this specification, this specification is authoritative.
- URI is defined in [RFC3986]

6.2.3.4.2 xsi:schemaLocation

CFF Timed Text documents contained within a text subtitle track SHOULD have an `xsi:schemaLocation` attribute defined on the `<tt>` element.
The value of this `xsi:schemaLocation` attribute is to be set as follows:

- “http://www.w3.org/ns/ttml cff-tt-text-ttaf1-dfxp-{DMEDIA_VERSION_POINTS}.xsd” SHOULD be included.
- If the `forcedDisplayMode` extension defined in Section 6.2.2.1 or the `progressivelyDecodable` extension defined in Section 6.2.2.2 is used in the document, “http://www.decellc.org/schema/2012/01/cff-tt-meta cff-tt-meta-{DMEDIA_VERSION_POINTS}.xsd” SHOULD be included.
- The built-in XML Schema namespaces “http://www.w3.org/2001/XMLSchema” and “http://www.w3.org/2001/XMLSchema-instance” and any namespace declaration which has a prefix beginning with the three-letter sequence "xml" SHOULD NOT be included.
- All other schemas for all the namespaces declared in the document SHOULD be included with the following exception: if a schema defines multiple namespaces, it SHOULD only be present in the `xsi:schemaLocation` once (for example, only ...ns/ttml is recommended to be included, not both ...ns/ttml and ...ns/ttml#style).

Note: {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.

6.2.3.4.3 Feature restrictions

In addition to the restrictions defined in Section 6.2.2.3.1, the following restrictions SHALL apply to CFF Timed Text documents contained within a text subtitle track.

Table 6-4 - CFF Text Subtitle TTML Feature Restrictions

FEATURE	CONSTRAINT

`#extent-region`	• length expressions SHALL use "px" (pixel) scalar units or "percentage" representation. "em" (typography unit of measure) SHALL NOT be used. Note: per the #length-cell restriction defined in Table 6-1, it is prohibited to use "c" (cell) scalar unit representations. • SHOULD be large enough for text content layout without clipping in accordance with the Hypothetical Render Model defined in Section 6.6.4.2.
`#fontFamily`	• A `tts:fontFamily` of either “monospaceSerif” or “proportionalSansSerif” SHOULD be specified for all presented text content. • A `tts:fontFamily` of “default” SHALL be equivalent to “monospaceSerif”.
`#fontSize-anamorphic`	SHALL NOT be used.
`#origin`	"em" (typography unit of measure) SHALL NOT be used. Note: per the #length-cell restriction defined above, it is prohibited to use "c" (cell) scalar unit representations.
`#profile`	A document SHALL contain a ttp:profile element under the <head> element, where the use attribute of that element is specified “http://www.decellc.org/profile/cff-tt-text-{DMEDIA_VERSION_POINTS}”. Note: {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.
`#textOutline`	If specified, the border thickness SHALL be 10% or less than the associated font size.
`#textOutline-blurred`	SHALL NOT be used.

6.2.3.4.4 SMPTE Extension Restrictions

In addition to the restrictions defined in Section 6.2.2.3.1, the following restrictions SHALL apply to CFF Timed Text documents contained within an text subtitle track.

Table 6-5 - CFF Text Subtitle TTML SMPTE Extension Restrictions

EXTENSION	CONSTRAINT
`#backgroundImage`	SHALL NOT be used.
`#backgroundImageHorizontal`	SHALL NOT be used.
`#backgroundImageVertical`	SHALL NOT be used.
`#image`	SHALL NOT be used.

6.2.3.5 Image Subtitle Profile

6.2.3.5.1 XML Schema Document

URI reference: “`cff-tt-image-ttaf1-dfxp-{DMEDIA_VERSION_POINTS}.xsd`”

URI reference: “`cff-tt-image-smpte-tt-{DMEDIA_VERSION_POINTS}.xsd`”

Notes:

- {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.
- In any case where the XML schema document conflicts with this specification, this specification is authoritative.
- URI is defined in [RFC3986]

6.2.3.5.2 xsi:schemaLocation

CFF Timed Text documents contained within an image subtitle track SHOULD have an `'xsi:schemaLocation'` attribute defined on the `'<tt>'` element. The value of this `xsi:schemaLocation` attribute is to be set as follows:

- "http://www.w3.org/ns/ttml cff-tt-image-ttaf1-dfxp-{DMEDIA_VERSION_POINTS}.xsd" and http://www.smpte-ra.org/schemas/2052-1/2010/smpte-ttcff-tt-image-smpte-tt-{DMEDIA_VERSION_POINTS}.xsd" SHOULD be included.
- If the `forcedDisplayMode` extension defined in Section 6.2.2.1 or the `progressivelyDecodable` extension defined in Section 6.2.2.2 is used in the document, "http://www.decellc.org/schema/2012/01/cff-tt-meta cff-tt-meta-{DMEDIA_VERSION_POINTS}.xsd" SHOULD be included.
- The built-in XML Schema namespaces "http://www.w3.org/2001/XMLSchema" and "http://www.w3.org/2001/XMLSchema-instance" and any namespace declaration which has a prefix beginning with the three-letter sequence "xml" SHOULD NOT be included.
- All other schemas for all the namespaces declared in the document SHOULD be included with the following exception: if a schema defines multiple namespaces, it SHOULD only be present in the `xsi:schemaLocation` once (for example, only ...ns/ttml is recommended to be included, not both ...ns/ttml and ...ns/ttml#style).

Note: {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.

6.2.3.5.3 Feature Restrictions

In addition to the restrictions defined in Section 6.2.2.3.1, the following restrictions SHALL apply to CFF Timed Text documents contained within an image subtitle track.

Table 6-6 - CFF Image Subtitle TTML Feature Restrictions

FEATURE	CONSTRAINT
`#bidi`	SHALL NOT be used.
`#color`	SHALL NOT be used.
`#content`	`<p>`, `<span>`, `<br>` SHALL NOT be used.
`#direction`	SHALL NOT be used.
`#displayAlign`	SHALL NOT be used.
`#fontFamily`	SHALL NOT be used.
`#fontSize`	SHALL NOT be used.
`#fontStyle`	SHALL NOT be used.
`#fontWeight`	SHALL NOT be used.
`#length-em`	SHALL NOT be used.
`#lineBreak-uax14`	SHALL NOT be used.
`#lineHeight`	SHALL NOT be used.
`#nested-div`	SHALL NOT be used.
`#nested-span`	SHALL NOT be used.
`#padding`	SHALL NOT be used.

#profile	A document SHALL contain a ttp:profile element where the use attribute of that element is specified "http://www.decellc.org/profile/cff-tt-image-{DMEDIA_VERSION_POINTS}". Note: {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.
#textAlign	SHALL NOT be used.
#textDecoration	SHALL NOT be used.
#textOutline	SHALL NOT be used.
#wrapOption	SHALL NOT be used.
#writingMode-vertical	SHALL NOT be used.

6.2.3.5.4 Element Restrictions

In addition to the restrictions defined in Section 6.2.2.3.1, the following restrictions SHALL apply to CFF Timed Text documents contained within an image subtitle track.

Table 6-7 – CFF Image Subtitle TTML Element Restrictions

ELEMENT	CONSTRAINT
div	If a smpte:backgroundImage attribute is applied to a <div>, the width and height of the region extent associated with the <div> SHALL equate to the width and height of the image source referenced by the smpte:backgroundImage. Note: see the #length-percentage constraint in Table 6-1 for more information on equating a "percentage" length representation of a region to pixels in the image source referenced by the smpte:backgroundImage.
region	For each Subtitle Event, there SHALL be at most one <div> with the smpte:backgroundImage attribute applied associated with any "presented" region (see Section 6.6 for a definition of "Subtitle Event" and Section 6.2.2.3.2 for a definition of when a region is "presented).

6.2.3.5.5 SMPTE Extension Restrictions

In addition to the restrictions defined in Section 6.2.2.3.1, the following TTML restrictions SHALL apply to CFF Timed Text documents contained within an image subtitle track.

Table 6-8 - CFF Image Subtitle TTML SMPTE Extension Restrictions

EXTENSION	CONSTRAINT
#backgroundImageHorizontal	SHALL NOT be used. Note: the smpte:backgroundImage attribute remains available for use.
#backgroundImageVertical	SHALL NOT be used. Note: the smpte:backgroundImage attribute remains available for use.
#image	<smpte:image> SHALL NOT be used.

6.2.4 CFF-TT Coordinate System

The root container origin SHALL be "0 0". The spatial extent of the CFF-TT root container SHALL be determined using the width and height specified in the CFF-TT document Track Header Box (`'tkhd'`) and the width and height of the associated video track in accordance with [ISOTEXT]. In addition, the matrix values in the video and subtitle track headers are the default value. The position of the subtitle display region is determined on the notional 'square' (uniform) grid defined by the root container extent. The display region `'tts:origin'` values determine the position, and the `'tts:extent'` values determine the size of the region. Figure 6-1 illustrates an example of the subtitle display region position.

Note: Subtitles can only be placed within the encoded video active picture area. If subtitles need to be placed over black matting areas, the additional matting areas need to be considered an integral part of the video encoding and included within the video active picture area for encoding.

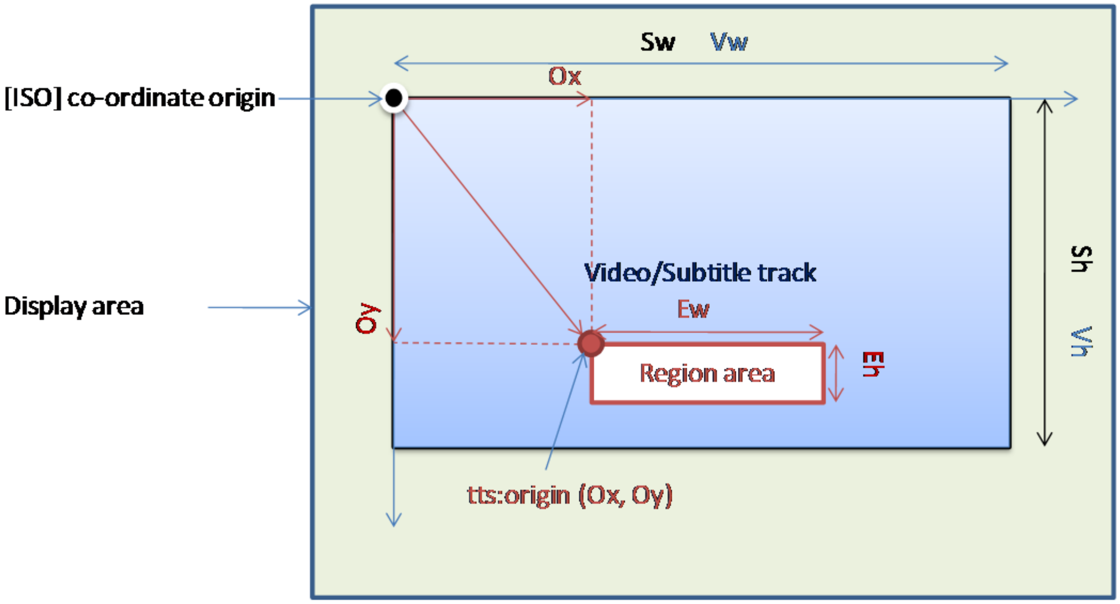


Figure 6-1 – Example of subtitle display region position

In Figure 6-1, the parameters are denoted as follows:

- Vw, Vh – Video track header width and height, respectively.
- [ISO] co-ordinate origin - the origin of the CFF-TT root container.
- Sw, Sh – Subtitle track width and height, respectively which is also the spatial extent of the CFF-TT root container.
- Ew, Eh – CFF-TT display region `'tts:extent'`.
- Ox, Oy – CFF-TT display region `'tts:origin'`.
- Region area – area defined in the CFF-TT document that sets the rendering in which text is flowed or images are drawn.
- Display area – rendering area of the CFF-TT processor.

6.2.5 CFF-TT External Time Interval

The CFF-TT Document's External Time Interval SHALL equal the duration of the subtitle track on the ISO media timeline. The external time interval is the temporal beginning and ending of a document instance as specified in [W3C-TT] and incorporated in [SMPTE-TT].

6.3 CFF-TT Subtitle Event and Video Frame Synchronization

CFF-TT is designed to synchronize with video at the video frame level - that is, Subtitle Events (see Section 6.6) will first be displayed on a specific frame of video on the video frame grid and will be removed on a specific frame of video on the video frame grid. The following equation is used to calculate the video frame represented by a media time value calculated from a '<timeExpression>' value in a CFF-TT document (where M is the media time in seconds):

$$\mathrm{F} = \mathrm{ceiling}\big(\mathrm{M} \times (\mathrm{ttp{:}\,frameRateMultiplier} \times \mathrm{ttp{:}\,frameRate})\big)$$

In order to determine the video frame with which a Subtitle Event is actually first displayed or removed from a '<timeExpression>' value in a CFF-TT document, the video frame SHALL be calculated from the '<timeExpression>' value and the timing model defined in Section 6.6.2 SHALL be applied.

Note: Section 6.2.2.3 requires that the value of 'ttp:frameRate' is that of the video track (and if set in the document it is required to be equal to the video track framerate).

6.4 CFF-TT Encoded Image Format

Images SHALL conform to PNG image coding as defined in Sections 7.1.1.3 and 15.1 of [MHP], with the following additional constraints:

- PNG images SHALL NOT be required to carry a pHYs chunk indicating pixel aspect ratio of the bitmap. If present, the pHYs chunk SHALL indicate square pixels.

Note: If no pixel aspect ratio is carried, the default of square pixels will be assumed.

6.5 CFF-TT Structure

6.5.1 Track Format

A CFF subtitle track is either a text subtitle track or an image subtitle track.

Text subtitle tracks SHALL contain one or more CFF-TT XML documents all of which are compliant with Section 6.2.2.4. Text subtitle tracks SHALL NOT contain any image data. Text subtitle tracks SHALL comply with [ISOTEXT].

Image subtitle tracks SHALL contain one or more CFF-TT XML documents, all of which are compliant with Section 6.2.2.5. CFF-TT documents in image subtitle tracks SHALL incorporate images in their presentation by reference only and images are not considered within the document size limit. In this case, referenced images SHALL be stored in the same sample as the document that references them and SHALL NOT exceed the maximum sizes specified in in Table 6-9. Image subtitle tracks SHALL comply with [ISOTEXT].

Note:

- Per [ISOTEXT], each CFF-TT document in a CFF subtitle track is stored in a single subtitle sample which is a "sync sample", meaning that it is independently decodable.
- "sync sample" in movie fragments cannot be signaled by the absence of the Sync Sample box ('stss') or by the presence of the Sync Sample box ('stss'), since this box is not designed to list sync samples in movie fragments. Instead, signaling can be achieved by other means such as setting the 'sample_is_non_sync_sample' flag to "0" in the 'default_sample_flags' field in the Track Extends box ('trex').

6.5.2 Sample Format

6.5.2.1 Definition

Subtitle sample storage SHALL comply with [ISOTEXT]. In image subtitle tracks, each subtitle sample SHALL also contain all images referenced in the CFF-TT document and storage of images SHALL comply with [ISOTEXT]. Each subtitle track fragment SHALL contain exactly one subtitle sample.

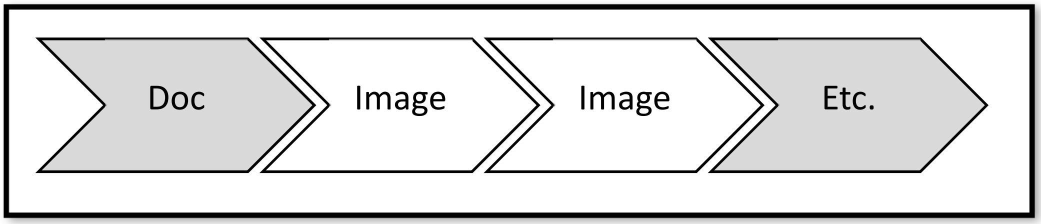


Figure 6-2 – Storage of images following the related SMPTE TT document in a sample

6.5.2.2 Images

Image formats used for subtitles (i.e. PNG) SHALL be specified in a manner such that all of the data necessary to independently decode an image (i.e. color look-up table, bitmap, etc.) is stored together within a single sub-sample.

