ISO/IEC JTC 1/SC 29/WG 03 N1277

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Technologies under Consideration for ISO/IEC 14496-15 Carriage of NAL unit structured video in ISOBMFF

**Status:** Approved

**Date of document:** 2024-05-10

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 26 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N1277**

**July 2024 – Sapporo, JP**

|  |  |
| --- | --- |
| **Title** | **Technologies under Consideration for ISO/IEC 14496-15 Carriage of NAL unit structured video in ISOBMFF** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **24123** |

**Abstract**

The document contains following technologies under consideration for the ISO/IEC 14496-15 Carriage of NAL unit structured video in ISOBMFF.

Table of Contents

[1 Signalling of non-reference layer (from m56045) 3](#_Toc166241690)

[2 APS Roll Recovery (from m54403) 3](#_Toc166241691)

[3 14496-15 (NAL video file formats) errata items (from m55192) 5](#_Toc166241692)

[4 EDRAP and Random access (from m56766) 6](#_Toc166241693)

[5 Generic Codecs Parameter (from m59046) 7](#_Toc166241694)

[5.1 EVC Codecs Parameter 7](#_Toc166241695)

[6 Single-Track LCEVC Enhancement bitstream carriage 9](#_Toc166241696)

[6.1 NALU approach 10](#_Toc166241697)

[6.1.1 AVC/H264 NALU header format 10](#_Toc166241698)

[6.1.2 HEVC/H265 NALU header format 11](#_Toc166241699)

[6.1.3 VVC/H266 NALU header format 11](#_Toc166241700)

[6.1.4 LCEVC NALU header format 12](#_Toc166241701)

[6.1.5 Compatibility of interleaving LCEVC NAL units with AVC/HEVC/VVC NAL units 13](#_Toc166241702)

[6.1.6 Dual Configuration Boxes in Single Track 14](#_Toc166241703)

[6.2 SEI approach 16](#_Toc166241704)

[6.2.1 Carriage of LCEVC NALUs in SEI messages 16](#_Toc166241705)

[6.2.2 Suggested solution for SEI carriage 17](#_Toc166241706)

[6.3 Aggregators approach 17](#_Toc166241707)

[6.3.1 AVC/H264 NALU header format 17](#_Toc166241708)

[7 On codecs string extensions for L-HEVC 19](#_Toc166241709)

[7.1 Required MIME type signaling for L-HEVC 20](#_Toc166241710)

[7.1.1 Problem description 20](#_Toc166241711)

[7.1.2 General 20](#_Toc166241712)

[7.1.3 Extension of the codecs parameter 21](#_Toc166241713)

[7.1.4 Alternatives 22](#_Toc166241714)

[8 Enabling multiple layers in single layer track 23](#_Toc166241718)

[8.1 Multi-layer configuration box in a sample entry 23](#_Toc166241719)

[8.1.1 Clause 4 changes 23](#_Toc166241720)

[8.1.2 New Annex 24](#_Toc166241721)

[8.2 Samples of the single-layer track 25](#_Toc166241722)

[8.3 Auxiliary information for multi-layer bitstream 25](#_Toc166241723)

# Signalling of non-reference layer (from m56045)

Signalling of a sample group for non-reference samples was proposed in m56045. Non-reference samples are never used as a reference samples. A layer non reference flag is signaled in the sample group. It is asserted that knowing information about non-reference pictures sample group to indicate samples which are not used as a reference is useful. For example, this information can be used to drop samples and not decode them if the playback is lagging on a resource-constrained system. Additionally, this information can be used by media aware network element to drop samples if there is bandwidth starvation.

The proposed addition is shown below compared to WG03N0035.

**11.8.12 Non reference sample group**

**11.8.12.1 Definition**

Group Types: 'nref'  
Container: Sample Group Description Box ('sgpd')  
Mandatory: No  
Quantity: Zero or one

This sample group is used to mark non reference samples. A non reference sample is never used as a reference sample for any other sample. An accompanying instance of the SampleGroupDescriptionBox with the same grouping type shall be present. The grouping\_type\_parameter is not defined for the SampleToGroupBox with grouping type 'nref'.

**11.8.12.2 Syntax**

class NonReferenceEntry() extends VisualSampleGroupEntry ('nref')  
{  
 bit(7) reserved = 0;  
 unsigned int(1) layer\_non\_ref\_only\_flag;  
}

**11.8.12.3 Semantics**

layer\_non\_ref\_only\_flag equal to 1 specifies that for this sample a current picture is never used as a reference picture for any picture within its own layer and may or may not be used a reference picture for inter-layer prediction for pictures of a different layer. layer\_non\_ref\_only\_flag equal to 0 specifies that the current sample is never used as a reference sample.

# APS Roll Recovery (from [m54403](http://wg11.sc29.org/doc_end_user/current_document.php?id=75250&id_meeting=183))

**11.8.X APS Roll Recovery**

**11.8.X.1 Definition**

The 'apsr' sample group indicates that a VVC sync sample from a VVC track requires additional gathering of prefix and suffix APS NAL units from preceding samples and rewriting these as APS prefix NAL units to be a self-contained sync sample. This avoids having to duplicate APS information at each sync sample in the track.

A sync sample not belonging to an 'apsr' sample group does not require any additional processing to gather the dependent APS (i.e., all APS required are in the sample entry or in the sample). The 'apsr' sample group shall only be present in a VVC track or a VVC base track with no dependencies to a VVC non-VCL track; it shall not be present in VVC subpicture tracks or VVC non-VCL tracks.

The aps\_roll\_count shall be such that all samples described by the roll operation are available in the track, track fragment or ISOBMFF segment (as indicated by the roll\_type) being processed.

A sample associated to an 'apsr' sample group description entry shall be a sync sample or a sample with SAP type 3 or 4 (potentially associated to a 'roll' sample group description entry).

A sample associated to an 'apsr' sample group description entry with aps\_roll\_count not equal to 0 shall be considered as a SAP type 4.

**11.8.X.2 Syntax**

aligned(8) class APSRollRecoveryEntry () extends VisualSampleGroupEntry('apsr')  
{  
 unsigned int(2) roll\_type;  
 unsigned int(2) roll\_mode;  
 unsigned int(4) reserved=0;  
 if (roll\_type == 0) {  
 unsigned int(16) aps\_roll\_count;  
 }  
}

**11.8.X.3 Semantics**

roll\_type indicates the pre-roll distance for APS NAL units when producing a sync sample with sample number N belonging to this group. The following values are defined:

0: APS NAL units are gathered starting from the sample located aps\_roll\_count samples before the sample belonging to the group

1: APS NAL units are gathered starting from the first sample of the track or track fragment

2: APS NAL units are gathered starting from the first sample of the associated ISOBMFF segment

3: reserved

roll\_mode indicates which samples in the identified roll sample window should be analyzed for APS NAL unit gathering. The following values are defined:

0: the required APS NAL units may be present in any sample

1: the required APS NAL units are only present in the first sample

2: the required APS NAL units are only present in samples that are either sync samples or samples marked as 'rap '

3: the required APS NAL units are only present in samples of the current track fragment

aps\_roll\_count indicates the number of samples to rewind for APS gathering; a value of 0 indicates that the associated sample contains all APS NAL units for its processing.

