

AV2 Codec ISO Media File Format Binding

Working Group Draft, 15 July 2026



This version:

<https://aomediacodec.github.io/av2-isobmff/v1.0.0.html>

Issue Tracking:

[GitHub](#)

Editors:

TBD

TBD

Warning

This specification is still at draft stage and should not be referenced other than as a working draft.

Copyright 2026, [Alliance for Open Media](#)

Licensing information is available at <https://www.aomedia.org/license/>

The MATERIALS ARE PROVIDED “AS IS.” The Alliance for Open Media, its members, and its contributors expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to the materials. The entire risk as to implementing or otherwise using the materials is assumed by the implementer and user. IN NO EVENT WILL THE ALLIANCE FOR OPEN MEDIA, ITS MEMBERS, OR CONTRIBUTORS BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS DELIVERABLE OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Abstract

This document specifies the storage format for [AV2] bitstreams in [ISOBMFF] tracks as well as in [CMAF] files.

Table of Contents

Pages

1 Bitstream features overview

3

2 Basic Encapsulation Scheme

4

2.1 General Requirements & Brands

4

2.2 AV2 Sample Entry and Configuration

4

2.2.1 AV2 Sample Entry

4

2.2.2 AV2 Codec Configuration Box

5

2.2.3 configOBUs

6

2.3 AV2 Sample Format

6

2.3.1 General

6

2.3.2 OBUs in samples

7

2.3.3 Sync sample

8

3 CMAF AV2 track format

8

4 Common Encryption

8

5 Codecs Parameter String

8

Conformance

8

Index

9

Terms defined by this specification

9

Terms defined by reference

10

References

11

Normative References

11

List of Figures

Figure 2.1: Example mapping of AV2 temporal units into the AV2 ISOBMFF sample format.

5

§ 1. Bitstream features overview

An [AV2 bitstream](#) is composed of a sequence of [OBUs](#) (Open Bitstream Units), grouped into [Temporal Units](#). A Temporal Unit consists of a [Temporal Delimiter OBU](#) and all following OBUs up to, but not including, the next Temporal Delimiter OBU. Each Temporal Unit is associated with a single presentation time.

Every OBU begins with a 1- or 2-byte header that carries the [obu_type](#) and the temporal layer identifier [obu_tlayer_id](#). When the OBU header extension is present (i.e. when [obu_header_extension_flag](#) is equal to 1 in that header), two additional layer identifiers are carried: the embedded layer identifier [obu_mlayer_id](#) and the extended layer identifier [obu_xlayer_id](#). Depending on its type, an OBU can carry configuration information (e.g., a [Sequence Header OBU](#) or [Layer Configuration Record OBU](#)), metadata, or coded video data.

A Temporal Unit may contain multiple coded pictures but only one is output. AV2 defines a number of coded-frame OBU types, including [Closed Loop Key OBUs](#), [Open Loop Key OBUs](#), regular and leading tile-group OBUs, [Switch OBUs](#), [OBU_RAS_FRAME](#), [Bridge Frame OBUs](#), and TIP (Temporally Interpolated Prediction) frames.

Frames that can be decoded without reference to other frames are [Key Frames](#) (carried in a [Closed Loop Key OBU](#) or an [Open Loop Key OBU](#)) and [Intra frames](#). All other coded frames - including [Inter Frames](#), [Switch Frames](#), RAS frames, Bridge frames, TIP frames, and frames with [show_existing_frame](#) equal to 1 - have coding dependencies on other frames in the same or previous Temporal Units.

A [Closed Random Access](#) is the random access process that applies to an extended layer whose first coded frame is a [Key Frame](#) carried in a [Closed Loop Key OBU](#). It starts a new coded video sequence for the extended layer; the layer's reference frame buffers are invalidated and every subsequent frame of that layer can be decoded. This is the primary random-access mechanism.

An [Open Random Access](#) is the random access process that applies to an extended layer whose first coded frame is a [Key Frame](#) carried in an [Open Loop Key OBU](#). During sequential decoding, an [Open Random Access](#) point does not start a new coded video sequence; however, when a decoder initiates decoding at that point, the decoding process is treated as if it were the start of a new coded video sequence for the extended layer. [Leading frames](#) that follow an [Open Random Access](#) point may reference frames that precede it, and are discarded when decoding begins at the [Open Random Access](#) point.