The total size of image data stored in a sample SHALL NOT exceed the values indicated in Table 6-9. “Image data” SHALL include all data in the sample except for the CFF-TT document, which SHALL be stored at the beginning of each sample to control the presentation of any images in that sample.

The CFF-TT document SHALL reference each image using a URN, per [ISOTEXT], of the form:

`urn:dece:container:subtitleimageindex:<index>.<ext>`

Where:

- `<index>` is as defined in [ISOTEXT].
- `<ext>` is as defined in [ISOTEXT].

Note: A CFF-TT document might reference the same image multiple times within the document. In such cases, there will be only one sub-sample entry in the Sub-Sample Information Box (`‘subs’`) for that image, and the URI used to reference the image each time will be the same. However, if an image is used by multiple CFF-TT documents, that image is required to be stored once in each sample for which a document references it.

An example of image referencing is shown below:

```
<head>
  <layout>
    <region tts:extent="250px 50px" tts:origin="200px 800px" xml:id="r1"/>
    <region tts:extent="200px 50px" tts:origin="200px 800px" xml:id="r2"/>
  </layout>
</head>
<body>
  <div region="r1"
smpte:backgroundImage="urn:dece:container:subtitleimageindex:1.png"/>
  <div region="r2"
smpte:backgroundImage="urn:dece:container:subtitleimageindex:2.png"/>
```

```
</body>
```

6.5.2.3 Constraints

CFF-TT subtitle samples SHALL NOT exceed the following constraints:

Table 6-9 – Constraints on Subtitle Samples

Property	Constraint
CFF-TT document size	Single XML document size <= 500 x 2^{10} bytes

6.6 CFF-TT Hypothetical Render Model

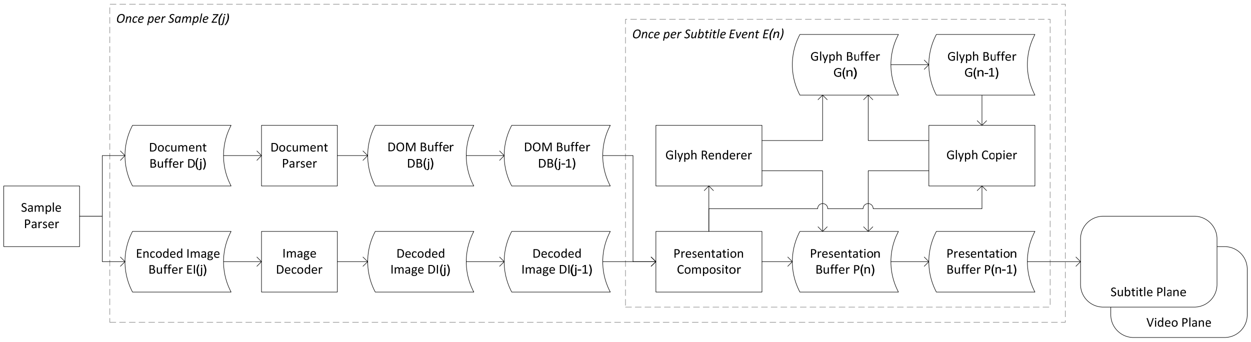


Figure 6-3 – Block Diagram of Hypothetical Render Model

This Section defines the CFF-TT Hypothetical Render Model. CFF-TT documents SHALL NOT exceed the limits and constraints defined by this model.

6.6.1 Functional Model

The hypothetical render model for CFF-TT subtitles is shown in Figure 6-3. It includes separate input buffers $D_{(j)}$ and $EI_{(j)}$ for one CFF-TT document, and a set of images contained in one sample, respectively. Each buffer has a minimum size determined by the maximum document and sample size specified.

The Document Object Model (DOM) buffers, $DB_{(j)}$ and $DB_{(j-1)}$, store the DOMs produced by parsing a CFF-TT document. DOM buffers do not have a specified size because the amount of memory required to store compiled documents depends on how much memory a media handler implementation uses to represent them. A-CFF-TT processor implementation can determine a sufficient size based on document size limits and worst-case code complexity.

The model includes two DOM buffers in order to enable the processing and presentation of the currently active CFF-TT document in $DB_{(j-1)}$ while the next CFF-TT document is received and parsed in $DB_{(j)}$ in preparation for it becoming active. See Section 6.6.2 for more information on the timing model of when documents are active and inactive.

For the purposes of performing presentation processing, the active time duration of the CFF-TT document is divided into a sequence of Subtitle Events. For any given Subtitle Event $E_{(n)}$, all visible pixels for Subtitle Event $E_{(n)}$ are painted.

A Subtitle Event SHALL occur whenever there is any change to subtitle presentation. Each Subtitle Event is associated with an intermediate synchronic document. [W3C-TT] Section 9.3.2, as incorporated by [SMPTE-TT], dictates when an intermediate synchronic document is constructed. Note: A change to subtitle presentation caused by the <set> animation element will result in a new Subtitle Event.

The Presentation Compositor retrieves presentation information for each Subtitle Event from the applicable Doc DOM (according to the current subtitle fragment); presentation information includes presentation time, region positioning, style information, etc. associated with the Subtitle Event. The Presentation Compositor constructs an intermediate synchronic document for the Subtitle Event, in accordance with [W3C-TT] Section 9.3.2, as incorporated by [SMPTE-TT], and paints the corresponding Subtitle Event into the Presentation Buffer $P_{(n)}$.

The Glyph Buffers $G_{(n)}$ and $G_{(n-1)}$ are used to store rendered glyphs across Subtitle Events, allowing glyphs to be copied into the Presentation Buffer instead of rendered, a more costly operation. This enables scenarios where the same glyphs are used in multiple successive Subtitle Events, e.g. to convey a CEA-608/708-style roll-up. To paint Subtitle Event $E_{(n)}$, the Presentation Compositor has access in Glyph Buffer $G_{(n-1)}$ to the glyphs used during Subtitle Event $E_{(n-1)}$ and in Glyph Buffer $G_{(n)}$ to all glyphs used during Subtitle Event $E_{(n)}$. Once processing of a Subtitle Event is completed, the Presentation Buffer $P_{(n)}$ is copied to $P_{(n-1)}$ and the Glyph Buffer $G_{(n)}$ to $G_{(n-1)}$.

The Presentation Buffer $P_{(n)}$ acts as a “back buffer” in the model (the “back buffer” is the secondary buffer in this “double buffer” model – it is used to store the result of every paint operation involved in creating the Subtitle Event but it is not used for the display of Subtitle Event in this model).

The Presentation Buffer $P_{(n-1)}$ stores a Subtitle Event available for display with video and acts as a “front buffer” in the model (the “front buffer” is the primary buffer in this “double buffer” model – it is used for the display of the completed Subtitle Event in this model).

The Video Plane stores each frame of decoded video. The Presentation Buffers $P_{(n)}$ and $P_{(n-1)}$, Subtitle Plane and Video Plane have the same horizontal and vertical size as the CFF-TT root container.

After video/subtitles have been composited, the resulting image is then provided over external video interfaces if any and/or presented on an integrated display.

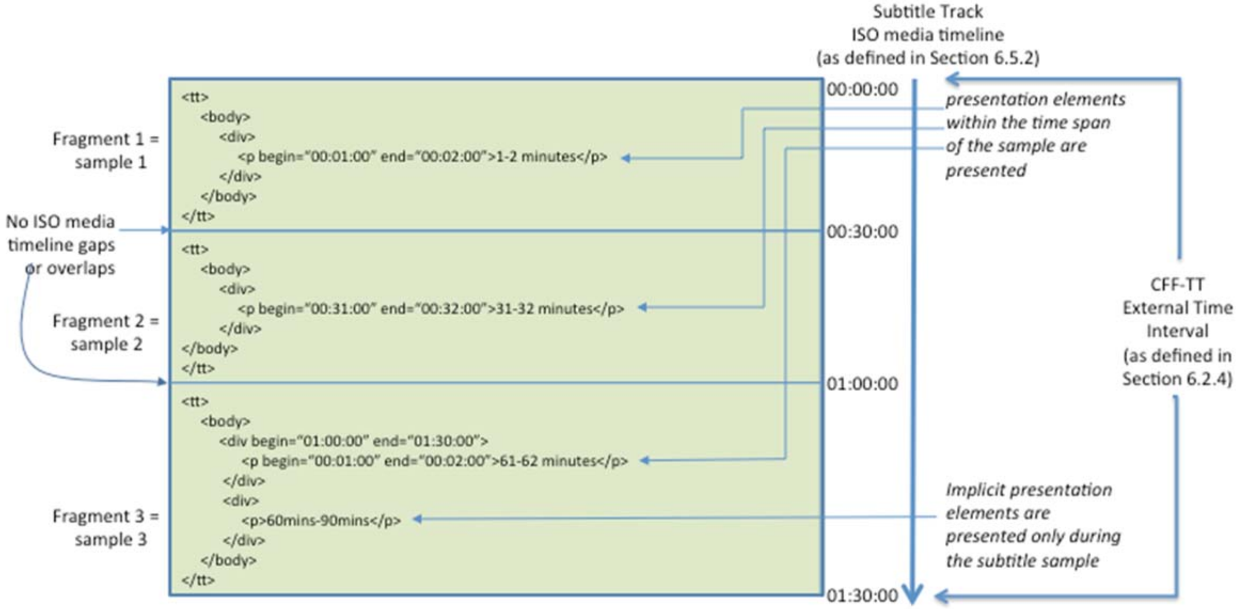
The above provides an overview of a hypothetical model only. Any CFF-TT processor implementation of this model is allowed as long as the observed presentation behavior of this model is satisfied. In particular, some CFF-TT processor implementations might render/paint and scale to different resolutions than the SMPTE TT root container in order to optimize presentation for the display connected to (or integrated as part of) the CFF-TT processor implementation but in such cases CFF-TT processor implementations are required to maintain the same subtitle and video relative position (regardless of differences in resolution between the display and SMPTE TT root container).

6.6.2 Timing Model

Although, per Section 6.2.4 all CFF-TT Documents have an External Time Interval equal to the subtitle track duration, only one CFF-TT document is presented at any one point in time by the render model. The render model presents a CFF-TT document only when the CFF-TT document is active. A CFF-TT document is active only during the time span of its associated subtitle sample on the ISO media timeline and at all other times the CFF-TT document is inactive. Consequently all presentation defined in the CFF-TT document will be shown when the document is active. Any portion of presentation associated with a time when the document is inactive will not be presented with the following exception - if the document becomes inactive during a coded video frame, the presentation will continue until the next nearest coded video frame at which time any presentation defined in the CFF-TT document will not be shown.

This timing relationship is defined in [ISOTEXT] and depicted in Figure 6-4 below. Therefore, during playback of a subtitle track, at the end of a subtitle sample the Document associated with the subtitle sample will become inactive and the Document associated with the next subtitle sample, which is immediately adjacent on the ISO media timeline, will immediately become active at the start of the next subtitle sample – thus subtitle presentation will continue seamlessly over subtitle samples (and fragments) on the ISO media timeline without interruption to subtitle presentation.

Note: The time span of the subtitle sample always starts at the time represented by the sum of all previous subtitle sample durations and always lasts for the length of time represented by the sample_duration determined from the `default_sample_duration` and `sample_duration` fields associated with the subtitle sample.



```
<tt>
  <body>
    <div>
      <p begin="00:01:00" end="00:02:00">1-2 minutes</p>
    </div>
  </body>
</tt>
```

```
<tt>
  <body>
    <div>
      <p begin="00:31:00" end="00:32:00">31-32 minutes</p>
    </div>
</body>
</tt>
```

```
<tt>
  <body>
    <div begin="01:00:00" end="01:30:00">
      <p begin="00:01:00" end="00:02:00">61-62 minutes</p>
    </div>
    <div>
      <p>60mins-90mins</p>
    </div>
  </body>
</tt>
```

Subtitle Track
ISO media timeline
(as defined in Section 6.5.2)
Fragment 1 = sample 1
No ISO media timeline gaps or overlaps
Fragment 2 = sample 2
Fragment 3 = sample 3
00:00:00
00:30:00
01:00:00
01:30:00
presentation elements within the time span of the sample are presented
CFF-TT External Time Interval (as defined in Section 6.2.4)
Implicit presentation elements are presented only during the subtitle sample

Figure 6-4 – Time relationship between CFF-TT documents and the CFF-TT track ISO media timeline

The performance available for painting Subtitle Events is bounded by constraints on key aspects of the model, e.g. drawing and rendering rates – see Annex A, B and C. Whenever applicable, these constraints are specified relative to the root container dimensions, allowing CFF-TT Documents to be authored independently of video resolution.

The Presentation Compositor starts painting pixels for the first Subtitle Event in the CFF-TT document at the decode time of the subtitle fragment. If Subtitle Event $\mathrm{E_{(n)}}$ is not the first in a CFF-TT document, the Presentation Compositor starts painting pixels for Subtitle Event $\mathrm{E_{(n)}}$ at the "start time" of the immediately preceding Subtitle Event $\mathrm{E_{(n-1)}}$. All data for Subtitle Event $\mathrm{E_{(n)}}$ is painted to the Presentation Buffer for each Subtitle Event.

For each Subtitle Event, the Presentation Compositor clears the pixels within the root container (except for the first Subtitle Event $\mathrm{E_{(FIRST)}}$) and then paints, according to stacking order, all background pixels for each region, then paints all pixels for background colors associated with text or image subtitle content and then paints the text or image subtitle content. The Presentation Compositor needs to complete painting for the Subtitle Event $\mathrm{E_{(n)}}$ prior to the start time of Subtitle Event $\mathrm{E_{(n)}}$. The duration, in seconds, for painting a Subtitle Event in the Presentation Buffer is as follows for any given Subtitle Event $\mathrm{E_{(n)}}$within the CFF-TT document:

$$\mathrm{DURATION}\left(\mathrm{E_{(n)}}\right) = \frac{\mathrm{S_{(n)}}}{\mathrm{BDraw}} + \mathrm{C_{(n)}}$$

Where:

- $S_{(n)}$ is the normalized size of the total drawing area for Subtitle Event $E_{(n)}$, as defined below.
- BDraw is the normalized background drawing performance factor (see Annex A, B, C for the background drawing performance factor defined for each Profile).
- $C_{(n)}$ is the duration, in seconds, for painting the text or image subtitle content for Subtitle Event $E_{(n)}$. See the details defined in Section 6.7 and Section 6.8 below.

Note: BDraw effectively sets a limit on fillings regions - for example, assuming that the root container is ultimately rendered at 1920×1080 resolution, a BDraw of 12 s^{-1} would correspond to a fill rate of $1920 \times 1080 \times 12/\mathrm{s} = 23.7 \times 2^{20}\mathrm{pixels/s}$.

$S_{(FIRST)}$

The normalized size of the total drawing area for the first Subtitle Event $E_{(FIRST)}$ that is to be decoded by the CFF-TT processor implementation for the CFF-TT subtitle track is defined as:

$$\mathrm{S}_{(\mathrm{FIRST})} = \sum_{\mathrm{i}=0}^{\mathrm{i}<r} (\mathrm{NSIZE}(\mathrm{E}_{(\mathrm{FIRST})}.\mathrm{R}_{(\mathrm{i})}) \times \ \mathrm{TBG}_{(\mathrm{R}_{(\mathrm{i})})})$$

Where:

- r is the number of regions that are presented in this Subtitle Event. See Section 6.2.2.3.2 for a definition of when a region is considered to be presented.
- $\mathrm{NSIZE}(\mathrm{E}_{(\mathrm{FIRST})}.\mathrm{R}_{(\mathrm{i})})$ is equal to:

  $$(\text{width of } \mathrm{R}_{(i)} \times \text{height of } \mathrm{R}_{(i)}) \div (\text{root container height } \times \text{ root container width})$$

  $\mathrm{R}_{(\mathrm{i})}$is a region that will be presented in the Subtitle Event $\mathrm{E}_{(\mathrm{FIRST})}$.
- $\mathrm{TBG}_{(\mathrm{R}_{(\mathrm{i})})}$is the total number of `’tts:backgroundColor’` attributes associated with the given region $\mathrm{R}_{(\mathrm{i})}$ in this Subtitle Event (see “Notes about the model” below for a definition of when a `’tts:backgroundColor’` attribute is associated with a region in a Subtitle Event).

Example: For a region $\mathrm{R}_{(\mathrm{k})}$ with `tts:extent="250px 50px"` within a root container with `tts:extent="1920px 1080px"`, $\mathrm{NSIZE}(\mathrm{E}_{(\mathrm{FIRST})}.\mathrm{R}_{(\mathrm{k})})$ = 0.603.