# 14496-15 (NAL video file formats) errata items (from [m55192](https://wg11.sc29.org/doc_end_user/current_document.php?id=76249&id_meeting=184))

[Editor’s note] The following changes have been captured in the current TuC document in lack of a Defect Report on 14496-15 issued at MPEG #132. This paragraph is thus meant to be moved in a more appropriate document at the next meeting.

1. Search and replace "RBSP payload" (1 instance) with "NAL unit payload", because in the context where the phase is used, the start code emulation bytes that are part of "NAL unit payload" but not part of "RBSP payload" should be considered.
2. Search and replace "byte stream payload" (2 instances) with "NAL unit payload", for similar reason as above.
3. Change, in clause 8.4.1.1.1, the following:

"When the sample entry name is 'hvc1', the default and mandatory value of array\_completeness is 1 for arrays of all types of parameter sets, and 0 for all other arrays. When the sample entry name is 'hev1', the default value of array\_completeness is 0 for all arrays."

to

"When the sample entry name is 'hvc1', the ~~default and mandatory~~ value of array\_completeness ~~is~~ shall be equal to 1 for the arrays of all types of parameter sets~~, and 0 for all other arrays~~. When the sample entry name is 'hev1', the ~~default~~ value of array\_completeness shall be equal to 1 for the arrays of all types of parameter sets ~~is 0 for all arrays~~."

1. Change, in clause 9.5.3.1.1, the following:

"When the sample entry name is 'lhv1', the default and mandatory value of array\_completeness is 1 for arrays of all types of parameter sets, and 0 for all other arrays. When the sample entry name is 'lhe1', the default value of array\_completeness is 0 for all arrays."

to

"When the sample entry name is 'lhv1', the ~~default and mandatory~~ value of array\_completeness ~~is~~ shall be equal to 1 for the arrays of all types of parameter sets~~, and 0 for all other arrays~~. When the sample entry name is 'lhe1', the ~~default~~ value of array\_completeness shall be equal to 1 for the arrays of all types of parameter sets ~~is 0 for all arrays~~."

1. Change, in clause 8.3.3.1.3, the following:

array\_completeness when equal to 1 indicates that all NAL units of the given type are in the following array and none are in the stream; when equal to 0 indicates that additional NAL units of the indicated type may be in the stream; the default and permitted values are constrained by the sample entry name.

to the following:

array\_completeness when equal to 1 indicates that all NAL units of the given type are in the following array and none are in the stream; when equal to 0 indicates that additional NAL units of the indicated type may be in the stream; the ~~default and~~ permitted values are constrained by the sample entry name.

1. Change, in clause 8.3.3.1.1, the following:

The level indication general\_level\_idc shall indicate a level of capability equal to or greater than the highest level indicated for the highest tier in all the parameter sets.

to the following:

The level indication general\_level\_idc shall indicate a level of capability equal to or greater than the highest level ~~indicated for the highest tier~~ in all the parameter sets.

Because the highest level of the highest tier could be lower than the highest level of the lowest tier, while level determines spatial resolution etc., which is of vital importance for determining the required decoding capability.

# EDRAP and Random access (from m56766)

The ‘edrp’ sample group from ISO/IEC 14496-12 TuC proposes to signal additional random access point in bitstream that could outperform DRAP in coding efficiency thank to the possibility to refer to previous EDRAP for inter prediction as represented in the figure below:

Logo

Description automatically generated

Figure 1: Inter prediction reference relationship among the RAP pictures in the EDRAP case.

In current version of the Part-12 TuC, no information is provided in case an EDRAP depends on APS NAL units present in prior samples.

For example, the figure below is an example of such VVC bitstream.

A picture containing logo

Description automatically generated

When Random Accessing from the EDRAP6 sample, the ‘edrp’ sample group indicates that only IDR0 and EDRAP2 are needed for reference to decode the bitstream (yellow or grey arrows). The ‘edrp’ sample group is silent about possible references to APS NAL units in previous samples (red arrows). For example, EDRAP6 may reference APSs that are not part of the sample used as references in the EDRAP as per section 11.3.4 of ISO/IEC 14496-15 (recalled hereafter for convenience):

*When the sample entry name is 'vvc1'and the track does not have a track reference of type 'vvcN', the following applies:*

* *If the sample is a sync sample, all APSs needed for decoding that sample shall be included either in the sample entry or in the sample itself.*
* ***Otherwise (the sample is not a sync sample), all APSs needed for decoding the sample shall be included either in the sample entry or in any of the samples since the previous sync sample to the sample itself, inclusive.***

A screenshot of a video game

Description automatically generated

From the discussion at MPEG #134 (<http://mpegx.int-evry.fr/software/MPEG/Systems/FileFormat/NALuFF/-/issues/133>), the following observation has been made:

* All non-VCL NAL units needed to decode the EDRAP should be referenced from the EDRAP sample group.

# Generic Codecs Parameter (from m59046)

## EVC Codecs Parameter

DASH and other applications require defined values for the Codecs parameter specified in IETF RFC 6381 for ISO BMFF Media tracks. The 'codecs' parameter string for the EVC codec is defined as follows:

<sample entry 4CC>.<key1><value1>.<key2><value2>.….<keyN><valueN>

Keys are defined as 4CCs. An set of keys and the associated value pairs are defined in **Table *1***. Additional keys may be specified as 4CCs. Preferably, keys are aligned with ISO/IEC 23091-2. [Ed. (MH/KB): If the keys derived from ISO/IEC 23091-2 are kept in this document, a normative reference to ISO/IEC 23091-2 has to be added.]

If a specific key is not provided, then the value takes the default value specified in the table, or the value is unknown if no default is specified.

NOTE: The parameters from 'vbit' onwards in Table 1 are not EVC specific and are applicable to any video codec. It is foreseen that these values are moved to ISO/IEC 14496-12 in future versions and may then be applicable to other video codecs as well. [Ed. (MH): FI\_106-207 resolution: Keys from vbit onwards moved to ISO/IEC 14496-12.]