In addition to Key Frames, AV2 provides two further mechanisms that support random access or stream switching: a RAS frame ([OBU_RAS_FRAME](#)) is an inter-predicted frame that uses long-term reference frames and enables random access or switching without inserting a full key frame; a [Switch Frame](#) ([OBU_SWITCH](#)) is an inter-predicted frame that enables switching between representations at a given point, typically in adaptive streaming scenarios.

AV2 supports three scalability dimensions: temporal layers (identified by [obu_tlayer_id](#)), embedded layers (identified by [obu_mlayer_id](#)), and extended layers (identified by [obu_xlayer_id](#)). The reserved [obu_xlayer_id](#) value [GLOBAL_XLAYER_ID](#) (equal to 31, as defined in [\[AV2\]](#)) denotes global scope and is used by OBUs that do not belong to any specific extended layer (such as the [Temporal Delimiter OBU](#)). A bitstream that uses a single distinct non-global value of [obu_xlayer_id](#) is a [Singlestream](#); a bitstream that uses two or more distinct non-global values is a [Multistream](#).

AV2 defines two distinct [Metadata OBU](#) types for carrying metadata: `OBU_METADATA_SHORT`, using the [metadata short OBU syntax](#), and `OBU_METADATA_GROUP`, using the [metadata group OBU syntax](#). A separate OBU type carries film grain synthesis parameters; its presence is indicated by the `film_grain_params_present` flag in the [Sequence Header OBU](#).

The [Sequence Header OBU](#) also carries the bitstream's profile, level, and - when applicable - tier, as defined in Annex A of [\[AV2\]](#).

The [Sequence Header OBU](#) carries a `monotonic_output_order_flag` that defines the output mode of the associated coded video sequence.

When `monotonic_output_order_flag` is equal to 1, the output order of coded output frame equals their decoding order within the coded video sequence. Within a sample, the composition time (CTS) of the associated output frame is equal to the sample's decoding time (DTS); the sample may additionally contain hidden frames.

When `monotonic_output_order_flag` is equal to 0, the output order of coded output frames may differ from their decoding order. The composition time of each sample is then derived from a `'ctts'` box that signals the offset between the sample's decoding time (DTS) and the composition time of its associated output frame. As in the monotonic case, a sample may additionally contain hidden frames.

NOTE: Non-monotonic output mode is new in AV2; AV1 supports only the monotonic mode.

§ 2. Basic Encapsulation Scheme

This section describes the basic data structures used to signal encapsulation of [AV2 bitstreams](#) in [\[ISOBMFF\]](#) containers.

§ 2.1. General Requirements & Brands

A file conformant to this specification satisfies the following:

- It shall conform to the normative requirements of [\[ISOBMFF\]](#)
- It shall have the `'av02'` brand among the compatible brands array of the `FileTypeBox`
- It shall contain at least one track using an [AV2SampleEntry](#), possibly transformed by encryption as specified in [§4 Common Encryption](#)
- It should indicate a structural ISOBMFF brand among the compatible brands array of the `FileTypeBox`, such as `'iso6'`

§ 2.2. AV2 Sample Entry and Configuration

§ 2.2.1. AV2 Sample Entry

The **AV2SampleEntry** extends [VisualSampleEntry](#) and signals a track carrying AV2 coded video samples.

```
class AV2SampleEntry extends VisualSampleEntry('av02') {  
    AV2CodecConfigurationBox config;  
}
```

An [AV2SampleEntry](#) shall contain exactly one [AV2CodecConfigurationBox](#).

An [AV2SampleEntry](#) shall contain a ['colr'](#) box with the ['nclx'](#) colour type. The values of `colour primaries`, `transfer characteristics`, `matrix coefficients` and `full_range_flag` carried in that ['colr'](#) box should equal the corresponding effective values of the AV2 bitstream carried by the track.

NOTE: An [AV2SampleEntry](#) may contain additional sub-boxes permitted by [\[ISOBMFF\]](#) for a [VisualSampleEntry](#), such as ['pasp'](#), ['clli'](#), ['mdcv'](#) or ['btrt'](#).