$S_{(>FIRST)}$

The total normalized drawing area for Subtitle Event $\mathrm{E}_{(\mathrm{n})}$ after presentation of the first Subtitle Event $\mathrm{E}_{(\mathrm{FIRST})}$ is defined as:

$$\mathrm{S}_{(\mathrm{n})} = \mathrm{CLEAR}(\mathrm{E}_{(\mathrm{n})}) + \mathrm{PAINT}(\mathrm{E}_{(\mathrm{n})})$$

Where:

- $\mathrm{CLEAR}(\mathrm{E}_{(\mathrm{n})})$ = 1 and corresponds to the root container in its entirety.
- $\mathrm{PAINT}(\mathrm{E}_{(\mathrm{n})})$ is a function which calculates the normalized area that is to be painted for any regions that are used in Subtitle Event $\mathrm{E}_{(\mathrm{n})}$ in accordance with the following:

  $$\mathrm{PAINT}(\mathrm{E}_{(\mathrm{n})}) = \sum_{\mathrm{i}=0}^{\mathrm{i}<r} (\mathrm{NSIZE}(\mathrm{E}_{(\mathrm{n})}.\mathrm{R}_{(\mathrm{i})}) \ \times \ \mathrm{NBG}_{(\mathrm{R}_{(\mathrm{i})})})$$

  Where:

  - r is the number of regions that are presented in this Subtitle Event. See Section 6.2.2.3.2 for a definition of when a region is considered to be presented.
  - $\mathrm{NSIZE}(\mathrm{E}_{(\mathrm{n})}.\mathrm{R}_{(\mathrm{i})})$ is equal to:

$$\left(\text{width of } \mathrm{R}_{(i)} \times \text{height of } \mathrm{R}_{(i)}\right) \div (\text{root container height } \times \text{ root container width})$$

$\mathrm{R}_{(\mathrm{i})}$is a region that will be presented in the Subtitle Event $\mathrm{E}_{(\mathrm{n})}$.

- $\mathrm{NBG}_{(\mathrm{R}_{(\mathrm{i})})}$ is the total number of `'tts:backgroundColor'` attributes associated with the given region $\mathrm{R}_{(\mathrm{i})}$ in this Subtitle Event (see "Notes about the model" below for a definition of when a `'tts:backgroundColor'` attribute is associated with a region in a Subtitle Event).

At the "start time" of Subtitle Event $\mathrm{E}_{(\mathrm{n})}$, the content of the Presentation Buffer is instantaneously transferred to the Subtitle Plane and blended with video at the video frame corresponding to the "start time" of Subtitle Event $\mathrm{E}_{(\mathrm{n})}$ (or the subsequent video frame if the "start time" does not align with a frame of video on the video frame grid). The content of the Subtitle Plane is instantaneously cleared at the video frame corresponding to the "finish time" of Subtitle Event $\mathrm{E}_{(\mathrm{n})}$ (or the subsequent video frame if the "finish time" does not align with a frame of video on the video frame grid).

Notes about the model:

- To ensure consistency of the Presentation Buffer, a new Subtitle Event requires clearing of the root container.
- Each `'tts:backgroundColor'` attribute associated with a region in a Subtitle Event requires an additional fill operation for all region pixels.
  - ➢ A `'tts:backgroundColor'` attribute is associated with a region in a Subtitle Event when a `'tts:backgroundColor'` attribute is explicitly specified (either as an attribute in the element, or by reference to a declared style) in the following circumstances:
    - ▪ It is specified on the `'region'` layout element that defines the region.
    - ▪ It is specified on a `'div'`, `'p'`, `'span'` or `'br'` content element that is to be flowed into the region for presentation in the Subtitle Event (see [W3C-TT], as incorporated in [SMPTE-TT], for more details on when a content element is followed into a region).
    - ▪ It is specified on a `'set'` animation element that is to be applied to content elements that are to be flowed into the region for presentation in the Subtitle Event (see [W3C-TT], as incorporated in [SMPTE-TT], for more details on when a `'set'`animation element is applied to content elements).
  - ➢ Even if a specified `'tts:backgroundColor'` is the same as specified on the nearest ancestor content element or animation element, specifying any `'tts:backgroundColor'` will require an additional fill operation for all region pixels.
- The Presentation Compositor retains state over subtitle fragments i.e. when a subtitle fragment change occurs during presentation of a CFF-TT subtitle track, the first Subtitle Event in the CFF-TT document associated with the new subtitle fragment is treated as Subtitle Event $\mathrm{E}_{(\mathrm{n})}$ and the last Subtitle Event in the CFF-TT document associated with the previous subtitle fragment is treated as Subtitle Event $\mathrm{E}_{(\mathrm{n}-1)}$.
- It is possible for the content of Subtitle Event $\mathrm{E}_{(\mathrm{n})}$ to be overwritten in the Subtitle Plane with Subtitle Event $\mathrm{E}_{(\mathrm{n}+1)}$ prior to Subtitle Event $\mathrm{E}_{(\mathrm{n})}$ being composited with video - this would happen when the content of Subtitle Event $\mathrm{E}_{(\mathrm{n})}$was in the Subtitle Plane but had not yet been composited with video as a new frame of video had not yet been presented since the "start time" of Subtitle

Event $E_{(n)}$), and the “start time” of Subtitle Event $E_{(n+1)}$ occurred before the new frame of video was presented.

6.6.3 Image Subtitles

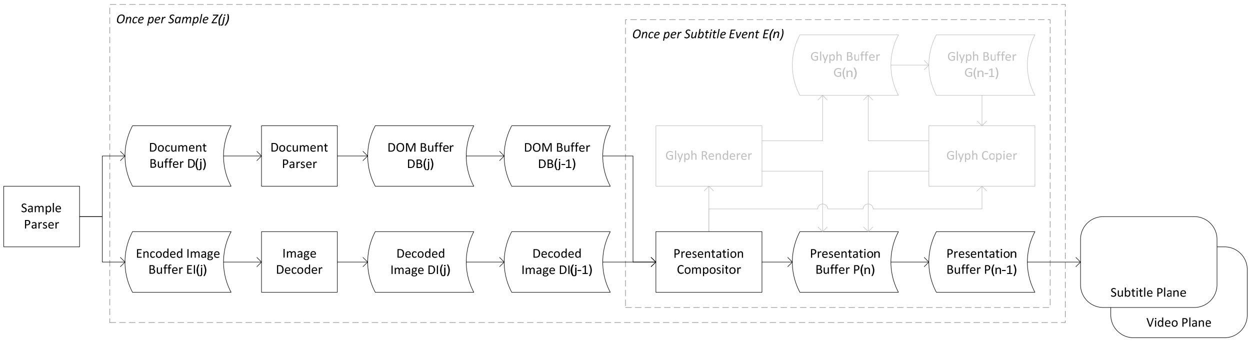


Figure 6-5 – Block Diagram of CFF-TT Image Subtitle Hypothetical Render Model

This section defines the performance model applied to CFF image subtitles.

In the model, encoded images are stored in the Encoded Image Buffer $E_{(j)}$. The Image Decoder decodes encoded images in the Encoded Image Buffer $E_{(j)}$ to the Decoded Image Buffer $DI_{(j)}$ with the image decoding rate (see Annex A, B, C for the image decoding rate defined for each Profile). Two Decoded Image Buffers, $DI_{(j)}$ and $DI_{(j-1)}$, are used in order to allow the Presentation Compositor to process the currently active CFF-TT document in $DI_{(j-1)}$ while the next CFF-TT document is being processed in $DI_{(j)}$ in preparation for presentation - this allows image subtitles referenced by CFF-TT documents from two consecutive samples/fragments to be displayed without delay. Note that both the “current” subtitle fragment and the “next” subtitle fragment MAY be acquired and decoded prior to presentation time.

The Presentation Compositor behaves as specified in Sections 6.6.1 and 6.6.2. The Presentation Compositor paints all pixels for images to be presented in the Subtitle Event using the corresponding raster data in the Decoded Image Buffer. The duration, in seconds, for painting a Subtitle Event in the Presentation Buffer is as follows for any given Subtitle Event $E_{(n)}$:

$$\mathrm{DURATION}\left(E_{(n)}\right) = \frac{S_{(n)}}{\mathrm{BDraw}} + C_{(n)}$$

For image-based CFF-TT subtitles, $C_{(n)}$ is as follows:

$$C_{(n)} = \sum_{i=0}^{i<nd} \frac{\mathrm{NSIZE}\left(E_{(n)}.I_{(i)}\right)}{\mathrm{ICpy}}$$

Where:

- nd is the number of div elements which have a `smpte:backgroundImage` attribute that is associated with a region which is presented in Subtitle Event $E_{(n)}$. See Section 6.2.2.3.2 for a definition of when a region is considered to be presented.
- $\mathrm{NSIZE}(E_{(n)}.I_{(i)})$ is equal to:

$$\left(\text{width of } I_{(i)} \times \text{height of } I_{(i)}\right) \div (\text{root container height } \times \text{ root container width})$$

$I_{(i)}$ is an image subtitle that will be presented in Subtitle Event $E_{(n)}$.

- ICpy is the normalized image copy performance factor (see Annex B and C for the image copy performance factor defined for each Profile).

Note: Image decoding performance is not included in the above equations as the model requires that the images associated with a subtitle fragment are decoded in full into one of the (two) decoded image buffers in advance of the start of the subtitle fragment presentation time.

6.6.4 Text Subtitles

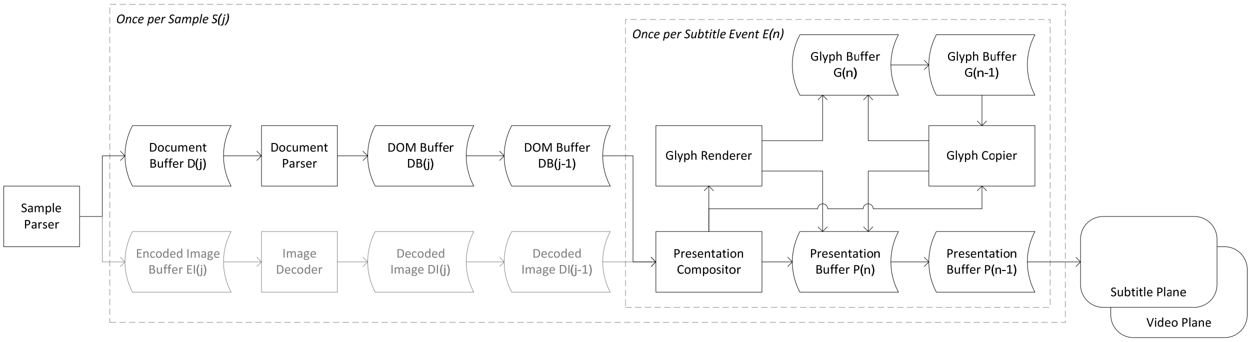


Figure 6-6 – Block Diagram of CFF-TT Text Subtitle Hypothetical Render Model

6.6.4.1 Performance Model

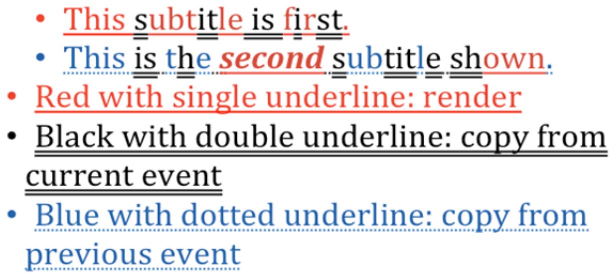
For each glyph displayed in Subtitle Event $E_{(n)}$, the Presentation Compositor will:

1) if an identical glyph is present in Glyph Buffer $G_{(n)}$, copy the glyph from Glyph Buffer $G_{(n)}$ to the Presentation Buffer $P_{(n)}$ using the Glyph Copier; or
2) if an identical glyph is present in Glyph Buffer $G_{(n-1)}$, i.e. an identical glyph was present in Subtitle Event $E_{(n-1)}$, copy using the Glyph Copier the glyph from Glyph Buffer $G_{(n-1)}$ to both the Glyph Buffer $G_{(n)}$ and the Presentation Buffer $P_{(n)}$; or
3) Otherwise render using the Glyph Renderer the glyph into the Presentation Buffer $P_{(n)}$ and Glyph Buffer $G_{(n)}$ using the corresponding style information.

Two glyphs are identical if and only if the following TTML styles are identical:

- `tts:color`
- `tts:fontFamily`
- `tts:fontSize`
- `tts:fontStyle`
- `tts:fontWeight`
- `tts:textDecoration`
- `tts:textOutline`

Figure 67- provides an example of Presentation Compositor behavior.



- This subtitle is first.
- This is the ***second*** subtitle shown.

- Red with single underline: render
- Black with double underline: copy from current event
- Blue with dotted underline: copy from previous event

Figure 6-7 – Example of Text Subtitle Presentation Compositor Behavior

The Normalized Rendered Glyph Area of a given rendered glyph is defined as:

`Normalized Rendered Glyph Area ≡ (fontSize as percentage of root container height)`^{2}

Note: The Normalized Rendered Glyph Area calculation does not take into account glyph decorations (e.g. underline), glyph effects (e.g. outline) or actual glyph aspect ratio. A CFF-TT processor implementation can determine an actual buffer size needs based on worst-case glyph size complexity.

The Normalized Size of the Glyph Buffers $G_{(n)}$ or $G_{(n-1)}$ is defined as:

```
Normalized Glyph Buffer Size ≡
sum of the Normalized Rendered Glyph Area of the glyphs stored in the buffer within
a given time
```

Note: Setting a maximum Glyph Buffer Normalized Size effectively sets a limit on the total number of distinct glyphs present in any given Subtitle Event $E_{(n)}$. For example, assuming a maximum Normalized Glyph Buffer Size of 1 and the default `tts:fontSize` of 1c are used, the glyph's height as percentage of root container height is $\frac{1}{15}$, and the maximum number of distinct glyphs that can be buffered is $1 \div \left(\frac{1}{15}\right)^2 = 225$ glyphs. In this example, an implementation rendering at 1920x1080 would need to allocate a glyph buffer no smaller than$(1920 \div 32) \times (1080 \div 15) \times 225 = \sim 1$ Mpixels.

See Annex A, B, C for Glyph Buffer Normalized Size limits defined for each Profile.

The duration, in seconds, for painting a Subtitle Event in the Presentation Buffer is calculated as follows for any given Subtitle Event $E_{(n)}$:

$$\mathrm{DURATION}\left(\mathrm{E}_{(n)}\right) = \frac{\mathrm{S}_{(n)}}{\mathrm{BDraw}} + \mathrm{C}_{(n)}$$

For text-based CFF-TT subtitles, $C_{(n)}$ is calculated as follows for each Subtitle Event $E_{(n)}$:

$$\mathrm{C}_{(n)} = \sum_{i=0}^{i<cn} \left(\frac{1}{\mathrm{Ren}_i} \times \mathrm{NRGA}_i\right) + \sum_{j=0}^{j<cp} \left(\frac{1}{\mathrm{GCpy}} \times \mathrm{NRGA}_j\right)$$

Where:

- *cn* is the total number of glyphs rendered into the Presentation Buffer $P_{(n)}$ in Subtitle Event $E_{(n)}$.
- Ren_i is the text rendering performance factor of the ith rendered glyph in the Subtitle Event $E_{(n)}$.
- GCpy is the normalized glyph copy performance factor for glyphs that are copied to the Presentation Buffer $P_{(n)}$ using the Glyph Copier.
- NRGA_i and NRGA_j are the Normalized Rendered Glyph Area of the ith rendered glyph and jth copied glyph, respectively.
- *cp* is the total number of copied glyphs to the Presentation Buffer $P_{(n)}$ using the Glyph Copier in Subtitle Event $E_{(n)}$.

The $C_{(n)}$calculation SHALL include only glyphs in region elements presented in the Subtitle Event $E_{(n)}$ – see Section 6.2.2.3.2 for a definition of when a region is considered to be presented.

The $C_{(n)}$calculation SHALL NOT include a character (Unicode Code Point) if it does not result in a change to presentation (e.g. the Code Point is ignored by the CFF-TT Processor).

See Annex A, B, C for the definition of Ren and GCpy for each Profile.