**Table 1 - Definition of Set of Keys and Values defined for EVC**

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Key Definition** | **Value** | **Default Value** |
| 'vprf' | Defines the video profile | profile\_idc | 1 |
| 'vlev' | Defines the video level | level\_idc | 51 |
| 'vtoh' | Defines the toolset | toolset\_idc\_h in hex decimal | x1FFFFF |
| 'vtol' | Defines the toolset | toolset\_idc\_l in hex decimal | x000000 |
| 'vbit' | Defines the video bit depth for luma and chroma | Value is a 2 digit decimal . The first digit is the luma bit depth minus 8. The second digit is the chroma bit depth minus 8 | unknown |
| 'vcss' | Defines the chroma subsampling | The subsampling scheme is expressed as a three part ratio J:a:b (e.g. 4:2:2) that describes the number of luminance and chrominance samples in a conceptual region that is J pixels wide, and 2 pixels high. The parts are (in their respective order):   * J: horizontal sampling reference (width of the conceptual region). Usually, 4. * a: number of chrominance samples (Cr, Cb) in the first row of J pixels. * b: number of changes of chrominance samples (Cr, Cb) between first and second row of J pixels.   Then the value is as follows. The first digit is J, the second digit is a and the third digit is b. | 420 |
| 'vcpr' | Defines colour primaries (ColourPrimaries) as defined in ISO/IEC 23091-2. | Defines colour primaries (ColourPrimaries) values as defined in ISO/IEC23091-2. Value is a 2 digit decimal with a possible preceding 0. | 01 |
| 'vtrc' | Defines transfer characteristics (TransferCharacteristics) as defined in ISO/IEC23091-2. | Defines transfer characteristics (TransferCharacteristics) values as defined in ISO/IEC23091-2. Value is a 2 digit decimal with a possible preceding 0. | 01 |
| 'vmac' | Defines matrix coefficients (MatrixCoefficients) as defined in ISO/IEC23091-2. | Defines matrix coefficients (MatrixCoefficients) values as defined in ISO/IEC23091-2. Value is a 2 digit decimal with a possible preceding 0. | 01 |
| 'vfrf' | Defines VideoFullRangeFlag as defined in ISO/IEC23091-2. | Defines VideoFullRangeFlag as defined in ISO/IEC23091-2. Value is a 1 digit decimal restricted to values 1 or 0. | 0 |
| 'vfpq' | Defines video frame packing type (VideoFramePackingType) as defined in ISO/IEC23091-2 together with the QuincunxSamplingFlag. | Defines video frame packing type (VideoFramePackingType) values as defined in ISO/IEC23091-2. Value is a 2 digit decimal with the first one being the QuincunxSamplingFlag and the second digit the VideoFramePackingType. | If not present, then no frame packing is used. |
| 'vpci' | Defines Packed content interpretation type (PackedContentInterpretationType) as defined in ISO/IEC23091-2. | Packed content interpretation type (PackedContentInterpretationType) as defined in ISO/IEC23091-2. Value is a 1 digit. | If not present, then no packed content is used. |
| 'vsar' | Defines Sample aspect ratio indicator (SampleAspectRatio) as defined in ISO/IEC23091-2. | Defines Sample aspect ratio indicator (SampleAspectRatio) values as defined in ISO/IEC23091-2. Value is a 2 digit decimal with a possible preceding 0. | 01 |

For example, codecs="evc1.vprf3.vlev51.vto1FFFFF.vtoh000000.vbit20.vcss420.‌vcpr09.vtrc16.vmac09.vsar01" represents EVC Main Profile, level 5.1, with 4:2:0 chroma subsampling co-located with (0, 0) luma sample, a restricted tool set, ITU-R BT.2100 color primaries, ITU-R BT.2100 PQ transfer characteristics, ITU-R BT.2100 YCbCr color matrix and sample aspect ratio 1:1. [Ed. (KB): If the keys derived from ISO/IEC 23091-2 are kept in this document, these ITU-Rs should be added to the Bibiography.]

All keys in **Table *1*** must be recognized, if the 'evc1' sample entry is recognized. If a key is not recognized, the key value pair is ignored.

# Single-Track LCEVC Enhancement bitstream carriage

This section documents three approaches for carriage of a Base bitstream (e.g. AVC, HEVC, EVC, VVC) and an LCEVC Enhancement bitstream in a “Single-Track”, that is without the need to use two separate PIDs or Tracks, but rather carrying the base NALUs and the LCEVC NALUs as if they were a single bitstream.

Three possible solutions to carry a complete set composed by Base and LCEVC Enhancement bitstreams are:

(1) Interleaved NALUs approach: insert LCEVC NALUs as interleaved NALUs within the Base NALUs.

(2) SEI approach: insert LCEVC NALUs as SEI messages in the Base NALUs.

(3) Aggregators to aggregate the Base bitstream with the LCEVC bitstream.

The following sections describe the three possible Single-Track solutions making explicit reference to four existing MPEG Video Coding specification:

• ISO/IEC 14496-10 Advanced Video Coding (AVC/H264)

• ISO/IEC 23008-2 High Efficiency Video Coding (HEVC/H265)

• ISO/IEC 23090-3 Versatile Video Coding (VVC/H266)

• ISO/IEC 23094-1 Essential Video Coding (EVC)

## NALU approach

LCEVC encoded data units are Network Abstraction Layer units (NALU) as defined in ISO/IEC 23094-2, Sec. 7.3.2.

Each of the Base video coding standards under considerations here (AVC, HEVC, VVC) defines its own NALU syntax.

The format of the LCEVC NALU, in fact, allows for their unambiguous detection even when parsed according to the NALU syntax of the AVC, HEVC or VVC Base bitstream. This property allows for an “interleaved” single stream base plus enhancement, which means a single stream where LCEVC NALUs are inserted among base NALUs, within the same NALU sequence.

In the resulting interleaved single stream, each Access Unit, defined as the set of NALUS that result in each decoded picture, will contain both the LCEVC NALUs (an LCEVC Access Unit contains only one LCEVC NAL unit) and the Base NALUs relevant for the specific Access Unit.



Figure 3: Diagram of Interleaved NALU “single track” for LCEVC.