§ 2.2.2. AV2 Codec Configuration Box

The ***AV2CodecConfigurationBox*** carries the decoder initialisation data required to decode the AV2 coded video samples of a track.

```
aligned(8) class AV2CodecConfigurationBox extends Box('av2C')
{
    unsigned int(8)    configurationVersion = 1;

    unsigned int(5)    seq_profile_idc;
    unsigned int(5)    seq_level_idc;
    unsigned int(1)    seq_tier;

    unsigned int(3)    chroma_format_idc;
    unsigned int(3)    bit_depth_idc;

    unsigned int(1)    monotonic_output_order_flag;
    unsigned int(1)    still_picture;
    unsigned int(1)    film_grain_params_present;
    unsigned int(1)    seq_initial_display_delay_present_flag;
    unsigned int(4)    seq_initial_display_delay_minus_1;
    unsigned int(7)    reserved1 = 0;

    unsigned int(16)    max_frame_width_minus_1;
    unsigned int(16)    max_frame_height_minus_1;

    unsigned int(8)    configOBUs[];
}
```

semantics TBD

NOTE: In [\[AV2\]](#), `chroma_format_idc` and `bit_depth_idc` are coded with `uvlc()` and therefore have no inherent upper bound, even though [\[AV2\]](#) currently defines only a small number of values for each (4 and 2 respectively). The 3-bit fields used here accommodate all values currently defined by [\[AV2\]](#) with additional headroom for future extensions. If a future revision of [\[AV2\]](#) defines values that do not fit in 3 bits, a new `configurationVersion` of the [AV2CodecConfigurationBox](#) can be defined.

`reserved` shall be set to 0. Readers shall ignore its value.

`configOBUs` is a byte array carrying one or more OBUs as described in [§ 2.2.3 configOBUs](#).

§ 2.2.3. configOBUs

Each OBU carried in configOBUs shall be framed using the [Length delimited bitstream format](#) specified in Annex B of [AV2]: each OBU is preceded by a `leb128() num_bytes_in_obu` value giving the byte length of the OBU that follows.

configOBUs shall contain a [Sequence Header OBU](#) as its first OBU.

configOBUs may contain additional OBUs whose content is constant for the track's samples, such as static [Metadata OBUs](#).

configOBUs shall NOT contain any of the following:

- a [Temporal Delimiter OBU](#);
- any OBU carrying coded frame data, including [Closed Loop Key OBUs](#), [Open Loop Key OBUs](#), [Switch OBUs](#), [Bridge Frame OBUs](#), regular or leading tile-group OBUs, and switch-eligible or TIP frame OBUs;

The values of the fields in the [AV2CodecConfigurationBox](#) shall equal the corresponding effective values of the fields of the [Sequence Header OBU](#) carried in configOBUs.

§ 2.3. AV2 Sample Format

§ 2.3.1. General

Each AV2 sample carries exactly one [Temporal Unit](#) of the [AV2 bitstream](#); the sample boundary delimits the temporal unit.

Each [OBU](#) within a sample shall be framed using the [Length delimited bitstream format](#) specified in Annex B of [AV2]: each OBU is preceded by a `leb128() num_bytes_in_obu` field giving the byte length of the OBU that follows. [Figure 2.1](#) illustrates this mapping.