Note: GCpy effectively sets a limit on animating glyphs. For example, assuming that the root container is ultimately rendered at 1920×1080 resolution and no regions need to have background color painted (so only a CLEAR($E_{(n)}$) operation is required for the normalized drawing area for the Subtitle Event), a GCpy and BDraw of 12 s^{-1} would mean that a group of 160 glyphs with a tts:fontSize equal to 5% of the root container height could be moved at most approximately $12s^{-1} \div \left(1 + (160 \times 0.05^2)\right) = 8.6$ times per second.

Note: Ren_i effectively sets a limit on the glyph rendering rate. For example, assuming that the root container is ultimately rendered at a 1920×1080 resolution, a Ren_i of 1.2 s^{-1} would mean that at most 120 glyphs with a fontSize of 108 px (10% of 1080 px and NGRA = 0.01) could be rendered every second.

6.6.4.2 Layout Model

Proper region size, glyph sizing and glyph layout will avoid clipping of text content.

The width of a region is indicated by the tts:extent <length> parameter associated with width. The width of rendered characters depends on the width of individual glyphs and the spacing between the glyphs (kerning). As this particular to each font, the layout model assumes particular reference font families as defined in Annex D.3; fonts that are 'metric compatible' can be substituted (a font is 'metric compatible' if it does not change text flow, although appearance may be different). Glyphs painted into the Presentation Buffer SHALL be laid out horizontally with the following parameters:

- Behavior is defined for two font families (tts:fontFamily):
  - "monospaceSerif"
  - "proportionalSansSerif"
- If text content has a "monospaceSerif" or "proportionalSansSerif" tts:fontFamily applied, width and spacing (kerning) is consistent with the Reference Font Family for the given tts:fontFamily.
- The Reference Font Family for a given subtitle language is defined in Annex D.3.
- Notes:
  - To guarantee that text content flowed into a region renders without clipping, it is necessary for the tts:extent <length> (width) parameter to be at least the sum each glyph on the line and its associated spacing (kerning). Insufficient <length> more typically results in vertical clipping as text flow moves some text to unanticipated additional lines.

The height of the region is specified by the tts:extent <length> parameter associated with height. As specified in [W3C-TT] and incorporated in [SMPTE-TT], the height of each line is a function of the size of largest font used within that line and the tts:lineHeight setting. Glyphs painted into the Presentation Buffer SHALL be laid out vertically within a region in accordance with the following parameters:

- A tts:lineHeight of "100%" corresponds with the size of the largest font on a line. For example, a line with characters of "10px", "12px" and "14px" and a tts:lineHeight of "100%" is equivalent to a tts:lineHeight of "14px".
- Inter-baseline separation is defined as follows:
  - The value of tts:lineHeight if tts:lineHeight is specified
  - tts:lineHeight="120%" if tts:lineHeight="normal" (note: if tts:lineHeight is not specified, "normal" is the default value applied).
  - actual line height is rounded up to whole pixels. For example, a line height of "12px" at "120%" results in 14.4 px and is rounded up to 15 px.

  - White space in excess of `tts:fontHeight` is allocated evenly, within a pixel, top and bottom (half-leading). Note that inter-baseline separation includes any leading.
- Notes:
  - To guarantee that content flowed into a region renders without vertical clipping, it is necessary for the `tts:extent` <length> (height) parameter to be at least the sum of the actual height of each line of text in the region. For example, three lines of 12px characters with a `tts:lineHeight=`"normal" would require a <length> of (12px * 120% *3) pixels.
  - Glyphs that do not extend beyond the specific `tts:fontSize` value will not clip top or bottom.

6.6.5 Constraints

The following constraints apply to the CFF-TT hypothetical render model.

Table 6-10 – Hypothetical Render Model Constraints

Property	Constraint
Document Buffer Size	500 x 2^{10} bytes minimum for one document
DOM Buffer Sizes	No specific limitations. The DOM buffer sizes are limited by the XML document size, but the size of the DOM buffer relative to document size depends on the specific implementation. It is up to the decoder implementation to ensure that sufficient memory is available for the 2 DOMs.

6.7 Data Structure for CFF-TT Track

6.7.1 Introduction

In this section, the operational rules for boxes and their contents of the Common File Format for CFF-TT subtitle tracks are described.

6.7.2 Track Header Box ('tkhd')

The Track Header Box ('`tkhd`') SHALL conform to the definition in Section 2.3.5, with the following additional or modified constraints:

- The following fields SHALL be set as defined:
  - `layer` = -1 (in front of video plane)
  - `flags` = 0x000007, indicating that `track_enabled`, `track_in_movie`, and `track_in_preview` are each 1
- The `width` and `height` SHALL be set in accordance with [ISOTEXT] with the following additional constraints:
  - Image subtitle tracks: the `width` and `height` SHALL be set to the width and height of the associated video; and
  - Text subtitle tracks which use scalar lengths in one or more CFF-TT documents: the `width` and `height` SHALL be set to the width and height of the associated video; and

  - Text subtitle tracks which do not use scalar lengths in any CFF-TT documents: the `width` and `height` SHOULD be set to the width and height of the associated video or SHOULD indicate an aspect ratio which matches the aspect ratio of the associated video track.
- Other template fields SHALL be set to their default values.

6.7.3 Movie Fragment Box (‘moof’)

Movie Fragments in subtitle tracks are required to conform to the following constrains:

- Every subtitle track Movie Fragment except the last Movie Fragment of a subtitle track SHALL have a duration of at least one second.
- The last Movie Fragment of a subtitle track MAY have a duration of less than one second.

6.7.4 Media Header Box (‘mdhd’)

The Media Header Box (‘mdhd’) SHALL conform to the definition in Section 2.3.6, with the following additional constraint:

- The `timescale` SHALL be set to the same value as the `timescale` of the associated video track's Media Header Box (‘mdhd’).

6.7.5 Handler Reference Box (‘hdlr’)

The syntax and values for the Handler Reference Box (‘hdlr’) for CFF-TT subtitle tracks SHALL conform to [ISO] with the following additional constraints:

- The `handler_type` field SHALL be set to “subt”
- The `name` field of the Handler Reference Box (‘hdlr’) for CFF-TT subtitle tracks SHOULD be set to the value of the "MetatadataMovie/TrackMetadata/Track/Subtitle/Type" assigned to the CFF-TT subtitle track in Multi-Track Required Metadata (see Section 2.1.2.1) when Multi-Track Required Metadata is present in the DCC. When there is more than one Type value defined for the subtitle track, the values SHALL be concatenated and comma-separated in the `name` field of the Handler Reference Box (‘hdlr’).

6.7.6 Subtitle Media Header Box (‘sthd’)

The syntax and values for the Subtitle Media Header Box (‘sthd’) SHALL conform to [ISO] Section 8.4.5.

6.7.7 Sample Description Box (‘stsd’)

For CFF-TT subtitle tracks, the Sample Description Box (‘stsd’) SHALL contain a `XMLSubtitleSampleEntry` that complies with [ISOTEXT] with the following additional constraints:

- The `namespace` field of `XMLSubtitleSampleEntry` SHALL list all of the XML namespaces declared in any of the CFF-TT track documents in the track with the following exceptions:
  - the built-in XML Schema namespaces “http://www.w3.org/2001/XMLSchema” and http://www.w3.org/2001/XMLSchema-instance SHALL NOT be listed.

  - ➢ any namespace declaration which has a prefix beginning with the three-letter sequence "xml" SHALL NOT be listed.
  - ➢ if a schema defines multiple namespaces, it SHOULD only be listed once (for example, only ...ns/ttml is recommended to be included, not both ...ns/ttml and ...ns/ttml#style).

  The namespace field SHALL NOT list any other namespaces. See also the requirements defined in Section 6.8.
- The schema_location field of XMLSubtitleSampleEntry SHALL include the XML schemas for all of the namespaces set in the namespace field. See also the requirements defined in Section 6.8.
- The auxiliary_mime_type field of XMLSubtitleSampleEntry SHALL be set to "image/png" if images are used in the CFF-TT subtitle track.

6.7.8 Sub-Sample Information Box ('subs')

The syntax and values for the Sub-Sample Information Box ('subs') SHALL conform to [ISOTEXT].

6.7.9 Track Fragment Run Box ('trun')

- One Track Fragment Run Box ('trun') SHALL be present in each subtitle track fragment.
- The data-offset-present, sample-size-present and sample-duration-present flags SHALL be set and corresponding values provided. Other flags SHALL NOT be set.

6.7.10Track Fragment Random Access Box ('tfra')

- One Track Fragment Random Access Box ('tfra') SHALL be stored in the Movie Fragment Random Access Box ('mfra') for each subtitle track if the Movie Fragment Random Access Box ('mfra') is present.
- The 'tfra' for a subtitle track SHALL list each of its subtitle track fragments as a randomly accessible sample.

6.8 Signaling for CFF-TT Tracks

6.8.1 Text Subtitle Tracks

A CFF-TT text subtitle track has the following characteristics:

- Each CFF-TT Document in the subtitle track complies with the restrictions defined in Section 6.2.2.4.
- The subtitle track does not contain any image files.

A CFF-TT text subtitle track SHALL be signaled as follows:

- The namepsace field of XMLSubtitleSampleEntry in the CFF-TT subtitle track Sample Description Box ('stsd') SHALL include "http://www.w3.org/ns/ttml".
- The schema_location field of XMLSubtitleSampleEntry in the CFF-TT subtitle track Sample Description Box ('stsd') SHALL include "http://www.w3.org/ns/ttml cff-tt-text-ttaf1-dfxp-{DMEDIA_VERSION_POINTS}.xsd".

- The “MetadataMovie/TrackMetadata/Track/Subtitle/Format” assigned to the CFF-TT subtitle track in Multi-Track Required Metadata (see Section 2.1.2.1) SHALL be set to “Text” if Multi-Track Required Metadata is present in the DCC.

Note: {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.

6.8.2 Image Subtitle Tracks

A CFF-TT image subtitle track has the following characteristics:

- Each CFF-TT Document in the subtitle track complies with the restrictions defined in Section 6.2.2.5.

A CFF-TT image subtitle track SHALL be signaled as follows:

- The namepsace field of XMLSubtitleSampleEntry in the CFF-TT subtitle track Sample Description Box (‘stsd’) SHALL include “http://www.w3.org/ns/ttml”.
- The schema_location field of XMLSubtitleSampleEntry in the CFF-TT subtitle track Sample Description Box (‘stsd’) SHALL include “http://www.w3.org/ns/ttml cff-tt-image-ttaf1-dfxp-{DMEDIA_VERSION_POINTS}.xsd”.
- The “MetadataMovie/TrackMetadata/Track/Subtitle/Format” assigned to the CFF-TT subtitle track in Multi-Track Required Metadata (see Section 2.1.2.1) SHALL be set to “Image” if Multi-Track Required Metadata is present in the DCC.

Note: {DMEDIA_VERSION_POINTS} is a parameter, defined in Annex E.

6.8.3 Combined Subtitle Tracks

The “MetadataMovie/TrackMetadata/Track/Subtitle/Format” assigned to the CFF-TT subtitle track in Multi-Track Required Metadata (see Section 2.1.2.1) SHALL NOT be set to “combined”.

Note: “combined” CFF-TT subtitle tracks are prohibited per Section 6.2.2.

6.9 Subtitle Language Considerations

6.9.1 Overview

CFF-TT subtitle tracks are associated with a “language”. This Section is intended to provide additional information regarding CFF-TT subtitle “languages”.

In this section, unless explicitly specified otherwise, the term “Primary Language Subtag” is as defined in [RFC5646] and specified Language Subtags are per those defined in [IANA-LANG].

6.9.2 Recommended Unicode Code Points per Subtitle Language

Table 6-11 defines the set of Unicode Code Points that SHOULD be used in text-based CFF-TT subtitle tracks that are associated with a “language” containing the specified “Primary Language Subtag”. Unicode Code Points are per those defined in [UNICODE].

Table 6-11 – Recommended Unicode Code Points per Language

Language (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
All	“x-ALL” (for the purposes of this specification,	(Basic Latin) U+0020 - U+007E

	this [RFC5646] private use subtag sequence is considered to represent all possible languages as defined in [IANA-LANG])	(Latin-1 Supplement) U+00A0 - U+00FF
		(Latin Extended-A) U+0152 : LATIN CAPITAL LIGATURE OE U+0153 : LATIN SMALL LIGATURE OE U+0160 : LATIN CAPITAL LETTER S WITH CARON U+0161 : LATIN SMALL LETTER S WITH CARON U+0178 : LATIN CAPITAL LETTER Y WITH DIAERESIS U+017D : LATIN CAPITAL LETTER Z WITH CARON U+017E : LATIN SMALL LETTER Z WITH CARON
		(Latin Extended-B) U+0192 : LATIN SMALL LETTER F WITH HOOK
		(Spacing Modifier Letters) U+02DC : SMALL TILDE
		(Combining Diacritical Marks) U+0301 : COMBINING ACUTE ACCENT
		(General Punctuation) U+2010 - U+2015 : Dashes U+2016 - U+2027 : General punctuation U+2030 - U+203A : General punctuation
		(Currency symbols) U+20AC : EURO SIGN
		(Letterlike Symbols) U+2103 : DEGREES CELSIUS U+2109 : DEGREES FAHRENHEIT U+2120: SERVICE MARK SIGN U+2122 : TRADE MARK SIGN
		(Number Forms) U+2153 – U+215F : Fractions
		(Mathematical Operators) U+2212: MINUS SIGN U+221E : INFINITY
		(Box Drawing) U+2500: BOX DRAWINGS LIGHT HORIZONTAL U+2502: BOX DRAWINGS LIGHT VERTICAL U+250C: BOX DRAWINGS LIGHT DOWN AND RIGHT U+2510: BOX DRAWINGS LIGHT DOWN AND LEFT U+2514: BOX DRAWINGS LIGHT UP AND RIGHT U+2518: BOX DRAWINGS LIGHT UP AND LEFT
		(Block Elements) U+2588: FULL BLOCK
		(Geometric Shapes) U+25A1: WHITE SQUARE
		(Musical Symbols) U+2669: QUARTER NOTE U+266A : EIGHTH NOTE U+266B: BEAMED EIGHTH NOTES
Albanian Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Albanian	“sq”	Same as defined for the “x-ALL” subtag sequence
Baltic Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Latvian, Lithuanian	“lv”, “lt”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” which is re-defined below
		(Latin Extended-A) U+0100 - U+017F
Finnic Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)

Finnish	"fi"	Same as defined for the “x-ALL” subtag sequence
Estonian	“et”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” which is re-defined below
		(Latin Extended-A) U+0100 - U+017F
Germanic Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Danish, Dutch/Flemish, English, German, Icelandic, Norwegian, Swedish	"da", “nl", “en”, "de", "is", "no", "sv"	Same as defined for the “x-ALL” subtag sequence
Greek Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Greek	“el”	Same as defined for the “x-ALL” subtag sequence
		(Combining Diacritical Marks) U+0308 : COMBINING DIAERESIS
		(Greek and Coptic) U+0386 : GREEK CAPITAL LETTER ALPHA WITH TONOS U+0387 : GREEK ANO TELEIAU+0388 – U+03CE : Letters
Romanic Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Catalan, French, Italian	“ca”, "fr", "it"	Same as defined for the “x-ALL” subtag sequence
Portuguese, Spanish	“pt”, “es”	(Currency symbols) U+20A1 : COLON SIGN U+20A2 : CRUZEIRO SIGN U+20B3 : AUSTRAL SIGN
Romanian	“ro”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” and “(Latin Extended-B)” which is re-defined below
		(Latin Extended-A) U+0100 - U+017F
		(Latin Extended-B) U+0192 : LATIN SMALL LETTER F WITH HOOK U+0218 : LATIN CAPITAL LETTER S WITH COMMA BELOW U+0219 : LATIN SMALL LETTER S WITH COMMA BELOW U+021A : LATIN CAPITAL LETTER T WITH COMMA BELOW U+021B : LATIN SMALL LETTER T WITH COMMA BELOW
Semitic Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Arabic	“ar”	Same as defined for the “x-ALL” subtag sequence
		U+0609: ARABIC-INDIC PER MILLE SIGN U+060C – U+060D : Punctuation U+061B : ARABIC SEMICOLON U+061E : ARABIC TRIPLE DOT PUNCTUATION MARK U+061F : ARABIC QUESTION MARK U+0621 – U+063A : Based on ISO 8859-6 U+0640 – U+064A : Based on ISO 8859-6 U+064B – U+0652 : Points from ISO 5559-6 U+0660 – U+0669 : Arabic-Indic digits U+0670: ARABIC LETTER SUPERSCRIPT ALEF U+066A – U+066D : Punctuation
Hebrew	“he”	Same as defined for the “x-ALL” subtag sequence