### AVC/H264 NALU header format

The AVC NALU header is defined in ISO/IEC 14496-10, Sec. 7.3.1, with the following syntax:

| ***Syntax*** | **Descriptor** |
| --- | --- |
| nal\_unit\_header( ) { |  |
| forbidden\_zero\_bit | u(1) |
| nal\_ref\_idc | u(2) |
| nal\_unit\_type | u(5) |
| } |  |

Table 6 – AVC NALU header syntax

The NALU type values and semantics for AVC are specified in Table 7-1 of the specification (IS 14496-10). Table 7 summarizes the usage of the AVC NALU types. Since the AVC NALU type is a field of 5 bits, the possible values are from 0 to 31.

|  |  |  |  |
| --- | --- | --- | --- |
| nal\_unit\_type | Name of nal\_unit\_type | Content of NAL unit and RBSP syntax structure | NAL unit type class |
| 0 - 5 | (…) | (…) | VCL |
| 6 - 20 | (…) | (…) | Non VCL |
| 21 - 23 | RSV | Reserved | Non VCL |
| 24 - 31 | UNSPEC | Unspecified | Non VCL |

Table 7 – AVC NALU types

### HEVC/H265 NALU header format

The HEVC NALU header is defined in ISO/IEC 23008-2, Sec. 7.3.1.2, with the following syntax:

| ***Syntax*** | **Descriptor** |
| --- | --- |
| nal\_unit\_header( ) { |  |
| forbidden\_zero\_bit | f(1) |
| nal\_unit\_type | u(6) |
| nuh\_layer\_id | u(6) |
| nuh\_temporal\_id\_plus1 | u(3) |
| } |  |

Table 8 – HEVC NALU header syntax

The NALU type values and semantics for HEVC are specified in Table 7-1 of the specification (IS 23008-2). Table 9 summarizes the usage of the HEVC NALU types. Since the HEVC NALU type is a field of 6 bits, the possible values are from 0 to 63.

|  |  |  |  |
| --- | --- | --- | --- |
| nal\_unit\_type | Name of nal\_unit\_type | Content of NAL unit and RBSP syntax structure | NAL unit type class |
| 0 - 21 | (…) | (…) | VCL |
| 22 - 31 | RSV | Reserved | VCL |
| 32 - 40 | (…) | (…) | Non VCL |
| 41 - 47 | RSV | Reserved | Non VCL |
| 48 - 63 | UNSPEC | Unspecified | Non VCL |

Table 9 – HEVC NALU types

### VVC/H266 NALU header format

The VVC NALU header is defined in ISO/IEC 23090-3, Sec. 7.3.1.2, with the following syntax:

| ***Syntax*** | **Descriptor** |
| --- | --- |
| nal\_unit\_header( ) { |  |
| forbidden\_zero\_bit | f(1) |
| nuh\_reserved\_zero\_bit | u(1) |
| nuh\_layer\_id | u(6) |
| nal\_unit\_type | u(5) |
| nuh\_temporal\_id\_plus1 | u(3) |
| } |  |

Table 10 – VVC NAL unit header syntax

The NALU type values and semantics for VVC are specified in Table 5 of the specification (IS 23090-3). Table 11 summarizes the usage of the VVC NALU types. Since the VVC NALU type is a field of 5 bits, the possible values are from 0 to 31.

|  |  |  |  |
| --- | --- | --- | --- |
| nal\_unit\_type | Name of nal\_unit\_type | Content of NAL unit and RBSP syntax structure | NAL unit type class |
| 0 - 11 | (…) | (…) | VCL |
| 12 - 25 | (…) | (…) | Non VCL |
| 26 - 27 | RSV | Reserved | Non VCL |
| 28 - 31 | UNSPEC | Unspecified | Non VCL |

Table 11 – VVC NALU types

### LCEVC NALU header format

The LCEVC NALU header is defined in ISO/IEC 23094-2, Sec. 7.3.2, with the following syntax:

| ***Syntax*** | **Descriptor** |
| --- | --- |
| nal\_unit\_header( ) { |  |
| forbidden\_zero\_bit | u(1) |
| forbidden\_one\_bit | u(1) |
| nal\_unit\_type | u(5) |
| reserved\_flag | u(9) |
| } |  |

Table 12 - LCEVC NAL unit header

The NALU type values and semantics for LCEVC are specified in Table 17 of the specification (IS 23094-2). Table 13 summarizes the usage of the LCEVC NALU types. Since the LCEVC NALU type is a field of 5 bits, the possible values are from 0 to 31.

|  |  |  |  |
| --- | --- | --- | --- |
| nal\_unit\_type | Name of nal\_unit\_type | Content of NAL unit and RBSP syntax structure | NAL unit type class |
| 0 - 27 | UNSPEC | Unspecified | Non VCL |
| 28 - 29 | (…) | (…) | VCL |
| 30 | RSV | Reserved | Non VCL |
| 31 | UNSPEC | Unspecified | Non VCL |

Table 13 – LCEVC NALU types

The two NALU type values used to identify VCL NALUs are:

28 = 0x1C = 0b1.1100 (LCEVC NALU type 28Non IDR)

29 = 0x1D = 0b1.1101 (LCEVC NALU type 29IDR)

The two NALU header bytes for the two VCL NALU types of LCEVC are as follows:

0111.1001:1111.1111 (LCEVC NALU type 28 Non-IDR)

0111.1011:1111.1111 (LCEVC NALU type 29 IDR)

### Compatibility of interleaving LCEVC NAL units with AVC/HEVC/VVC NAL units

The following table summarizes the position and semantic of the main fields of the NALU headers of the four MPEG specifications: AVC, HEVC, VVC, LCEVC:

* in yellow the NALU Type field
* in blue the Layer ID field
* in green the Temporal ID field

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | 14 | 15 |
| AVC | 0 |  |  | NALU Type | | | | |  |  |  |  |  |  | |  |  |
| HEVC | 0 | NALU Type | | | | | | Layer ID (6) | | | | | | | Temp ID (3) | | | |
| VVC | 0 | 0 | Layer ID (6) | | | | | | NALU type | | | | | | Temp ID (3) | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | 14 | 15 |
| LCEVC | 0 | 1 | 1 | 1 | 1 | 0/1 | 0/1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 |
| LCEVC 28 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 |
| LCEVC 29 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 |

The following table reports how the base parsers for AVC, HEVC, VVC would interpret the two bytes of NALU headers for the significant LCEVC NALUs.

|  |  |  |  |
| --- | --- | --- | --- |
|  | LCEVC 28|29 | Layer ID | Temp ID |
| AVC parser | AVC 25|27 |  |  |
| HEVC parser | HEVC 60|61 | 63 | 7 |
| VVC parser | VVC 31|31 | 57|59 | 7 |

For the AVC parser, the LCEVC NALU types (interpreted as 25 or 27) fall in the Unspecified range from 24 to 31.

For the HEVC parser, the LCEVC NALU types (interpreted as 60 or 61) fall in the Unspecified range from 48 to 63.

For the VVC parser, the LCEVC NALU types (interpreted as 31) fall in the Unspecified range from 28 to 31.

### Dual Configuration Boxes in Single Track

One open issue in the “NALU approach” proposal, where Base codec NALUs and LCEVC Enhancement codec NALUs are interleaved, is how to carry two different DecoderConfigurationBoxes in a Single Track, preserving backward compatibility with existing MP4 players.