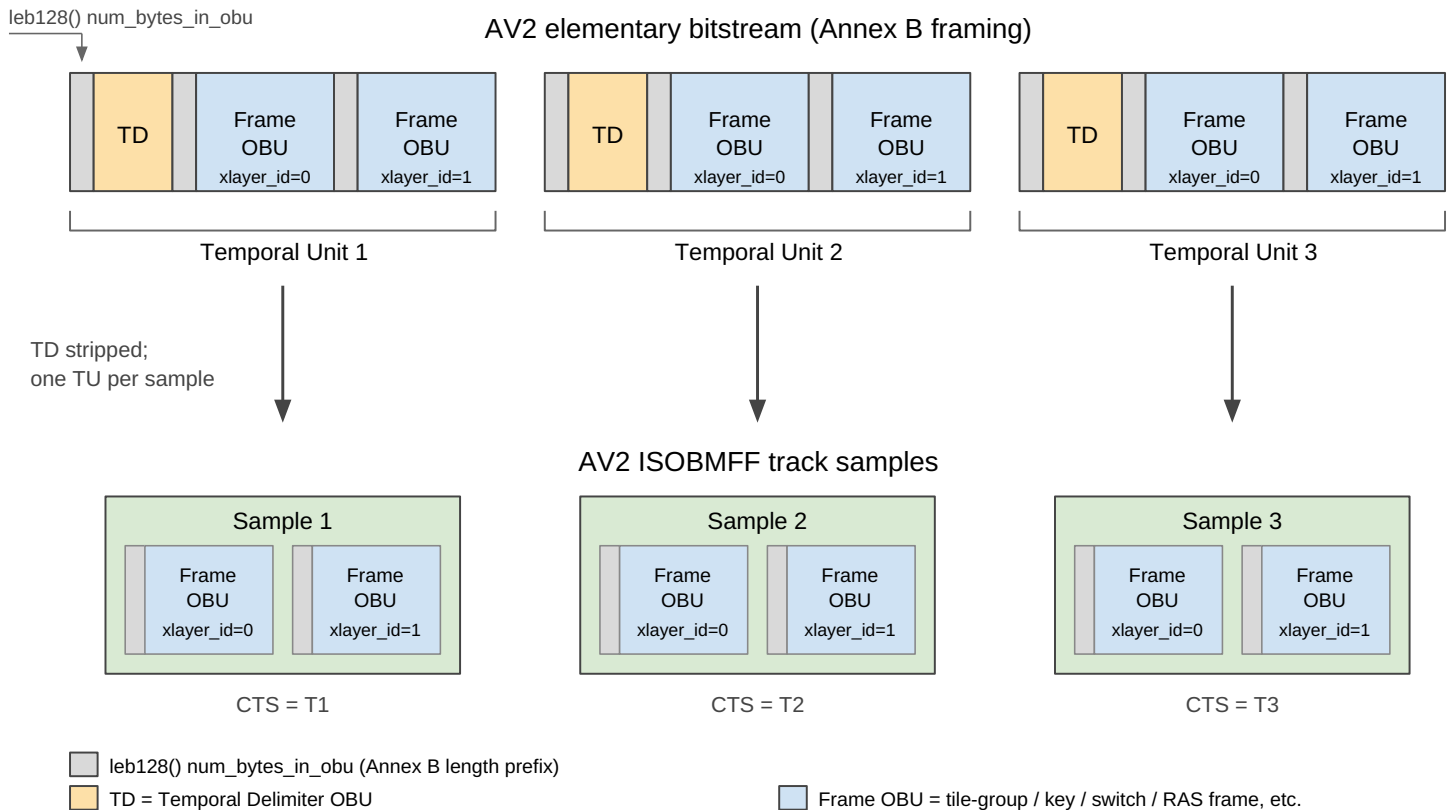


Figure 2.1: Example mapping of AV2 temporal units into the AV2 ISOBMFF sample format.

§ 2.3.2. OBUs in samples

The OBUs within a sample shall appear in the order specified by the [ordering of OBUs in a temporal unit](#) in [\[AV2\]](#), with the following constraints:

- a sample shall not contain any OBU with obu_type equal to OBU_TEMPORAL_DELIMITER, since the sample boundary already delimits the temporal unit;
- a sample shall not contain any OBU with obu_type equal to OBU_PADDING; byte-level padding within a track is achieved using ISOBMFF mechanisms (e.g., [‘free’/‘skip’](#) boxes or sample size signalling);
- the trailing bits of an OBU carried in a sample shall be limited to those required for byte alignment as specified in [\[AV2\]](#); trailing bits shall not be used to pad an OBU beyond byte alignment;
- a sample shall not contain a [Sequence Header OBU](#); sequence headers are carried in the configOBUs of the [AV2CodecConfigurationBox](#) of the active [AV2SampleEntry](#);
- a sample shall not contain a [Layer Configuration Record OBU](#); layer configuration records are carried in the configOBUs of the active [AV2SampleEntry](#);
- a sample should not contain any other OBU whose contents do not change for the duration of the active [AV2SampleEntry](#), in order to avoid redundant data; for example, a [Metadata OBU](#) whose contents are constant should be carried in configOBUs rather than repeated in every sample, or discarded from the samples and conveyed at the ISOBMFF level.