		(Hebrew) U+05B0 – U+05C3 : Points and punctuation U+05D0 – U+05EA : Based on ISO 8859-8 U+05F3 – U+05F4 : Additional punctuation
Slavic Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Croatian, Czech, Polish, Slovenian, Slovak	“hr”, “cs”, “pl”, “sl”, “sk”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” which is re-defined below
		(Latin Extended-A) U+0100 - U+017F
Bosnian, Bulgarian, Macedonian, Russian, Serbian, Ukrainian	“bs”, “bg”, “mk”, “ru”, “sr”, “uk”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” which is re-defined below
		(Latin Extended-A) U+0100 - U+017F
		(Spacing Modifier Letters) U+02BC : MODIFIER LETTER APOSTROPHE
		(Cyrillic) U+0400 – U+040F : Cyrillic extensions U+0410 – U+044F : Basic Russian alphabet U+0450 – U+045F : Cyrillic extensions U+048A – U+04F9: Extended Cyrillic
		(Letterlike Symbols) 6.9.2.1 U+2116 : NUMERO SIGN
Turkic Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Turkish	“tr”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” which is re-defined
		(Latin Extended-A) U+0100 - U+017F
Kazakh	“kk”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” which is re-defined
		(Latin Extended-A) U+0100 - U+017F
		(Cyrillic) U+0400 – U+040F : Cyrillic extensions U+0410 – U+044F : Basic Russian alphabet U+0450 – U+045F : Cyrillic extensions U+048A – U+04F9: Extended Cyrillic
Ugric Languages (Informative)	Primary Lang Subtags (Normative)	Unicode Code Points (Normative)
Hungarian	“hu”	Same as defined for the “x-ALL” subtag sequence, except for “(Latin Extended-A)” which is re-defined below
		(Latin Extended-A) U+0100 - U+017F
		(General Punctuation) U+2052: COMMERCIAL MINUS SIGN
		(Miscellaneous Mathematical Symbols-A) U+27E8: MATHEMATICAL LEFT ANGLE BRACKET U+27E9: MATHEMATICAL RIGHT ANGLE BRACKET

Note: it is expected that additional Language Subtags and associated Unicode Code Points will be added to a future release of this specification.

6.9.3 Reference Font Family per Subtitle Language

tts:fontFamily	Primary Lang Subtags	Reference Font Family
monospaceSerif	All languages defined in D.2 above.	Courier New: http://www.microsoft.com/typography/fonts/family.aspx?FID=10
proportionalSansSerif	All languages defined in D.2 above with the exception of Semitic Languages.	Arial: http://www.microsoft.com/typography/fonts/family.aspx?FID=8 Helvetica: http://www.linotype.com/en/526/Helvetica-family.html

Notes:

- the Reference Font Families were chosen because of their common use, general availability and the availability of license-free metric equivalent font families.
- Per Section 6.6.4.2, fonts that are ‘metric compatible’ can be substituted for Reference Font Family (a font is ‘metric compatible’ if it does not change text flow, although appearance may be different).
- It is expected that additional Reference Font Family definitions for other Languages will be added to a future release of this specification.

6.9.4 Typical Subtitle Practice per Region (Informative)

Table 6-12 below provides an informative summary of subtitle languages commonly used in the listed regions. Primary language and Primary Language Subtag are indicated, with additional common region or script variant Language Subtags in brackets.
Note that Table 6-11 provides Unicode Code Points associated with “Primary Language Subtag”.

Table 6-12 – Subtitles per Region

Region	Subtitle Languages (Lang Subtags)
ALL (worldwide)	English (“en”)
America (North)	
ALL	French (“fr”) [Québécois (“fr-CA”) or Parisian (“fr-FR”)]
United States	Spanish (“es”) [Latin American (“es-419”)]
America (Central and South)	
ALL	Spanish (“es”) [Latin American (“es-419”)]
Brazil	Portuguese (“pt”) [Brazilian (“pt-BR”)]
Asia, Middle East, and Africa	
China	Chinese (“zh”) [Simplified Mandarin (“zh-cmn-Hans”)]
Egypt	Arabic (“ar”)
Hong Kong	Chinese (“zh”) [Cantonese (“zh-yue”)]
India	Hindi (“hi”) Tamil (“ta”) Telugu (“te”)
Indonesia	Indonesian (“id”)
Israel	Hebrew (“he”)
Japan	Japanese (“ja”)

Region	Subtitle Languages (Lang Subtags)
Kazakhstan	Kazakh (“kk”)
Malaysia	Standard Malay (“zsm”)
South Korea	Korean (“ko”)
Taiwan	Chinese (“zh”) [Traditional Mandarin (“zh-cmn-Hant”)]
Thailand	Thai (“th”)
Vietnam	Vietnamese (“vi”)
Europe	
Benelux (Belgium, Netherlands, and Luxembourg)	French (“fr”) [Parisian (“fr-FR”)] Dutch/Flemish (“nl”)
Denmark	Danish (“da”)
Finland	Finnish (“fi”)
France	French (“fr”) [Parisian (“fr-FR”)] Arabic (“ar”)
Germany	German (“de”) Turkish (“tr”)
Italy	Italian (“it”)
Norway	Norwegian (“no”)
Spain	Spanish (“sp”) [Castilian (“sp-ES”)] Catalan (“ca”)
Sweden	Swedish (“sv”)
Switzerland	French (“fr”) [“fr-CH” or “fr-FR”] German (“de”) [“de-CH”] Italian (“it”) [“it-CH”]
Albania	Albanian (“sq”)
Bulgaria	Bulgarian (“bg”)
Croatia	Croatian (“hr”)
Czech Republic	Czech (“cs”)
Estonia	Estonian (“et”)
Greece	Greek (“el”)
Hungary	Hungarian (“hu”)
Iceland	Icelandic (“is”)
Latvia	Latvian (“lv”)
Lithuania	Lithuanian (“lt”)
Macedonia	Macedonian (“mk”)
Poland	Polish (“pl”)
Portugal	Portuguese (“pt”) [Iberian (“pt-PT”)]

Region	Subtitle Languages (Lang Subtags)
Romania	Romanian (“ro”)
Russia	Russian (“ru”)
Serbia	Serbian (“sr”)
Slovakia	Slovak (“sk”)
Slovenia	Slovenian (“sl”)
Turkey	Turkish (“tr”)
Ukraine	Ukrainian (“uk”)

Note: it is expected that additional Language Subtags will be added to future releases of this specification.

6.10 Closed Caption Subtitles Transcoded from CEA 608 or CEA 708

Subtitle tracks authored for the deaf and hard of hearing (closed captioning) are signaled in the required metadata with the type "SDH". All subtitle tracks are required to conform to a CFF-TT Profile which, as noted in Section 6.2.2, are derived from the SMPTE TT Profile defined in [SMPTE-TT].

When SDH tracks are transcoded from CEA 608 [CEA608] bitstreams the tracks SHALL conform with the metadata provisions defined in SMPTE RP2052-10 [SMPTE-608], Section 5.3, specifically that m608:channel SHALL be present and all other metadata present in the original CEA 608 bitstream SHOULD be included. Tracks SHOULD comply with the entirety of [SMPTE-608].

When SDH tracks are transcoded from CEA 708 [CEA708] bitstreams the tracks SHALL conform with the metadata provisions defined in SMPTE RP2052-11 [SMPTE-708], Section 5.4. Tracks SHOULD comply with the entirety of [SMPTE-708].

When tunnel data is provided, it SHALL comply with the tunnel encoding provisions of [SMPTE-TT], [SMPTE-608] and [SMPTE-708] as appropriate.

Annex A. CFF Parameters

Table A- 1 – Current Version

Parameter	Value	Description
DMEDIA_VERSION_NOPOINTS	200	This version of the DMedia specification, without point notation.
DMEDIA_VERSION_POINTS	2.0.0	This version of the DMedia specification, with point notation.

Table A- 2 – Compatible Version

DMedia Compatible Version	Description
2.0.0	Common File Format & Media Formats Specification V 2.0

Annex B. Media Profiles

B.1. PD Media Profile

B.1.1. Overview

The PD Media Profile defines an audio-visual content interoperability point for portable devices. The PD Media Profile is identified by the `'cfpd'` four character code registered with [MP4RA].

B.1.2. Constraints on Encryption

Content conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 3, Encryption of Track Level Data, with the additional constraints defined here.

- Encrypted tracks SHALL restrict the value of default_IV_size in `'tenc'` to 0x08, and the value of IV_size in `'seig'` (when sample groups are present) to 0x00 or 0x08; and
- Each encrypted audio track SHALL be encrypted using a single key; and
- Each encrypted video track SHALL be encrypted using a single key; and
- Subtitle tracks SHALL NOT be encrypted.

Note: Encryption is not mandatory.

B.1.3. Constraints on Video

Video tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 4, Video Elementary Streams, with the additional constraints defined here.

- The video track SHALL be an AVC Video track as defined in Section 4.3.

B.1.3.1. Constraints on [H264] Elementary Streams

B.1.3.1.1. Profile and Level

[H264] elementary streams conforming to this Media Profile SHALL comply with the following [H264] Profile and Level constraints:

- Constrained Baseline Profile as defined in [H264].
- Up to Level 1.3 as defined in [H264].

B.1.3.1.2. Maximum Bitrate

[H264] elementary streams conforming to this Media Profile SHALL NOT exceed the maximum bitrate defined by [H264] for the chosen [H264] Profile and Level (note: Constrained Baseline Profile at Level 1.3 is the worse case with a 768x10^3 bits/sec maximum bitrate). See Section 4.3.2.4 for more information on the calculation of [H264] elementary stream maximum bitrate.

B.1.3.2. Constraints on Mastering

NAL Structured Video streams conforming to this Media Profile SHALL be encoded using the color parameters defined by [R709]. Presentation of content is assumed to occur on a display which uses the electro-optic transfer function specified in [R1886] with a peak luminance of 100 cd/m2. It is recommended that content be graded with a reference viewing environment that complies with [R2035].

B.1.3.3. Constraints on Encoding

NAL Structured Video streams conforming to this Media Profile:

- SHALL have the following pre-determined values:
    - `video_full_range_flag`, if present, SHALL be set to 0.

B.1.3.4. Constraints on Picture Formats

NAL Structured Video streams conforming to this Media Profile SHALL NOT exceed the following coded picture format constraints:

- Maximum encoded vertical sample count of 240 samples.
- Maximum frame rate of 30000÷1000 (frame rate is calculated as per Section 4.3.2.5 and Section 4.4.2.6).

B.1.4. Constraints on Audio

Audio tracks conforming to this Media Profile SHALL comply with:

- all of the requirements and constraints defined in Section 5, Audio Elementary Streams,.
- one of the allowed combinations of audio format, maximum number of channels, maximum elementary stream bitrate, and sample rate defined below.

Table B - 1 – Allowed Audio Formats in PD Media Profile

Audio Format	Max. No. Channels	Sample Rate	Max. Bitrate	Bitrate Calculation
MPEG-4 AAC [2-Channel]	2	48 kHz	192 Kbps	Section 5.3.2.2.2.4
MPEG-4 HE AAC v2	2	48 kHz	192 Kbps	Section 5.3.4.2.2.3
MPEG-4 HE AAC v2 with MPEG Surround	5.1	48 kHz	192 Kbps	Section 5.3.5.2.2.3

B.1.5. Constraints on Subtitles

Subtitle tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 6, Subtitle Elementary Streams, with the following additional constraints:

- A CFF-TT subtitle track SHALL NOT exceed the constraints defined below.

Table B - 2 – Hypothetical Render Model Constraints (General)

Property	Constraint
Subtitle fragment/sample size	Total sample size <= 500 x 2^{10} bytes
Normalized background drawing performance factor (BDraw)	12 (Performance Factor (1/s)

- A CFF-TT text subtitle track SHALL NOT exceed the constraints defined below.

Table B - 3 – Hypothetical Render Model Constraints (Text subtitle)

Property	Constraint
Maximum Normalized Glyph Buffer Size	1 (Buffer Size)
Normalized glyph copy performance factor (GCpy)	12 (Performance Factor (1/s)
Non-CJK text rendering performance factor (Ren)	1.2 (Performance Factor (1/s)
CJK text rendering performance factor (Ren)	0.6 (Performance Factor (1/s)

Where:

- CJK = Chinese, Japanese, Korean Glyphs.
- The above table defines performance applying to all supported font styles (including provision of outline border).

- CFF-TT image subtitle tracks SHALL NOT be used.

B.2. SD Media Profile

B.2.1. Overview

The SD Media Profile defines an audio-visual content interoperability point for standard definition devices. The SD Media Profile is identified by the `'cfsd'` four character code registered with [MP4RA].

B.2.2. Constraints on Encryption

Content conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 3, Encryption of Track Level Data, with the additional constraints defined here.

- Encrypted tracks SHALL comply with the constraints defined in Annex B.1.2.
  Note: Encryption is not mandatory.

B.2.3. Constraints on Video

Video tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 4, Video Elementary Streams, with the additional constraints defined here.

B.2.3.1. Constraints on [H264] Elementary Streams

B.2.3.1.1. Profile and Level

[H264] elementary streams conforming to this Media Profile SHALL comply with the following [H264]Profile and Level constraints:

- Constrained Baseline Profile as defined in [H264].
- Up to Level 3 as defined in [H264].

B.2.3.1.2. Maximum Bitrate

H264] elementary streams conforming to this Media Profile SHALL NOT exceed the maximum bitrate defined by [H264] for the chosen [H264] Profile and Level (note: Constrained Baseline Profile at Level 3 is the worse case with a be $10x10^6$ bits/sec maximum bitrate). See Section 4.3.2.4 for more information on the calculation of [H264] elementary stream maximum bitrate.

B.2.3.2. Constraints on [H265] Elementary Streams

B.2.3.2.1. Profile, Tier and Level

[H265] elementary streams conforming to this Media Profile SHALL comply with the following [H265] Profile, Tier and Level constraints:

- Main Profile as defined in [H265].
- Main Tier as defined in [H265].
- Up to Level 3.1 as defined in [H265].

B.2.3.2.2. Maximum Bitrate

[H265] elementary streams conforming to this Media Profile SHALL NOT exceed the maximum bitrate defined by [H265[for the chosen [H265] Profile, Tier and Level (note: Main Profile, Main Tier and Level 3.1 is the worse case with a 10x10^6 bits/sec maximum bitrate). See Section 4.4.2.5 for more information on the calculation of [H265] elementary stream maximum bitrate.

B.2.3.3. Constraints on Mastering

NAL Structured Video streams conforming to this Media Profile:

- SHOULD be encoded using the color parameters defined by [R709]; and
- SHALL be encoded with the following color parameters:
  - for 24 Hz, 30 Hz & 60 Hz content: the color parameters defined by [R709], or the color parameters defined for 525-line video systems as per [R601]; or
  - for 25 Hz & 50 Hz content: the color parameters defined by [R709], or the color parameters defined for 625-PAL video systems as per [R1700].
- Presentation of content is assumed to occur on a display which uses the electro-optic transfer function specified in [R1886] with a peak luminance of 100 cd/m2. It is recommended that content be graded with a reference viewing environment that complies with [R2035].

B.2.3.4. Constraints on Encoding

NAL Structured Video streams conforming to this Media Profile:

- SHALL have the following pre-determined values:
  - `video_full_range_flag`, if present, SHALL be set to 0.
  - `colour_description_present_flag` SHALL be set to 1 if the color parameters from [R709] are not used.

B.2.3.5. Constraints on Picture Formats

NAL Structured Video streams conforming to this Media Profile SHALL NOT exceed the following coded picture format constraints:

- Maximum encoded vertical sample count of 480 samples.
- Maximum frame rate of 60000÷1000 (frame rate is calculated as per Section 4.3.2.5 and Section 4.4.2.6).

B.2.4. Constraints on Audio

Audio tracks conforming to this Media Profile SHALL comply with:

- all of the requirements and constraints defined in Section 5, Audio Elementary Streams,.
- one of the allowed combinations of audio format, maximum number of channels, maximum elementary stream bitrate, and sample rate defined in either of the tables below.