To solve the above issue, two options are proposed for a backward compatible solution to carry an LCEVC bitstream and a base bitstream in a single-track carriage.

#### Option 1 – Use of SampleDescriptionBox

In this option, we would insert the LCEVC Decoder Configuration associated to the Base Decoder configuration using the same SampleDescriptionBox containing the Base Decoder configuration.

The SampleDescriptionBox is described in ISO/IEC 14496-12 (ISO BMFF) in Section 8.5.2.2 and reported below for convenience:

aligned(8)  
class SampleDescriptionBox() extends FullBox ('stsd', version, 0) {  
 int i;  
 unsigned int(32) entry\_count;  
 for (i = 1 ; i <= entry\_count ; i++) {  
 SampleEntry(); // an instance of a class derived from SampleEntry  
 }  
}

The SampleDescriptionBox allows the allocation of one or more SampleEntry, depending on the field entry\_count. Thus, the LCEVC Sample Entry can be inserted after the Base Sample Entry in the same SampleDescriptionBox.

NOTE:

Since there cannot be two samples (Base and LCEVC) with the same time stamp in a single track, the LCEVC Sample should be packed in the corresponding Base Sample, or the LCEVC Sample should be carried in Sample Auxiliary Information, as specified in clauses 8.7.8 and 8.7.9 of 14496-12.

The approach using Sample Auxiliary Information might have limitations with respect to ISOBMFF tools, such as differential encryption of Base Samples and LCEVC SAI Samples.

NOTE:

We should have specific signaling in the Sample Entry to signal that the stream contains unspecified NALUs (with respect to the Base Sample Entry) that then are interpreted no longer as unspecified, but as LCEVC NALUs according to the MPEG4ExtensionDescriptorsBox.

#### Option 2 – Use of MPEG4ExtensionDescriptorsBox

With this option, we would insert the LCEVC Decoder Configuration using the MPEG4ExtensionDescriptorsBox present in the sample entry of the Base Decoder (in the example below, we show the one for AVC, others are reported in the Annex):

class AVCSampleEntry() extends VisualSampleEntry ('avc1' or 'avc3') {  
 AVCConfigurationBox config;  
 MPEG4ExtensionDescriptorsBox (); // optional  
}

class MPEG4ExtensionDescriptorsBox extends Box('m4ds') {  
 Descriptor Descr[0 .. 255];  
}

The MPEG4ExtensionDescriptorsBox is described in ISO/IEC 14496-15 (NALU FF) in Section 5.4.2.1.2 and reported below for convenience:

***MPEG4ExtensionDescriptorsBox***

*Descr is a descriptor that should be placed in the ElementaryStreamDescriptor when this stream is used in an MPEG-4 systems context. This does not include SLConfigDescriptor or DecoderConfigDescriptor, but includes the other descriptors in order to be placed after the SLConfigDescriptor.*

*Note that DecoderConfigDescriptor (Decoder Configuration Descriptor) is defined in 14496-1, clause 7.2.6.6 and SLConfigDescriptor (Sync Layer Configuration Descriptor) is defined in 14496-1, clause 7.3.2.3.*

*Since the specific ConfigurationBox for any Base codec of interest (AVCConfigurationBox, HEVCConfigurationBox, EVCConfigurationBox, VVCConfigurationBox) is not derived from the DecoderConfigDescriptor, it seems consistent with the existing 14496-15 specification, clause 5.4.2.1.3, to include in the MPEG4ExtensionDescriptorsBox an LCEVCConfigurationBox.*

NOTE:

Since there cannot be two samples (Base and LCEVC) with the same time stamp in a single track, the LCEVC Sample should be packed in the corresponding Base Sample, or the LCEVC Sample should be carried in Sample Auxiliary Information, as specified in clauses 8.7.8 and 8.7.9 of 14496-12.

The approach using Sample Auxiliary Information might have limitations with respect to ISOBMFF tools, such as differential encryption of Base Samples and LCEVC SAI Samples.

NOTE:

We should have specific signaling in the Sample Entry to signal that the stream contains unspecified NALUs (with respect to the Base Sample Entry) that then are interpreted no longer as unspecified, but as LCEVC NALUs according to the MPEG4ExtensionDescriptorsBox.

#### Specification amendments

This section reports the required amendments to the MPEG4 File Format specification, 14496-12, to support the presence of two DecoderConfigDescriptors in a single track, one for the Base codec, one for the LCEVC Enhancement codec.

##### Option 1 – Use of SampleDescriptionBox

In principle, in 14496-12, clause 8.5.2.2, no amendment would be strictly needed, since the current specification allows the insertion of a first Sample Entry for the Base and a second Sample Entry for the LCEVC Enhancement.

For clarity and completeness, the specific case of a second Sample Entry for LCEVC can be described in the text or a note to the specification.

This solution also gives the advantage of possibly associating more than one LCEVC bitstream to the same Base bitstream, without the need for duplication.

##### Option 2 – Use of MPEG4ExtensionDescriptorsBox

In 14496-15, clause 5.4.2.1.3, at the end of the Descr semantics, add the highlighted text:

*Descr is a descriptor that should be placed in the ElementaryStreamDescriptor when this stream is used in an MPEG-4 systems context. This does not include SLConfigDescriptor or DecoderConfigDescriptor, but includes the other descriptors in order to be placed after the SLConfigDescriptor. In particular, the descriptor for an LCEVC decoder can be present in the Descr field: in this case, one or more LCEVCConfigurationBox shall be considered as the descriptor(s) for one or more LCEVC decoder(s) associated as enhancement to the Base decoder identified by the ConfigurationBox preceding the MPEG4ExtensionDescriptorsBox.*

This solution also gives the advantage of possibly associating more than one LCEVC bitstream to the same Base bitstream, without the need for duplication.

## SEI approach

LCEVC encoded data units are Network Abstraction Layer (NAL) units as defined in ISO/IEC 23094-2, Sec. 7.3.2.

All the MPEG base video coding standards considered (e.g., AVC, HEVC, VVC) provide metadata messages that can be used for the carriage of LCEVC. AVC, HEVC and VVC, employ NAL units as basic data units, and additionally the type of NAL unit identified as Supplemental Enhancement Information (SEI) that can be used to embed the LCEVC NAL unit stream.



Figure 4: Diagram of Embedded SEI “single track” for LCEVC.