The handling of the [Temporal Delimiter OBU](#) depends on how a track’s samples are consumed. When the samples are passed to an AV2 decoder API that already treats each sample as one [Temporal Unit](#), a [Temporal Delimiter OBU](#) is not required. When the samples are reconstructed into an [AV2 bitstream](#) in the

[Length delimited bitstream format](#) specified in Annex B of [\[AV2\]](#), or into any other carriage that does not itself delimit temporal units, a [Temporal Delimiter OBU](#) (framed with its `leb128()` length prefix) shall be prepended ahead of each sample's OBUs that does not already start with one.

§ 2.3.3. Sync sample

A sample is a sync sample if, for every [coded extended layer unit](#) present in the sample, the first coded frame OBU of that coded extended layer unit has `obu_type` equal to `OBU_CLOSED_LOOP_KEY`. Such a sample is a [Closed Random Access](#) point for the carried bitstream and corresponds to a SAP of type 1 or 2 as defined in Annex I of [\[ISOBMFF\]](#).

A sync sample shall be self-sufficient for starting decoding at that point for all layers: every OBU referenced by the coded frame OBUs of the sync sample shall be available either by being carried in `configOBUs` of the active [AV2SampleEntry](#) or by being present in the sync sample itself, as required by the [availability of HLS OBUs](#) in [\[AV2\]](#).

NOTE: In ISOBMFF terms, sync samples are signalled using the SyncSampleBox ('[stss](#)') in unfragmented tracks, or by setting `sample_is_non_sync_sample` equal to 0 in fragmented tracks. When every sample in a track is a sync sample, the SyncSampleBox is omitted in unfragmented tracks (per [\[ISOBMFF\]](#)), and in fragmented tracks this is expressed through the default sample flags of the MovieExtendsBox ('[trex](#)') or TrackFragmentHeaderBox ('[tfhd](#)') with no per-sample override.

§ 3. CMAF AV2 track format

TBD

§ 4. Common Encryption

TBD

§ 5. Codecs Parameter String

TBD

§ Conformance

Conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in the normative parts of this document are to be interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification.

All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [\[RFC2119\]](#)

Examples in this specification are introduced with the words "for example" or are set apart from the normative text with `class="example"`, like this:

EXAMPLE 1

This is an example of an informative example.

Informative notes begin with the word “Note” and are set apart from the normative text with `class="note"`, like this:

Note, this is an informative note.

§ Index

§ Terms defined by this specification

`av02`, in § 2.1

`AV2CodecConfigurationBox`, in
§ 2.2.2

`AV2SampleEntry`, in § 2.2.1

§ Terms defined by reference

[AV2] defines the following terms:

AV2 bitstream
availability of HLS OBUs
Bridge Frame OBU
Closed Loop Key OBU
Closed Random Access
coded extended layer unit
Inter Frame
Intra frame
Key Frame
Layer Configuration Record OBU
Leading frame
Length delimited bitstream format
metadata group OBU syntax
Metadata OBU
metadata short OBU syntax
Multistream
OBU
obu_header_extension_flag
obu_mlayer_id
obu_tlayer_id
obu_type
obu_xlayer_id
Open Loop Key OBU
Open Random Access
ordering of OBUs in a temporal unit
Sequence Header OBU
show_existing_frame
Singlestream
Switch Frame
Switch OBU
Temporal Delimiter OBU
Temporal Unit

[ISOBMFF] defines the following terms:

btrt
cli
colr
ctts
iso6
mdcv
nclx
pasp
VisualSampleEntry

§ References

§ Normative References

[AV2]

AV2 Bitstream & Decoding Process Specification. Standard. URL: <https://av2.aomedia.org/v1.0.0/index.html>

[CMAF]

Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media. Standard. URL: <https://www.iso.org/standard/71975.html>

[ISOBMFF]

Information technology — Coding of audio-visual objects — Part 12: ISO base media file format. Standard. URL: <https://www.iso.org/standard/83102.html>

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997. Best Current Practice. URL: <https://datatracker.ietf.org/doc/html/rfc2119>