Table B - 4 – Allowed Audio Formats in SD Media Profile

Audio Format	Max. No. Channels	Sample Rate	Max. Bitrate	Bitrate Calculation
MPEG-4 AAC [2-Channel]	2	48 kHz	192 Kbps	Section 5.3.2.2.2.4
MPEG-4 HE AAC V2 level 4[5.1-channel]	5.1	48 kHz	1440 Kbps	Section 5.3.3.2.2.5
AC-3 (Dolby Digital)	5.1	48 kHz	640 Kbps	Section 5.5.1.2.3
Enhanced AC-3 (Dolby Digital Plus)	5.1	48 kHz	3024 Kbps	Section 5.5.2.2.5
DTS	5.1	48 kHz	1536 Kbps	Section 5.6.2.2
DTS-HD	5.1	48 kHz	3018 Kbps	Section 5.6.2.2

Table B - 5 – Allowed Enhanced Audio Formats in SD Media Profile

Audio Format	Max. No. Channels	Sample Rate	Max. Bitrate	Bitrate Calculation
MPEG-4 HE AAC V2 level 6 [5.1, 7.1-Channel]	7.1	48 kHz	2016 Kbps	Section 5.3.3.2.2.5
Enhanced AC-3 (Dolby Digital Plus)	7.1	48 kHz	3024 Kbps	Section 5.5.2.2.5
DTS	6.1	48 kHz	1536 Kbps	Section 5.6.2.2
	5.1	48 kHz, 96 kHz	1536 Kbps	Section 5.6.2.2
DTS-HD	7.1	48 kHz, 96 kHz	6123 Kbps	Section 5.6.2.2
DTS-HD Master Audio	8	48 kHz, 96 kHz, 192 kHz	24.5 Mbps	Section 5.6.2.2
MLP (Dolby TrueHD)	8	48 kHz, 96 kHz, 192 kHz	18 Mbps	Section 5.5.3.2.4

B.2.5. Constraints on Subtitles

Subtitle tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 6, Subtitle Elementary Streams, with the following additional constraints:

- A CFF-TT subtitle track SHALL NOT exceed the sample constraints defined in Annex B.1.5.
- A CFF-TT text subtitle track SHALL NOT exceed the constraints defined in Annex B.1.5.
- A CFF-TT image subtitle track SHALL NOT exceed the constraints defined below.

Table B - 6 – Hypothetical Render Model Constraints (Image subtitle)

Property	Constraint
Reference image size	Single image size <= 100 x 2^{10} bytes
Encoded Image Buffer Size	500 x 2^{10} bytes. Note: Sample size is limited to 500 x 2^{10} bytes, but a CFF-TT document can be arbitrarily small, so nearly the entire subtitle sample could be filled with image data.
Decoded Image Buffer size	2 x 2^{20} pixels for each of the two Decoded Image Buffers. A Decoded Image Buffer can buffer all de-compressed images from a subtitle sample.
Image Decoding rate	1 x 2^{20} pixels per second
Normalized image copy performance factor (ICpy)	6 (Performance Factor (1/s)

B.3. HD Media Profile

B.3.1. Overview

The HD Media Profile defines an audio-visual content interoperability point for high definition devices. The HD Media Profile is identified by the `'cfhd'` four character code registered with [MP4RA].

B.3.2. Constraints on Encryption

Content conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 3, Encryption of Track Level Data, with the additional constraints defined here.

- Encrypted tracks SHALL comply with the constraints defined in Annex B.1.2.
  Note: Encryption is not mandatory.

Note: Encryption is not mandatory.

B.3.3. Constraints on Video

Video tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 4, Video Elementary Streams, with the additional constraints defined here.

B.3.3.1. Constraints on [H264] Elementary Streams

B.3.3.1.1. Profile and Level
[H264] elementary streams conforming to this Media Profile SHALL comply with the following [H264] Profile and Level constraints:

- Constrained Baseline Profile, Main Profile or High Profile as defined in [H264].
- Up to Level 4 as defined in [H264].

B.3.3.1.2. Maximum Bitrate
[H264] elementary streams conforming to this Media Profile SHALL NOT exceed the maximum defined by [H264] for the chosen [H264] Profile and Level (note: High Profile at Level 4 is the worse case with a 25.0×10^6 bits/sec maximum bitrate). See Section 4.3.2.4 for more information on the calculation of [H264] elementary stream maximum bitrate.

B.3.3.2. Constraints on [H265] Elementary Streams

B.3.3.2.1. Profile, Tier and Level
[H265] elementary streams conforming to this Media Profile SHALL comply with the following [H265] Profile, Tier and Level constraints:

- Main Profile or Main 10 Profile as defined in [H265].
- Main Tier as defined in [H265].
- Up to Level 4.1 as defined in [H265].

B.3.3.2.2. Maximum Bitrate

[H265] elementary streams conforming to this Profile SHALL NOT exceed the maximum defined by [H265] for the chosen [H265] Profile, Tier and Level (note: Main 10 Profile, Main Tier and Level 4.1 is the worse case with a $20.0x10^6$ bits/sec maximum bitrate). See Section 4.4.2.5 for more information on the calculation of [H265] elementary stream maximum bitrate.

B.3.3.3. Constraints on Mastering

NAL Structured Video streams conforming to this Media Profile SHALL be encoded using the color parameters defined by [R709]. Presentation of content is assumed to occur on a display which uses the electro-optic transfer function specified in [R1886] with a peak luminance of 100 cd/m2. It is recommended that content be graded with a reference viewing environment that complies with [R2035].

B.3.3.4. Constraints on Encoding

NAL Structured Video streams conforming to this Media Profile:

- SHALL have the following pre-determined values:
    - `video_full_range_flag`, if present, SHALL be set to 0.

B.3.3.5. Constraints on Picture Formats

NAL Structured Video streams conforming to this Media Profile SHALL NOT exceed the following coded picture format constraints:

- Maximum encoded vertical sample count of 1080 samples.
- Maximum frame rate of 60000÷1000 (frame rate is calculated as per Section 4.3.2.5 and Section 4.4.2.6).

B.3.4. Constraints on Audio

Audio tracks conforming to this Media Profile SHALL comply with:

- all of the requirements and constraints defined in Section 5, Audio Elementary Streams,.
- one of the allowed combinations of audio format, maximum number of channels, maximum elementary stream bitrate, and sample rate defined in Annex B.2.4.

B.3.5. Constraints on Subtitles

Subtitle tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Annex B.2.5.

B.4. xHD Media Profile

B.4.1. Overview

The xHD Media Profile defines an audio-visual content interoperability point for high definition devices supporting high bitrate.
The xHD Media Profile is identified by the 'cfxd' code registered with [MP4RA].

B.4.2. Constraints on Encryption

Content conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 3, Encryption of Track Level Data, with the additional constraints defined here.

- Each encrypted audio track SHALL be encrypted using a single key; and
- Each encrypted video track MAY be encrypted using more than one key; and
- Subtitle tracks SHALL NOT be encrypted; and
- For a given KID, Initialization Vectors SHALL NOT be re-used and SHALL follow the guidelines outlined in [CENC] Section 9.2 and 9.3. This uniqueness constraint applies to all Initialization Vectors that are explicitly set in the file and those that are implicitly generated by AES-CTR.

Note: Encryption is not mandatory.

B.4.3. Constraints on Video

Video tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Section 4, Video Elementary Streams, with the additional constraints defined here.

B.4.3.1. Constraints on [H264] Elementary Streams

B.4.3.1.1. Profile and Level

[H264] elementary streams conforming to this Media Profile SHALL comply with the following [H264] Profile and Level constraints:

- Constrained Baseline Profile, Main Profile or High Profile as defined in [H264].
- Up to Level 4.1 as defined in [H264].

B.4.3.1.2. Maximum Bitrate

[H264] elementary streams conforming to this Media Profile SHALL NOT exceed a maximum input bitrate of 40.0x10^6 bits/sec. See Section 4.3.2.4 for more information on the calculation of [H264] elementary stream maximum bitrate.

B.4.3.2. Constraints on [H265] Elementary Streams

B.4.3.2.1. Profile, Tier and Level

[H265] elementary streams conforming to this Media Profile SHALL comply with the following [H265] Profile, Tier and Level constraints:

- Main Profile or Main 10 Profile as defined in [H265].
- Main Tier or High Tier as defined in [H265].
- Up to Level 4.1 as defined in [H265].

B.4.3.2.2. Maximum Bitrate

[H265] elementary streams conforming to this Profile SHALL NOT exceed a maximum input bitrate of 40.0x10^6 bits/sec maximum bitrate). See Section 4.4.2.5 for more information on the calculation of [H265] elementary stream maximum bitrate.

B.4.3.3. Constraints on Mastering

NAL Structured Video streams conforming to this Media Profile SHALL be encoded using the color parameters defined by [R709]. Presentation of content is assumed to occur on a display which uses the electro-optic transfer function specified in [R1886] with a peak luminance of 100 cd/m2. It is recommended that content be graded with a reference viewing environment that complies with [R2035].

B.4.3.4. Constraints on Encoding

NAL Structured Video streams conforming to this Media Profile:

- SHALL have the following pre-determined values:
  - ➢ `video_full_range_flag`, if present, SHALL be set to 0.

B.4.3.5. Constraints on Picture Formats

NAL Structured Video streams conforming to this Media Profile SHALL NOT exceed the following coded picture format constraints:

- Maximum encoded vertical sample count of 1080 samples.
- Maximum frame rate of 60000÷1000 (frame rate is calculated as per Section 4.3.2.5 and Section 4.4.2.6).

B.4.4. Constraints on Audio

Audio tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Annex B.3.4.

B.4.5. Constraints on Subtitles

Subtitle tracks conforming to this Media Profile SHALL comply with all of the requirements and constraints defined in Annex B.3.5.

Annex C. Delivery Targets

C.1. General Download Delivery Target Constraints

C.1.1. Constraints on File Structure

DCCs conforming to a Download Delivery Target SHALL comply with all of the requirements and constraints defined in Section 2, The Common File Format, with the additional constraints defined as follows:

- The DCC Movie Fragment SHALL contain one Media Data Box (`‘mdat’`).

C.1.2. Constraints on Video

DCCs conforming to a Download Delivery Target SHALL comply with all of the requirements and constraints defined in Section 4, Video Elementary Streams, with the additional constraints defined as follows:

- For each Video track the following requirements apply:
  - ➢ Video track Movie Fragments SHALL start with SAP type 1 or 2.
  - ➢ Video tracks utilizing (`‘avc1’`) sample entries as defined in Section 4.3.1.1: every AVC video track fragment SHALL contain a Trick Play Box (`‘trik’`).
  - ➢ Video tracks which have one or more Fragments with a duration greater than 3.003 secs and utilize (`‘avc3’`) sample entries as defined in Section 4.3.1.1 SHALL be signaled in the Track Fragment Box (`'traf'`) using Random Access Point (RAP) sample grouping as defined in [ISO] Section 10.4, with the following additional constraints:
    - ➢ All sync samples SHALL be signaled in a Sample to Group Box (`‘sbgp’`) with the following constraints:
      - o `grouping_type`: set to `'rap '`
      - o `group_description_index` for the sync sample: corresponds to a `VisualRandomAccessEntry` in the Sample Group Description Box (`‘sgpd’`) with the following field values
        - o `num_leading_samples_known=1`
        - o `num_leading_samples=0`
    - ➢ All AVC RA-I samples (defined in Section 4.2.6) SHALL be signaled in a Sample to Group Box (`‘sbgp’`) with the following constraints:
      - o `grouping_type`: set to `'rap '`
      - o `group_description_index` for the RAI sample: corresponds to a `VisualRandomAccessEntry` in the Sample Group Description Box (`‘sgpd’`) with the following field values
        - o `num_leading_samples_known=1`
        - o `num_leading_samples`=number of “leading samples”
        - o Note that a “leading sample” in RAP Sample Grouping is defined as a sample that precedes the RAP in presentation order and

immediately follows the RAP or another leading sample in decoding order, and cannot be correctly decoded when decoding starts from the RAP.

- Video tracks which utilize (`'hev1'`) sample entries as defined in Section 4.4.1.1 SHALL signal Type of sync samples using sync sample sample grouping as defined in [ISOVIDEO] Section 8.4.4.

C.2. Multi-Track Download Delivery Target

The Multi-Track Download Delivery Target is intended for applications where a standalone DCC is made available for download delivery.
The Multi-Track Download Delivery Target is defined by the `'cfd1'` brand, which is a code point on the ISO Base Media File Format defined by [ISO].

C.2.1. Constraints on File Structure

DCCs conforming to this Delivery Target SHALL comply with all of the requirements and constraints defined in Annex C.1.1 with the additional constraints defined as follows:

- The File Type Box (`'ftyp'`) SHALL list the `'cfd1'` brand as a `compatible_brand`.
- The Metadata Box (`'meta'`) contained in the Movie Box (`'moov'`) for Required Multi-Track Metadata as defined in Section 2.1.2.1 SHALL be present in the DCC.
- A Free Space Box (`'free'`) SHALL be the last box in the Movie Box (`'moov'`). Note: this provides reserved space for adding DRM-specific information.
- The DCC Footer SHALL contain a Movie Fragment Random Access Box (`'mfra'`).

C.2.2. Constraints on Encryption

C.2.2.1. Constraints on PD and SD Media Profile Encryption

DCCs conforming to this Delivery Target and the PD Media Profile SHALL comply with all of the requirements and constraints defined in Annex B.1.2 with the additional constraints defined in this Section.
DCCs conforming to this Delivery Target and SD Media Profile SHALL comply with all of the requirements and constraints defined in Annex B.2.2
with the additional constraints defined in this Section.

- All encrypted audio tracks in the DCC SHALL be encrypted using the same key ("audio key").
- All encrypted video tracks in the DCC SHALL be encrypted using the same key ("video key").
- The video key and audio key SHALL be the same key.

C.2.2.2. Constraints on HD Media Profile Encryption

DCCs conforming to this Delivery Target and the HD Media Profile SHALL comply with all of the requirements and constraints defined in Annex B.3.2 with the additional constraints defined in this Section.

- All encrypted audio tracks in the DCC SHALL be encrypted using the same key ("audio key").
- All encrypted video tracks in the DCC SHALL be encrypted using the same key ("video key").

- The video key SHOULD be separate (independently chosen) from the audio key.

C.2.2.3. Constraints on xHD Media Profile Encryption

DCCs conforming to this Delivery Target and the xHD Media Profile SHALL comply with all of the requirements and constraints defined in Annex B.4.2 with the additional constraints defined in this Section.

- All encrypted audio tracks in the DCC SHALL be encrypted using the same key ("audio key").
- video keys SHOULD be separate (independently chosen) from the audio key.

C.2.3. Constraints on Video

DCCs conforming to this Delivery Target shall comply with all of the requirements and constraints defined in Annex C.1.2 with the additional constraints defined as follows:

- DCCs SHALL contain exactly one video track, and that video track SHALL conform to Section 4.
- A video track fragment SHALL have a duration no greater than 3.003 seconds.

C.2.3.1. Constraints on Sequence Parameter Sets (SPS)

The condition of the following [H264] fields SHALL NOT change throughout an AVC Video track conforming to this Delivery Target:

- `pic_width_in_mbs_minus1`
- `pic_height_in_map_units_minus1`

The condition of the following [H265] fields SHALL NOT change throughout an HEVC Video track conforming to this Delivery Target:

- `pic_width_in_luma_samples`
- `pic_height_in_luma_samples`

C.2.3.1.1. Constraints on Visual Usability Information (VUI) Parameters

The condition of the following SHALL NOT change throughout AVC Video tracks conforming to this Delivery Target:

- `aspect_ratio_idc`
- Bitrate[] (calculated by `bit_rate_scale` and `bit_rate_value_minus1`)
- CpbSize[] (calculated by `cpb_size_scale` and `cpb_size_value_minus1`)

The condition of the following SHALL NOT change throughout HEVC Video tracks conforming to this Delivery Target:

- `aspect_ratio_idc`
- `aspect_ratio_idc`
- `cpb_cnt_minus1`
- `bit_rate_scale`
- `bit_rate_value_minus1`
- `cpb_size_scale`
- `cpb_size_value_minus1`

C.2.3.2. Constraints on Picture Formats

The Media Profile definitions in Annex B define permitted picture formats. This Section provides more detail on permitted picture formats per Media Profile for Download Delivery Targets. Constraints are defined in the form of frame size and frame rate.