### Carriage of LCEVC NALUs in SEI messages

When the base encoding for LCEVC is an MPEG standard, the elementary stream is a NALU stream. In this case, the encapsulation of LCEVC Access Units as metadata is implemented using the SEI messages specific for each Base codec. AVC, HEVC and VVC have each a different NALU format (i.e., with different NALU headers) breakdown of NALU types and payloads. However, all of them comprise SEI messages, identified with a nal\_unit\_type field as in the following table, where RBSP stands for raw byte sequence payload (see Table 1 below).

|  |  |  |
| --- | --- | --- |
| MPEG Standard | SEI nal\_unit\_type | Corresponding NAL unit payload |
| AVC | 6 | sei\_rbsp() |
| HEVC | 39\* | sei\_rbsp() |
| VVC | 23\* | sei\_rbsp() |
| \* prefix SEI | | |

Table 1 - SEI NALU type

### Suggested solution for SEI carriage

The proposed solution for using SEI encapsulation of an LCEVC bitstream consists in defining a new SEI message for LCEVC and referencing it as a new Payload Type in each of the base layer video coding specifications.

In AVC (IS 14496-10), the SEI message payloadType is a field of 8 bits (with an escape mechanism). The Payload Types are specified in Section D.1.1, with values ranging from 0 to 56, and from 137 to 201, with the other values up to 255 allocated as reserved\_sei\_message.

In HEVC (IS 23008-2), as well, the SEI message payloadType is a field of 8 bits (with an escape mechanism). The Payload Types are specified in Section D.2.1, with values ranging from 0 to 56, and from 128 to 201, with other values up to 255 allocated as reserved\_sei\_message.

In VVC (IS 23090-3), as well, the SEI message payloadType is a field of 8 bits (with an escape mechanism). The Payload Types are specified in Section D.2.1, with values ranging from 0 to 45, and from 129 to 204, with other values up to 255 allocated as reserved\_sei\_message.

The suggested value of payloadType to be allocated to LCEVC in the three specifications (AVC, HEVC, VVC) is payloadType value 57, that falls in the “reserved range” of all the three spec.

## Aggregators approach

In this proposal we provide a ISOBMFF based codec-agnostic solution for the carriage of LCEVC bitstream in a single-track.

### AVC/H264 NALU header format

The proposed changes, with respect to the current “Dual Track” carriage of LCEVC, are highlighted with track changes or yellow text.

The clause numbering refers to the numbering currently used in IS 14496-15, Clause 13.

**13 LCEVC elementary streams and sample definitions**

**13.1 Overview**

…

LCEVC elementary streams carry enhancement to a "base" codec such as the ones listed above. A LCEVC elementary stream and a “base” codec elementary stream may be present in the same track. When, a LCEVC elementary stream is in its own track, it makes a reference to a "base" codec elementary stream in a separate track, so that the LCEVC stream can be decoded in conjunction with the "base" stream, while the "base" stream can be decoded independently of the LCEVC stream.

An aggregator structure is specified to allow LCEVC elementary stream and a “base” codec elementary stream to be present in the same track. The aggregator structure enables grouping of NAL units from the base codec elementary stream into aggregated data units.

This clause defines the carriage of LCEVC elementary streams in the ISO base media file format as defined in this specification.

…

**13.4 Derivation from ISO base media file format**

**13.4.1 LCEVC video stream definition: sample entry name and format**

**13.4.1.1 Definition**

…

**13.4.1.2 Syntax**

…

**13.4.1.3 Semantics**

…

**13.4.2 LCEVC mixed sample entry**

**13.4.2.1 Definition**

Sample Entry and Box Types: 'lvms'

Container: Sample Description Box ('stsd')

Mandatory: The 'lvms' sample entry is mandatory

Quantity: One or more sample entries may be present

An LCEVC mixed sample entry shall contain a LCEVCConfigurationBox and a BaseConfigurationBox, as defined below. The BaseConfigurationBox contains the sample entry type and the decoder configuration box of the base stream (e.g. AVCConfigurationBox, HEVCConfigurationBox).

An optional BitRateBox may be present in the LCEVC mixed sample entry to signal the bit rate information of the LCEVC and the the base stream. Extension descriptors that should be inserted into the Elementary Stream Descriptor, when used in MPEG-4, may also be present.

The sample entry name 'lvms' specifies that the track to which this sample entry applies contains both a LCEVC stream and the base stream.

Base aggregators, as specified in subclause A.10, shall be used for aggregating the base stream in 'lvms' tracks. The order of all coded data included in a base aggregator is exactly the decoding order as if the coded data were present in a sample not containing aggregators or LCEVC NAL units.

If the sample of an 'lvms' track contains unspecified NAL unit types as defined in ISO/IEC 23094-2, the NAL units or NAL-unit-like structures having unspecified NAL unit types shall be discarded from the sample before providing the sample to the LCEVC decoder.

**13.4.2.2 Syntax**

class LCEVCMixedSampleEntry() extends VisualSampleEntry('lvms'){  
 LCEVCConfigurationBox config1;  
 BaseConfigurationBox config2;  
 MPEG4ExtensionDescriptorsBox(); // optional  
}

class BaseConfigurationBox() extends Box('blcf'){  
 unsigned int(32) base\_4cc;  
 Box config; // E.g., AVCConfigurationBox  
 Box other\_boxes[]; // optional boxes that are allowed for base\_4cc  
}

**13.4.2.3 Semantics**

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\014LCEVC Coding" being recommended (\014 is 10, the length of the string in bytes).

BaseConfigurationBox contains the decoder configuration box of the base stream (e.g. AVCConfigurationBox, HEVCConfigurationBox).

base\_4cc is the sample entry type that the base stream conforms to.

config is the decoder configuration box of the base stream (e.g. AVCConfigurationBox, HEVCConfigurationBox).

**13.4.3 LCEVC track structure**

…

When the base track is coded using EVC, the base track shall be constructed according to clause 12.

A LCEVC mixed track is a track containing both the external base layer stream and the LCEVC enhancement stream, forming a representation of a complete set of encoded information.

The picture dimensions of the base stream and the LCEVC stream, width and height in Luminance samples, are specified by the corresponding relevant DecoderConfigurationRecord(s).

**13.4.4 Parameter sets**

…

**13.4.5 'sync' sample**

…

# On codecs string extensions for L-HEVC

During MPEG #145 the contribution [m65896](https://dms.mpeg.expert/doc_end_user/current_document.php?id=90795) raised the question about the limitations of the currently defined codecs string MIME type in combination to layered HEVC coding carriage in mp4.

One of the possibilities to carry layered HEVC (L-HEVC) video in mp4 is by using the 'hvc1' or 'hev1' sample entry type as specified in clause 9 of ISO/IEC 14496-15 in a backwards compatible manner. Similar concept is also [utilized by Apple](https://developer.apple.com/av-foundation/HEVC-Video-with-Alpha-Interoperability-Profile.pdf) for the carriage of HEVC with alpha in the HEVC Video with Alpha Interoperability Profile where the sample entry type 'hvc1' is used. However, when constructing the MIME types 'codecs' parameter, according to Annex E of ISO/IEC 14496-15, the MIME type specification only includes signaling for profiles, tiers and levels from SPS NAL units of each particular layer. This signaling alone, while very useful, does not expose other important information such as the types of auxiliary information that would allow us to signal the presence of alpha in the track. In addition to that, some of that signaling is exposed as an additional MIME parameter which turns out to be problematic.

This section seeks for a solution to this problem and investigates an extension method for the Annex E to define additional signaling for the codecs string when multiple layers are present as for example in the 'hvc1' or 'hev1' L-HEVC track.

This topic was discussed in the dedicated AhG call before MPEG #146 and an additional contribution [m67864](https://dms.mpeg.expert/doc_end_user/current_document.php?id=93043) was proposed at MPEG #146. That contribution refined the initial proposal and proposed to generalize the signaling in ISOBMFF as it is not only applicable to L-HEVC but can also be seen as codec independent. However, no consensus on the contribution could be reached at MPEG #146 and the topic will be continued to be studied in the AhG call and at MPEG #147. In this TuC we update the below text based on [m67864](https://dms.mpeg.expert/doc_end_user/current_document.php?id=93043). The dedicated ISOBMFF TuC (MDS23807\_WG03\_N01197) is created to capture the codec-agnostic aspects of the contribution.

At MPEG #147, it was agreed to add specific processes into the ISOBMFF amendment Working Draft to specify additional rendering capabilities signaling. This includes defining a new rendering MIME type parameter to specify rendering attributes in a codec-agnostic manner. The rendering parameter string will encompass key attributes such as image type (e.g., regular video, alpha, or depth), color space, and subsampling, all based on CICP signaling, with guidance on retrieving this information from existing box definitions. Additionally, another approach will be outlined, allowing these rendering parameters to be integrated within the codecs string using the 'also' 4CC prefix.

## Required MIME type signaling for L-HEVC

### Problem description

The primary objective is to establish a method for signaling the presence of auxiliary video (such as HEVC with alpha) using the codecs string. This signal should also include additional information, as detailed later in this document. The solution should enable [HLS](https://datatracker.ietf.org/doc/html/rfc8216/) (or MPEG-DASH) players to recognize auxiliary stream support from the HLS multivariant playlist without needing to access the initialization segments.

Moreover, the solution should enable [MSE](https://www.w3.org/TR/media-source-2/)-based players to determine browser support for this content. Notably, at least one W3C API accepts a MIME type with no extra MIME parameters except for codecs. As per the W3C's [Media Capabilities API](https://www.w3.org/TR/media-capabilities/#http):

If the MIME type does not imply a codec, the string MUST also have one and only one parameter that is named codecs with a value describing a single media codec.

There is also uncertainty about whether adding unknown parameters to a request might cause errors in some clients, even if they might theoretically support such content.

It is important to note that the same issue also applies to the lhevcptl parameter from Annex E.4. It's unclear why this information, which refers to codec signaling and originates from parameter sets, is separated from the codecs string, and placed in a new MIME parameter.

### General

It is proposed to add additional signaling that can be used when NAL units with nuh\_layer\_id > 0 are present in the HEVC bitstream of a track. The proposed extension should ensure backwards compatibility and should be able to signal the following information for each layer:

* nuh\_layer\_id that can be signaled as a decimal number.
* type of each layer, such as Texture, Alpha, or Depth.
* bit-depth of each layer.
* chroma subsampling of each layer signaled as chroma\_format\_idc
* Colour properties
* profile information
* dependency on other layers

This new signaling should make it possible to expose the information from the parameter sets from several layers to the codecs string and provide necessary signaling to the application layer.

### Extension of the codecs parameter

Given the limitations described in section 2.1.1 one possible solution would be to define a new 4CC that can be used within the codecs string and signal the information from section 2.1.2. The format of the message could be defined such that it would allow future additions by treating any unrecognized attributes as unsupported.

One possible format that would provide some extensibility would be a dot (".") delimited list where for each layer elements are packaged into a string that consists of a first upper-case alphabetic character naming the attribute followed by one or more alphanumeric characters carrying the attribute value.

\* - mandatory parameter

|  |  |  |
| --- | --- | --- |
| Attribute Type  (Upper case letters only) | Attribute values | Description |
| N\* | Decimal number | Decimal number of the nuh\_layer\_id |
| T | One of the following decimal numbers  1 – texture 2 - auxiliary  s1 – alpha  s2 – depth 3 - other (e.g. 3D-HEVC depth) | The type of the layer.  We need to signal if the layer is a primary, aux or 'other'. ('other' can be used to support 3D HEVC)  If aux is signalled, then you indicate the actual type by the sub-type.  E.g. T2s1 means auxiliary alpha |
| B | Decimal number | Decimal number of bit depth minus 8 |
| S | One of the following decimal numbers  0 – monochrome 1 – 4:2:0 2 – 4:2:2 3 – 4:4:4 | Decimal number of the chroma\_format\_idc |
| C | The following lower-case letters followed by a decimal number signaling the CICP parameters:  m[N] - matrix coefficients  t[N] - transfer characteristics  c[N] - colour primaries | Colour properties signaled by decimal numbers as in CICP (or colr box).  E.g.: Cm3t4c5 |
| X | Alphanumeric characters. | Profile, tier and level signaling as defined in Annex E.3 but where dots (".") are replaced by underscores ("\_"). |
| D | Dependency indication. does this layer depend on other layers?  i - independent  array separated by 'L' followed by a decimal number of the layer id to indicate inter-layer references | Examples:  Di - independent layer  DL1L12L24 - this layer depends on layers 1, 12 and 24 |
| other upper-case characters are reserved for future use |  |  |

Such a new element could be defined as also and attached after a comma (",") right after the signalled codec. In the example below the profile, tier, level signaling is simar to the signaling in HEVC but the dot separators are replaced with underscores since dots are used to separate layers.

codecs= "hvc1.1.6.L93.B0,also.N0T1B2S1X1\_6\_L93\_B0.N1T2s1B0S1X\_1\_6\_L93\_B0"

It is important to note that the also 4CC will need to be registered as a new sample entry type at MP4RA, to avoid possible collisions with future codecs.

Note that the above signaling is able to replace some of the information from the lhevcptl since it actually contains the profile/tier/level signaling information from the HEVC parameter sets that belong to codec description. This should be further studied.

### Alternatives

This section discusses several alternative solutions that do not appear to cover all use cases.