- *Frame size* is defined as the maximum display width and height of the picture in square pixels after cropping and subsample rescaling is applied. For each picture format defined, one or more allowed value combinations are specified for horizontal and vertical sub-sample factors, which are necessary for selecting valid Track Header Box `width` and `height` properties, as specified in Section 2.3.5 and Section 4.2.1.
- When sub-sampling is applied, at least one of either the width or the height of the encoded picture size SHALL match the value specified in the "Max Size Encoded" column in the following Tables. See Section 4.5 for more information.
- *Frame rate* is defined as a ratio corresponding to a real number. This number SHALL precisely (with no rounding error permitted) match the value calculated per Section 4.3.2.5 and Section 4.4.2.6).
- `aspect_ratio_idc` SHALL be encoded as listed or set to 255 and `Extended_SAR` provided to match the aspect ratio defined.

C.2.3.2.1. Constraints on PD Media Profile Picture Formats

NAL Structured Video streams conforming to the PD Media Profile (see Annex B.1) and this Delivery Target SHALL comply with the picture format constraints listed below.

Table C - 1 – Picture Formats and Constraints of PD Media Profile for 24 Hz & 30 Hz Content

Picture Formats			Sub-sample Factors			Encoding Parameters
Frame size (*width* x *height*)	Picture aspect	Frame rate	Horiz.	Vert.	Max. size encoded	aspect_ratio_idc
320 x 180	1.778	24000÷1000, 24000÷1001, 30000÷1000, 30000÷1001	1	1	320 x 180	1
320 x 240	1.333	24000÷1000, 24000÷1001, 30000÷1000, 30000÷1001	1	1	320 x 240	1
416 x 240 (Note)	1.733	24000÷1000, 24000÷1001, 30000÷1000, 30000÷1001	1	1	416 x 240	1

Table C - 2 – Picture Formats and Constraints of PD Media Profile for 25 Hz Content

Picture Formats			Sub-sample Factors			Encoding Parameters
Frame size (*width* x *height*)	Picture aspect	Frame rate	Horiz.	Vert.	Max. size encoded	aspect_ratio_idc
320 x 180	1.778	25000÷1000	1	1	320 x 180	1
320 x 240	1.333	25000÷1000	1	1	320 x 240	1
416 x 240 (Note)	1.733	25000÷1000	1	1	416 x 240	1

Note: The 416 x 240 frame size corresponds to a 15.6:9 picture aspect ratio. Recommendations for preparing content in this frame size are available in Section 6 "Video Processing before AVC Compression" of [ATSC].

C.2.3.2.2. Constraints on SD Media Profile Picture Formats

NAL Structured Video streams conforming to the SD Media Profile (see Annex B.2) and this Delivery Target SHALL comply with the picture formats constraints listed below.

Table C - 3 – Picture Formats and Constraints of SD Media Profile for 24 Hz, 30 Hz & 60 Hz Content

Picture Formats			Sub-sample Factors			Encoding Parameters		
Frame size (*width* x *height*)	Picture aspect	Frame rate	Horiz.	Vert.	Max. size encoded	aspect_ratio_idc	sar_width	sar_ height
640 x 480	1.333	24000÷1000 , 24000÷1001 , 30000÷1000 , 30000÷1001	1.1	1	704 x 480	3	-	-
			1	1	640 x 480	1	-	-
			0.75	1	480 x 480	14	-	-
			0.75	0.75	480 x 360	1	-	-
			0.5	0.75	320 x 360	15	-	-
640 x 480	1.333	60000÷1000 , 60000÷1001	$^{464}/_{640}$	0.75	464 x 360	255	30	29
			0.5	0.75	320 x 360	15	-	-
854 x 480	1.778	24000÷1000 , 24000÷1001	1	1	854 x 480	1	-	-
			$^{704}/_{854}$	1	704 x 480	5	-	-
			$^{640}/_{854}$	1	640 x 480	14	-	-
			$^{640}/_{854}$	0.75	640 x 360	1	-	-
			$^{426}/_{854}$	0.75	426 x 360	15	-	-
854 x 480	1.778	30000÷1000 , 30000÷1001	$^{704}/_{854}$	1	704 x 480	5	-	-
			$^{640}/_{854}$	1	640 x 480	14	-	-
			$^{640}/_{854}$	0.75	640 x 360	1	-	-
			$^{426}/_{854}$	0.75	426 x 360	15	-	-
854 x 480	1.778	60000÷1000 , 60000÷1001	$^{426}/_{854}$	0.75	426 x 360	15	-	-

Table C - 4 – Picture Formats and Constraints of SD Media Profile for 25 Hz & 50 Hz Content

Picture Formats			Sub-sample Factors			Encoding Parameters		
Frame size (*width* x *height*)	Picture aspect	Frame rate	Horiz.	Vert.	Max. size encoded	aspect_ ratio_ idc	sar_ width	sar_ height
640 x 480	1.333	25000÷1000	1.1	1.2	704 x 576	2	-	-
			1	1	640 x 480	1	-	-
			0.75	1	480 x 480	14	-	-
			0.75	0.75	480 x 360	1	-	-
			0.5	0.75	320 x 360	15	-	-
640 x 480	1.333	50000÷1000	0.75	0.75	480 x 360	1	-	-
			0.5	0.75	320 x 360	15	-	-
854 x 480	1.778	25000÷1000	1	1	854 x 480	1	-	-
			$^{704}/_{854}$	1.2	704 x 576	4	-	-
			$^{640}/_{854}$	1	640 x 480	14	-	-
			$^{640}/_{854}$	0.75	640 x 360	1	-	-
			$^{426}/_{854}$	0.75	426 x 360	15	-	-
854 x 480	1.778	50000÷1000	$^{560}/_{854}$	0.75	560 x 360	255	9	8
			$^{426}/_{854}$	0.75	426 x 360	15	-	-

C.2.3.2.3. Constraints on HD and xHD Media Profile Picture Formats

NAL Structured Video streams conforming to the HD Media Profile (see Annex B.3) and xHD Media Profile (see Annex B.4) and this Delivery Target SHALL comply with the picture format constraints listed below.

Table C - 5 – Picture Formats and Constraints of HD and xHD Media Profile for 24 Hz, 30 Hz & 60 Hz Content

Picture Formats			Sub-sample Factors			Encoding Parameters
Frame size (*width* x *height*)	Picture aspect	Frame rate	Horiz.	Vert.	Max. size encoded	aspect_ratio_idc
1280 x 720	1.778	24000÷1000, 24000÷1001, 30000÷1000, 30000÷1001, 60000÷1000, 60000÷1001	1	1	1280 x 720	1
			0.75	1	960 x 720	14
			0.5	1	640 x 720	16
1920 x 1080	1.778	24000÷1000, 24000÷1001, 30000÷1000, 30000÷1001	1	1	1920 x 1080	1
			0.75	1	1440 x 1080	14
			0.75	0.75	1440 x 810	1
			0.5	0.75	960 x 810	15

Table C - 6 – Picture Formats and Constraints of HD and xHD Media Profile for 25 Hz & 50 Hz Content

Picture Formats			Sub-sample Factors			Encoding Parameters
Frame size (*width* x *height*)	Picture aspect	Frame rate	Horiz.	Vert.	Max. size encoded	aspect_ratio_idc
1280 x 720	1.778	25000÷1000, 50000÷1000	1	1	1280 x 720	1
			0.75	1	960 x 720	14
			0.5	1	640 x 720	16
1920 x 1080	1.778	25000÷1000	1	1	1920 x 1080	1
			0.75	1	1440 x 1080	14

			0.75	0.75	1440 x 810	1
			0.5	0.75	960 x 810	15

C.2.4. Constraints on Audio

DCCs conforming to this Delivery Target SHALL comply with the requirements and constraints defined in Section 5, Audio Elementary Streams, with the additional constraints defined here.

- A DCC SHALL contain at least one MPEG-4 AAC [2-Channel] audio track.
- A DCC SHALL NOT contain more than 32 audio tracks.
- A DCC with a video track which complies with the SD Media Profile SHALL only contain audio tracks that correspond with Table B-4 “Allowed Audio Formats in SD Media Profile”.

C.2.5. Constraints on Subtitles

DCCs conforming to this Delivery Target SHALL comply with the requirements and constraints defined in Section 6, Subtitle Elementary Streams with the following additional constraints:

- The associated video track SHALL be the video track in the DCC.
- A DCC MAY contain zero or more subtitle tracks, but SHALL NOT contain more than 255 subtitle tracks.
- The duration of a subtitle track SHALL NOT exceed the duration of the longest audio or video track in the DCC.
- A subtitle track fragment MAY have a duration up to the duration of the longest audio or video track in the DCC.
- A subtitle track’s Track Header Box (`‘tkhd’`) SHALL have the same `width` and `height` values as Track Header Box (`‘tkhd’`) for the video track in the DCC.

C.3. Single-Track Download Delivery Target

The Single-Track Download Delivery Target is intended to be used together with [DDMP] to support applications where a DCC is made available for download delivery as part of a Presentation defined within a Common Media Package:

- Unlike the Multi-Track DCCs defined in Annex C.2, a Single-Track DCC is intended to be played together simultaneously with other Single-Track DCCs.
- A "Presentation" is a set of single-track DCCs that are intended to be played together.
- Single-track DCCs and Presentations follow most of the encoding constraints of a track in a Multi-Track DCC (as defined in Annex C.2) except where Common Media Package metadata is utilized for equivalent functionality, such as metadata storage and track description as defined in [DDMP].
- See [DDMP] for more information on the Common Media Package and Presentations.

The Single-Track Download Delivery Target is defined by the `'cfd2'` brand, which is a code point on the ISO Base Media File Format defined by [ISO].

C.3.1. Constraints on File Structure

DCCs conforming to this Delivery Target SHALL comply with the requirements and constraints defined in Annex C.1.1 with the additional constraints defined as follows:

- The File Type Box (`'ftyp'`) SHALL list the `'cfd2'` brand as a `compatible_brand`.
- The DCC SHALL contain one and only one ISO Media track.
- Required Multi-Track Metadata, as defined in Section 2.1.2, and Optional Multi-Track Metadata, as defined in Section 2.1.4, SHOULD NOT be present.
- A Free Space Box (`'free'`) SHOULD NOT be present.
- The DCC Footer SHALL contain a Movie Fragment Random Access Box (`'mfra'`).

C.3.2. Constraints on Encryption

DCCs conforming to this Delivery Target SHALL comply with the additional constraints defined in this Section.

- All video keys SHOULD be separate (independently chosen) from audio keys.

C.3.3. Constraints on Video

Single-Track Video DCCs conforming to this Delivery Target SHALL comply with the video constraints defined in Annex C.1.2 with the additional constraints/exceptions defined as follows:

- AVC Video tracks SHALL utilize the (`'avc3'`) in-band sample entries as per Section 4.3.1.1.

C.3.4. Constraints on Audio

Single-Track Audio DCCs conforming to this Delivery Target SHALL comply with the requirements and constraints defined in Section 5, Audio Elementary Streams.

A Presentation utilizing this Delivery Target SHALL contain at least one MPEG-4 AAC [2-Channel] audio track.

C.3.5. Constraints on Subtitles

Single-Track Subtitle DCCs conforming to this Delivery Target SHALL comply with the requirements and constraints defined in Section 6, Subtitle Elementary Streams with the following additional constraints.

- The associated video track SHALL be the video track contained in the same Presentation as the subtitle track.

.

C.4. Pre-Packaged Delivery Target

The Pre-Packaged Delivery Target is intended to be used together with [DDMP] to support applications where a single-track DCC is made available with pre-packaged delivery as part of a Presentation defined within a Common Media Package; pre-packaged delivery means that the content is delivered in complete form e.g. on an optical disc.

DCCs conforming to this Delivery Target SHALL comply with all of the requirements and constraints defined in Annex C.3 except as defined below.

The Pre-Packaged DCC Delivery Target is defined by the `‘cfd3’` brand, which is a code point on the ISO Base Media File Format defined by [ISO].

C.4.1. Constraints on File Structure

DCCs conforming to this Delivery Target SHALL comply with the requirements and constraints defined in Annex C.3.1 with the additional constraints and exceptions defined as follows:

- The File Type Box (`‘ftyp’`) SHALL list the `‘cfd2’` brand as a `compatible_brand` only if all requirements defined in Annex C.3 are satisfied.
- The File Type Box (`‘ftyp’`) SHALL list the `‘cfd3’` brand as a `compatible_brand`.
- The DCC Header SHALL contain one Segment Index Box (`‘sidx’`). The sample at the start of each Movie Fragment SHALL be signaled in the Segment Index Box (`‘sidx’`).

C.4.2. Constraints on Video

Single-Track Video DCCs conforming to this Delivery Target SHALL comply with the requirements and constraints defined in Annex C.3.3 with the additional constraints and exceptions defined as follows:

- Video track Movie Fragments SHALL start with SAP type 1, 2, or 3.

C.5. Streaming Delivery Target

The Streaming Delivery Target is intended to be used together with [DStream] to support applications where a single-track DCC is made available for streaming delivery as part of an Adaptation Set within a [DASH] Media Presentation Description:

- Unlike the Single-Track DCCs defined in Annex C.3 and C.4, a Streaming DCC is intended to be one of several DCCs in an "Adaptation Set".
- An "Adaptation Set" is a set of single-track DCCs that are interchangeable encoded versions of the same Content which are seamlessly switchable in [DASH] adaptive streaming applications.
- "Seamlessly Switchable" means different Movie Fragments can be selected in sequence from different DCCs without presentation errors.
- A "Media Presentation Description" defines Adaptation Sets that are intended to be played together (similar to a Presentation as defined in Annex C.3).
- See [DStream] for more information on the Adaptation Sets and Media Presentation Descriptions.

The Streaming DCC Delivery Target is defined by the `'cfd4'` brand, which is a code point on the ISO Base Media File Format defined by [ISO].

C.5.1. Constraints on File Structure

DCCs conforming to a Streaming Delivery Target SHALL comply with all of the requirements and constraints defined in Section 2, The Common File Format, with additional constraints defined as follows:

- The File Type Box (`'ftyp'`) SHALL list the `'cfd4'` brand as a `compatible_brand`.
- The DCC Movie Fragment SHALL contain one Media Data Box (`'mdat'`).
- The DCC SHALL contain one and only one ISO Media track.
- The Segment Index Box (`'sidx'`), if present, SHALL comply with Section 2.3.21 and [DASH] Section 6.3.4.

C.5.2. Constraints on Encryption

DCCs conforming to this Delivery Target SHALL comply with additional constraints defined in this Section.

- All encrypted audio tracks listed by a Period in a [DStream] MPD SHOULD be encrypted using the same key ("audio key"). See also Annex C.5.6 "Constraints on Adaption Sets".
- All encrypted video tracks listed by a Period in a [DStream] MPD SHOULD be encrypted using the same key ("video key"). See also Annex C.5.6 "Constraints on Adaption Sets".
- All video keys SHOULD be separate (independently chosen) from the audio keys.

C.5.3. Constraints on Video

DCCs conforming to a Streaming Delivery Target SHALL comply with all of the requirements and constraints defined in Section 4, Video Elementary Streams, with additional constraints defined as follows:

- DCCs SHALL contain exactly one video track, and that video track SHALL conform to Section 4.
- AVC Video tracks SHALL utilize the (`'avc3'`) in-band sample entries as per Section 4.3.1.1.
- Video track Movie Fragments SHALL start with SAP type 1 or 2.

- negative composition offsets in the Track Run Box (`'trun'`) (see Section 2.4) SHALL be used to synchronize the composition times of video samples to the decode time of other tracks so that each sample is frame accurately synchronized to the movie and [DASH] MPD presentation timeline. Note: use of negative composition offsets provides a mechanism to match the presentation time of the first video frame to the decode time of Movie Fragments, and prevents gaps or overlaps in sample presentation times when switching between [DStream] Representations with different composition offsets due to different numbers of pictures in the decoded picture buffer necessary to decode and present the first video frame. [DStream] MPDs, Segment Index Box (`'sidx'`) indexes, and Segment addresses are based on the presentation timeline of the media, but DCC Movie Fragments are stored and time stamped with decode time.