#### Attaching the new signaling directly after the codecs parameter.

This would require the omission of constraint flags to be prohibited in accordance with Annex E.3 if such an extension is present. For example:

hvc1.1.6.L93.B0 -> hvc1.1.6.L93.B0.00.00.00.00.00

And add the extension as a new codecs sub-parameter after a new delimiter e.g. ("+"). The layers could be separated using a dot (".") and within the layer the components are packed.

For example, for 2 layered L-HEVC with 10 bit 4:2:0 texture (layer0) and 8 bit 4:2:0 alpha (layer1) the signaling would look like this:

codecs= "hvc1.1.6.L93.B0.00.00.00.00.00+N0T1B2S1.N1T2B0S1"

However, current implementations would most likely ignore such extension.

#### Definition of new, more constrained HEVC profiles

Another solution would be to define new profiles in HEVC. However, it would require a definition of constrained profiles that are subset of already defined profiles and it seems to be against usual practices in JVET. The constrained profiles will most likely not be able to signal the presence of alpha and are limited to signaling the number of layers, as it does not matter to a codec what kind of content is part of a layer.

#### New MIME type parameter

This option would be to define a new MIME type parameter and add the required signaling as described in section 7.1.2. For example, a new parameter called lhevci (L-HEVC information) could be defined where each layer is separated by a comma (",") and within a layer each element is separated by a dot (".").

For example, for 2 layered L-HEVC with 10 bit 4:2:0 texture (layer0) and 8 bit 4:2:0 alpha (layer1) the signaling would look like this:

codecs= "hvc1.1.6.L93.B0"; lhevci="0.1.2.1,1.2.0.1"

However, as discussed in section 7.1.1 this method does not satisfy all use-cases and would require a combination with the method proposed in 7.1.3:

codecs="hvc1.1.6.L93.B0,also.L0T1B2S1.L1T2B0S1"; lhevci="0.1.2.1,1.2.0.1"

# Enabling multiple layers in single layer track

During MPEG #145 the contribution [m66427](https://dms.mpeg.expert/doc_end_user/current_document.php?id=91326) proposed to specify an additional method that would allow to include data from different codec families, such as AVC base + LCEVC enhancement or AVC base + HEVC enhancement layer, under the sample entry type of the base codec (e.g. 'avc1'). It was proposed to store the enhancement layer data using the sample auxiliary information concept and allow multi-layer ConfigurationBox together with single layer ConfigurationBox in the single layer sample entry. The multi-layer ConfigurationBox includes the decoder configuration record for decoding the respective sample auxiliary information. When sample auxiliary info type is equal to the sample entry type of the multi-layer bitstream, the sample auxiliary information is formatted according to the sample format indicated by the sample entry type.

During MPEG #146 the proposal was refined in contribution [m67816](https://dms.mpeg.expert/doc_end_user/current_document.php?id=92995).

## Multi-layer configuration box in a sample entry

### Clause 4 changes

This technology would require defining a new subclause under Clause 4.

**4.xx Enhancement layer storage as sample auxiliary information**

**4.xx.1 General**

Subclause 4.xx specifies the storage of enhancement layer as sample auxiliary information. Consequently, a sample entry indicating a single-layer codec may be used together with enhancement layer(s) stored as sample auxiliary information.

**4.xx.2 Sample auxiliary information type**

When aux\_info\_type is equal to a sample entry type, the sample auxiliary information is formatted according to the sample format indicated by the sample entry type and aux\_info\_type\_parameter shall be equal to 0 unless specified otherwise for the sample entry type.

### New Annex

In addition, a new Annex will be required:

**Definition**

Sample Entry Types: single layer sample entry  
Container: Sample Description Box ('stsd')  
Mandatory: A single layer sample entry is mandatory   
Quantity: One or more sample entries may be present

A single layer sample entry may contain a SingleLayerConfigurationBox, which includes a SingleLayerDecoderConfigurationRecord and a MultiLayerSAIBox. When the MultiLayerSAIBox is present in the sample entry of a track it indicates that the track contains sample auxiliary information with multilayer bitstream.

The MultiLayerSAIBox contains the MultiLayerConfigurationBox, which includes a MultiLayerDecoderConfigurationRecord.

Extension descriptors that should be inserted into the Elementary Stream Descriptor, when used in MPEG-4, may be present.

Multiple sample entries may be used, as permitted by ISO/IEC 14496-12, to indicate sections of video that use different configurations or parameter sets.

**Syntax**

class MultiLayerSAIBox extends FullBox(version=0, flags, 'mlsi'){  
 MultiLayerConfigurationBox config2;  
}

Editors note: does this need to be a FullBox? Seems like a simple container to me. Also the type MutltiLayeronfigurationBox is not defined. MultiLayerSAIBox can most likely be defined as empty and the definitions of the config boxes need to add mlsi as an additional container.

class SingleLayerSampleEntry() extends VisualSampleEntry (4cc\_for\_single\_layer) {  
 SingleLayerConfigurationBox config1;  
 MultiLayerSAIBox mlSAI;  
 MPEG4ExtensionDescriptorsBox (); // optional  
}

Editors note: similar comment as above. SingleLayerConfigurationBox is not a defined type. The entire SingleLayerSampleEntry can probably be defined with the mandatory MultiLayerSAIBox and optional boxes in it. Is the order important?

**Semantics**

SingleLayerSampleEntry indicates any sample entry carrying single layer bitstream. For example, AVCSampleEntry carrying AVC single layer bitstream of type 'avc1' or 'avc3'.

4cc\_for\_single\_layer indicates the 4CC value of the corresponding SingleLayerSampleEntry.

config1 indicates a single layer configuration box carrying the respective decoder configuration record.

mlSAI indicates a MultiLayerSAIBox containing a MultiLayerConfigurationBox carrying the respective decoder configuration record. For example, LHEVCConfigurationBox carrying LHEVCDecoderConfigurationRecord.

## Samples of the single-layer track

The samples of the single-layer tracks conform to the respective SingleLayerSampleEntry.

## Auxiliary information for multi-layer bitstream

A new sample auxiliary information is defined called the multi-layer auxiliary samples.

**Definition**

Aux Info Type: sample entry type  
Container: Sample auxiliary information  
Mandatory: No  
Quantity: Zero, or one per sample when present

A multi-layer sample auxiliary is the sample auxiliary information payload for each sample in a track carrying the sample data of the multi-layer bitstream.

**Syntax**

SampleFormat MultiLayerSampleAuxiliaryDataFormat;

**Semantics**

SampleFormat indicates the structure of the sample as defined by the sample entry type equal to aux\_info\_type. For example, the sample format of subclause 4.2.3 for a L-HEVC sample with sample entry type 'lhv1'.

MultiLayerSampleAuxiliaryDataFormat indicates an instance of the SampleFormat.