C.5.3.1. Constraints on Picture Formats

Video tracks conforming to this Delivery Target MAY utilize dynamic subsampling of each Coded Video Sequence within a DCC Movie Fragment by allowing different encoded horizontal and vertical video spatial sample counts in each Coded Video Sequence (see Section 4.5.4).
Note: Dynamic subsampling within a CSF file is normally used to reduce video bitrate peaks and maintain consistent Media Segment size.
Note: NAL Structured Video parameter sets prevent the need to insert an Initialization Segment prior to each Media Segment in case the decoding parameters change due to dynamic subsampling or adaptive switching between DCCs ([DASH] Representations); the parameter sets provide the information necessary for display systems to scale each Coded Video Sequence to a common display resolution and position.

C.5.4. Constraints on Audio

Single-Track Audio DCCs conforming to a Streaming Delivery Target SHALL comply with the requirements and constraints defined in in Section 5, Audio Elementary Streams.
A [DStream] MPD utilizing this Delivery Target SHALL contain at least one MPEG-4 AAC [2-Channel] audio track.

C.5.5. Constraints on Subtitles

Single-Track Subtitle DCCs conforming to a Streaming Delivery Target SHALL comply with the requirements and constraints defined in Section 6, Subtitle Elementary Streams with additional constraints defined as follows:

- The associated video track SHALL be the video track contained in the same Period as the subtitle track.

C.5.6. Constraints on Adaptation Sets

Adaptation Sets conforming to this Delivery Target SHALL comply with all of the requirements and constraints defined in [DStream] with the additional constraints defined as follows:

- All encrypted audio tracks within an Adaption Set SHALL use the same key; and
- All encrypted video tracks within an Adaption Set SHALL use the same key.; and

- All video tracks in the same Adaption Set SHALL have matching `SampleEntry` type; and
- All DCC Adaptation Sets in a [DStream] MPD Period SHALL have the same Movie Header Box (`'mvhd'`) `timescale` value; and
  - Note: For a [DStream] Presentation, the Movie Header Box (`'mvhd'`) `timescale` is applied to the entire [DASH] Presentation Period, which is an independent timespan within the Presentation. This `timescale` value is independent of the track `timescale` in the Media Header Box (`'mdhd'`), which is specific to the media in the containing track, and should be different for different media sample frequencies in the same Period.
- All DCCs in the same Adaptation Set have matching Track Header Box (`'tkhd'`) fields as follows:
  - the values of `width` and `height` fields SHALL match; and
  - the values of the `timescale` field SHALL match; and
  - the values of the `track_ID` field SHALL match; and
- All DCCs in the same Adaptation Set SHALL have the same `default_KID` value for the Track Encryption Box (`'tenc'`) if the track is encrypted; and
- All DCCs in the same Adaptation Set SHALL be "time aligned" i.e. all DCCs in the same Adaptation Set SHALL have matching track fragment durations for all Movie Fragments with the same `baseMediaDecodeTime` and Movie Fragment `sequence_number`; and
- All DCCs in the same Adaptation Set SHALL be "spatially aligned" i.e. all encoded and cropped sample counts SHALL be exact ratios of the Normalized Display Width and Normalized Display Height; and
- All DCCs in a Group of Adaptation Sets MAY use the same `track_ID` value to identify all DCCs and Adaptation Sets in the Group i.e. [DStream] AdaptationSet@group attribute in MPD = `track_ID` in the `'tkhd'`, `'trex'`, and `'tfhd'` boxes); and
- [DStream] AdaptationSet@group attribute SHALL follow the assignment in Table C-7.

Table C - 7 – [DStream] Group Assignment

[DStream] Group	Track Type
1	Primary Video
2	Secondary Video
5	Main Audio
6	Secondary Audio
7	Tertiary Audio
3	Main Subtitle
4	Secondary Subtitle
8	Other

Note: Any DCC file header in the Adaptation Set followed by a continuous and ordered sequence of movie fragments from that DCC or a DCC in the Adaptation Set with lower video codec Level, Profile, and Tier will form a valid DCC and therefore conform to @bitstreamSwitching="true" per [DASH]. A single initialization of a player's video parsing, decryption, decoding, and presentation system is sufficient to present movie fragments selected from that subset of the Adaptation Set.

Annex D. Internet Media Type Template and Parameters

The Media Type signaling defined in this Section SHALL be used when delivering DCCs.

- Media Type Template: SHALL be set to "`video/vnd.dece.mp4`".
- Media Type Parameters:
    - `profiles`: SHALL list all ISO brands signaled in the File Type Box (`'ftyp'`) in accordance with [RFC6381] Section 4.
    - `profile-level-idc`: SHALL list the [MP4RA] four character code with the Media Profile of each track in a comma separated list, in ascending `track_id` order. The Media Profile of the track is defined as follows.
        - Video track: SHALL define the Media Profile associated with the Video track.
        - Audio track: SHALL signal the lowest Media Profile supported by the Audio track down to the SD Media Profile as defined in Annex B.2 (i.e. the PD Media Profile is not signaled).
        - Subtitle track: SHALL signal the Media Profile associated with the Subtitle track as follows:
            - If the Subtitle track has any CFF-TT document that has been authored for a specific Media Profile or set of Media Profiles, the lowest Media Profile supported by the Subtitle track SHALL be signaled.
                - Note: CFF-TT documents are authored for a specific Media Profile if they conform to the CFF-TT Image based Profile, or if they conform to CFF-TT Text based Profile and one or more of these documents uses scalar length attributes.
            - If the Subtitle track contains CFF-TT documents that have all been authored to be Media Profile independent then the `'cfad'` code registered with [MP4RA] SHALL be signalled.
                - Note: CFF-TT documents can be Media Profile independent if they conform to the CFF-TT Text based Profile, no scalar length attributes are used and the performance constraints defined in Annex B.1.5 are not exceeded.
    - `codecs`: SHALL list each track, in ascending `track_id` order, in accordance with [RFC6381] Section 3.3. The [MP4RA] codecs code point SHALL be used for video and audio tracks. "cfft" SHALL be used for CFF-TT tracks utilizing the Text Profile (see Section 6.2.3.4) and "cffi" SHALL be used for CFF-TT tracks utilizing the Image Profile (see Section 6.2.3.5).
    - `protection`: SHALL list which protection scheme applies to each track, in ascending track_id order in a comma separated list, using the [MP4RA] four character code registered protection scheme value or 'none' to indicate that the track is not protected.
    - `languages`: SHALL list the associated language of all tracks, in ascending `track_id` order in a comma separated list, in accordance with [RFC5646]. The video track language SHOULD represent the original release language of the content.

Annex E. The DECE File Format

The DECE File Format is based on the Common File Format, with enhancements to support the DECE Ecosystem. See [DSystem] for more information on the DECE Ecosystem.

E.1. DECE File Format

This Section defines an adaption of the Common File Format for the DECE Ecosystem.

The DECE File Format is defined by the `'uvvu'` brand, which is a code point on the ISO Base Media File Format defined by [ISO]. The brand `'uvvu'` SHALL incorporate support for all features of the `'ccff'` brand defined in Section 2 with additional requirements defined in this Annex.

E.1.1. Extensions to the Common File Format

The Section defines additional boxes that SHALL be supported by the brand `'uvvu'` in addition to the boxes required by the `'ccff'` brand.

Table D - 1 – Additional Boxes of the DECE File Format

NL 0	NL 1	NL 2	NL 3	NL 4	NL 5	Format Req.	Specification	Description
bloc						0/1	Annex E.1.1.1	Base Location Box
	ainf					0/1	Annex E.1.1.2	Asset Information Box

Format Req.: Number of boxes required to be present in the container, where '*' means "zero or more" and '+' means "one or more". A value of "0/1" indicates only that a box might or might not be present but does not stipulate the conditions of its appearance.

E.1.1.1. Base Location Box (`'bloc'`)

Box Type `'bloc'`
Container File
Mandatory No
Quantity Zero or One

The Base Location Box is a fixed-size box that contains the Base Location and Purchase Location strings necessary for license acquisition, as defined in Sections 8.3.2 and 8.3.3 of [DSystem].

The Base Location Box (`'bloc'`), if present in the DCC:

- SHOULD immediately follow the File Type Box (`'ftyp'`) in the DCC Header; and
- SHALL appear before the Movie Box (`'moov'`).

E.1.1.1.1. Syntax

```
aligned(8) class BaseLocationBox
   extends FullBox('bloc', version=0, flags=0)
{
   byte[256]  baseLocation;
   byte[256]  basePurlLocation;  // optional
   byte[512]  reserved = 0;
}
```

E.1.1.1.2. Semantics

- baseLocation – SHALL contain the Base Location defined in Section 8.3.2 of [DSystem], followed by null bytes (0x00) to a length of 256 bytes.
- basePurlLocation – optionally defines the Base Purl Location as specified in Section 8.3.3 of [DSystem], followed by null bytes (0x00) to a length of 256 bytes. If no Base Purl Location is defined, this field SHALL be filled with null bytes (0x00).
- Reserved – Reserve space for future use. Implementations conformant with this specification SHALL ignore this field.

E.1.1.2. Asset Information Box (‘ainf’)

The Asset Information Box (‘ainf’) SHOULD NOT be used. It is provided for backwards compatibility with previous releases of this specification. The Content Information Box (‘coin’) defined in Section 2.2.2 replaces the use the Asset Information Box (‘ainf’).

Box Type	‘ainf’
Container	Movie Box (‘moov’)
Mandatory	Yes
Quantity	Zero or One

The Asset Information Box (‘ainf’) contains file information to identify, license and play content within the DECE ecosystem.

The Asset Information Box (‘ainf’), if present in the DCC:

- SHOULD immediately follow the Content Information Box (‘coin’) in the DCC Header; and
- SHALL appear before the Movie Box (‘moov’).

E.1.1.3. Syntax

```
aligned(8) class AssetInformationBox
   extends FullBox(‘ainf’, version=0, flags)
{
   int(32)  profile_version;
   string   APID;
   Box      other_boxes[];    // optional
}
```

E.1.1.4. Semantics

- flags – 24-bit integer defined as follows:

  hidden: when set indicates that file should not be visible to the user. Flag value is ‘0x000001’.
- profile_version – indicates the Media Profile to which this container file conforms. The most significant 8 bits designate the profile (PD, SD, HD, etc.) as defined in the Annexes of this specification. The least significant 24 bits SHALL be set to the [UNICODE] UTF-8 representation of this DMedia Specification as defined in Annex A. , DMEDIA_VERSION_NOPOINTS.

- APID – indicates the Asset Physical Identifier (APID) of this container file, as defined in Section 5.5.1 “Asset Identifiers” of [DSystem].
- other_boxes – Available for private and future use.

E.1.2. Constraints on ISO Base Media File Format Boxes

E.1.2.1. File Type Box (‘ftyp’)

DCCs conforming to the DECE File Format SHALL include a File Type Box (‘ftyp’) as specified in Section 2.3.1 with the following constraints:

- The ‘uvvu’ brand SHALL be set as a compatible_brand in the File Type Box (‘ftyp’). Note: signaling of the ‘uvvu’ brand indicates that the file fully complies with the requirements in this Annex.
  - If the major_brand field is set to ‘uvvu’, the minor_version field SHALL be set to the integer representation of DMEDIA_VERSION_NOPOINTS (defined in Annex A.).

E.1.3. Constraints on ‘ccff’ Boxes

E.1.3.1. Content Information Box (‘coin’)

DCCs conforming to the DECE File Format shall include a Content Information Box (‘coin’) as specified in Section 2.2.2 with the following constraints:

- Brand:
  - An iso_brand field SHALL be included with the value “uvvu”.
  - The version field which accompanies the above iso_brand value SHALL be set to the integer representation of DMEDIA_VERSION_NOPOINTS (defined in Annex A.).
- Asset:
  - A namespace field SHALL be included with the value “urn:dece”.
  - The asset_id field which accompanies the above namespace field SHALL be set to the Asset Physical Identifier (APID) of the DCC, as defined in Section 5.5.1 “Asset Identifiers” of [DSystem].

E.2. Media Profiles

The Media Profiles defined in Annex B. are not extended or constrained by the ‘uvvu’ brand.

E.3. Delivery Targets

The Delivery Targets defined in Annex C. apply to the ‘uvvu’ brand with the additional requirements defined in this Section.

E.3.1. Multi-Track Download Delivery Target

DCCs conforming to this Delivery Target and the ‘uvvu’ brand SHALL comply with the requirements and constraints defined in Annex C.2 (Multi-Track Download Delivery Target) with the additional requirements defined in this Section.

E.3.1.1. Constraints on File Structure

DCCs conforming to this Delivery Target and the ‘uvvu’ brand SHALL comply with all of the requirements and constraints defined in Annex C.2.1 with the additional constraints defined as follows:

- The Base Location Box (‘bloc’) SHALL be present in the DCC.

E.4. Internet Media Type Template and Parameters

The Internet Media Type Template and Parameter requirements defined in Annex D. apply for the DECE File Format.

E.5. File Extensions

The file extensions defined in Table E-1 apply for the DECE Ecosystem.

Table E - 1 – DECE File Format File Extension Requirements

DCC Type	File Extension
DECE Multi-Track DCC	.uvu
DECE Video Single-Track DCC	.uvv
DECE Audio Single-Track DCC	.uva
DECE Subtitle Single-Track DCC	.uvt

E.6. DECE Interoperability Points

The Tables below define which combinations of Media Profiles, Delivery Targets and mandatory codecs are supported by the DECE Ecosystem. These combinations are called “Interoperability Points”.
These Tables use the following headings:

- “Interoperability Point” – identifies the Interoperability Point.
- “Media Profile” – refers to a Media Profile defined in Annex B.
- “Delivery Target” – refers to a Delivery Target defined in Annex C.
- “Mandatory Video Codecs” – defines which video codecs, within the listed Media Profile, are mandated by the Interoperability Point. Content that complies with the Interoperability Point will have at least one video track available which is encoded with a mandatory video codec.
- “Mandatory Audio Codecs” – defines which audio codecs within the listed Media Profile are mandated by the Interoperability Point. Content that complies with the Interoperability Point will have at least one audio track available which is encoded with a mandatory audio codec.

Table E - 2 – AVC Download and Prepackaged Interoperability Points

Interoperability Point	Media Profile	Delivery Target	Mandatory Video Codecs	Mandatory Audio Codecs
SD-MDL	SD	Multi-Track Download	AVC	MPEG-4 AAC 2-Channel

HD-MDL	HD	Multi-Track Download	AVC	MPEG-4 AAC 2-Channel
SD-AVC-SDL	SD	Single-Track Download	AVC	MPEG-4 AAC 2-Channel
HD-AVC-SDL	HD	Single-Track Download	AVC	MPEG-4 AAC 2-Channel
SD-AVC-PP	SD	Pre-Packaged	AVC	MPEG-4 AAC 2-Channel
HD-AVC-PP	HD	Pre-Packaged	AVC	MPEG-4 AAC 2-Channel

Table E - 3 – AVC Streaming Interoperability Points

Interoperability Point	Media Profile	Delivery Target	Mandatory Video Codecs	Mandatory Audio Codecs
SD-AVC-STR	SD	Streaming	AVC	MPEG-4 AAC 2-Channel
HD-AVC-STR	HD	Streaming	AVC	MPEG-4 AAC 2-Channel

Table E - 4 – HEVC Download and Prepackaged Interoperability Points

Interoperability Point	Media Profile	Delivery Target	Mandatory Video Codecs	Mandatory Audio Codecs
SD-HEVC-SDL	SD	Single-Track Download	AVC or HEVC	MPEG-4 AAC 2-Channel
HD-HEVC-SDL	HD	Single-Track Download	AVC or HEVC	MPEG-4 AAC 2-Channel
SD-HEVC-PP	SD	Pre-Packaged	AVC or HEVC	MPEG-4 AAC 2-Channel
HD-HEVC-PP	HD	Pre-Packaged	AVC or HEVC	MPEG-4 AAC 2-Channel

Table E - 5 –HEVC Streaming Interoperability Points

Interoperability Point	Media Profile	Delivery Target	Mandatory Video Codecs	Mandatory Audio Codecs
SD-HEVC-STR	SD	Streaming	AVC or HEVC	MPEG-4 AAC 2-Channel
HD-HEVC-STR	HD	Streaming	AVC or HEVC	MPEG-4 AAC 2-Channel

Note: the PD and xHD Media Profiles are not currently supported by the DECE Ecosystem.

END