INTERNATIONAL STANDARD

ISO/IEC 14496-15

Fifth edition
2019-09

AMENDMENT 1
2020-12

Information technology — Coding of audio-visual objects —

Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format

AMENDMENT 1: Improved support for tiling and layering

Technologies de l'information — Codage des objets audiovisuels —

Partie 15: Transport de vidéo structurée en unités NAL sur la couche réseau au format ISO de base pour les fichiers médias

AMENDEMENT 1: .

Reference number
ISO/IEC 14496-15:2019/Amd.1:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

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

A list of all parts in the ISO/IEC 14496 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Information technology — Coding of audio-visual objects —

Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format

AMENDMENT 1: Improved support for tiling and layering

Clause 4

At the end of Clause 4, add a new subclause as follows:

4.13 Alternative extraction source track grouping

Members of the track group with `track_group_type` equal to `'alte'` are alternatives to be used as a source for `'scal'` or `'sabt'` track reference. The value of (`flags` & 1) shall be equal to 1 in a `TrackGroupTypeBox` of type `'alte'` to indicate the uniqueness of `track_group_id` as specified in ISO/IEC 14496-12.

A `'scal'` or `'sabt'` track reference may refer to a `track_group_id` value of an `'alte'` track group. As implied by the general semantics of a track reference to a `track_group_id` specified in ISO/IEC 14496-12, any single track of an `'alte'` track group is a valid source for extraction as specified in A.3 and A.7 or for bitstream reconstruction from tile tracks as specified in subclause 10.5.4.

8.3.3.1.3

Add the following sentence at the end of the semantics of `nalUnit` in paragraph 13:

When one or more SEI NAL units containing an SEI manifest SEI message and/or an SEI prefix indication SEI message are available, they should be stored as instances of `nalUnit`.

9.2

Replace the paragraph that starts with 'When the samples of a track contain", including the note, with the following text:

When the samples of a track contain temporal sub-layers of an HEVC base layer but do not contain, natively or through extraction, the temporal sub-layer with TemporalId equal to 0 of an HEVC base layer, an `'hvc2'`, or `'hev2'` sample entry shall be used. When the samples of a track contain, natively or through extraction, an HEVC compatible base layer or a temporal subset of an HEVC base layer including a sub-layer with TemporalId equal to 0, an `'hvc1'`, `'hev1'`, `'hvc2'`, `'hev2'`, `'hvc3'`, or `'hev3'` sample entry shall be used.

NOTE When a sample entry type `'hvc2'`, `'hev2'`, `'hvc3'` or `'hev3'` is used in a track containing the base layer, parsers complying only with non-layered HEVC storage specified in Clause 8 are not able to process the track.

9.5.3.1.1

In the definition, add `'hvc3'` and `'hev3'` to the lists in both the "Sample Entry and Box types" and "Mandatory" rows.

In the second paragraph below the definition, replace:

When the sample entry name is `'hev2'`

with:

When the sample entry name is `'hev2'` or `'hev3'`

In the third paragraph below the definition, replace:

or when the sample entry name is `'hev1'` or `'hev2'`

with:

or when the sample entry name is `'hev1'`, `'hev2'` or `'hev3'`

Replace the paragraph immediately above NOTE 2 with the following:

In case two parameter sets with different content but using the same ID are present, it may not be possible to use a single sample entry of type `'hvc1'`, `'hvc2'`, `'hvc3'` or `'lhv1'`; file packagers should create either different sample entries of type `'hvc1'`, `'hvc2'`, `'hvc3'` or `'lhv1'`, or use in-band parameter sets through `'hev1'`, `'hev2'`, `'hev3'` or `'lhe1'` sample entries.

Replace the two paragraphs immediately below NOTE 3 with the following:

For an HEVC or L-HEVC bitstream carried in more than one track, when the sample entry name of the base track is `'hvc1'`, `'hvc2'` or `'hvc3'`, the sample entry name of other tracks carrying the same bitstream shall be `'hvc2'`, `'hvc3'` or `'lhv1'`, and when the sample entry name of the base track is `'hev1'`, `'hev2'` or `'hev3'`, the sample entry name of other tracks carrying the same bitstream shall be `'hev2'`, `'hev3'` or `'lhe1'`.

For an L-HEVC bitstream whose base layer is an AVC bitstream, when the sample entry name of the base track is `'avc1'` or `'avc2'`, the sample entry name of the other tracks carrying the associated L-HEVC bitstream shall be `'hvc2'`, `'hvc3'`, or `'lhv1'`, and when the sample entry name of the base track is `'avc3'` or `'avc4'`, the sample entry name of the other HEVC tracks carrying the associated L-HEVC bitstream shall be `'hev2'`, `'hev3'` or `'lhe1'`.

In both NOTE 6 and the sentence immediately following NOTE 6, replace (two occurrences):

`'hvc1'`, `'hev1'`, `'hvc2'`, or `'hev2'`

with:

`'hvc1'`, `'hev1'`, `'hvc2'`, `'hev2'`, `'hvc3'` or `'hev3'`

Replace Table 11 with the following:

Table 11 — Use of sample entries for HEVC and L-HEVC tracks

sample entry name	with configuration records	meaning
'hvc1' or 'hev1'	HEVC Configuration Only	A plain HEVC track without NAL units with nuh_layer_id greater than 0; Extractors and aggregators shall not be present.
'hvc1' or 'hev1'	HEVC and L-HEVC Configurations	An L-HEVC track with both NAL units with nuh_layer_id equal to 0 and NAL units with nuh_layer_id greater than 0; Extractors and aggregators shall not be present.
'hvc2' or 'hev2'	HEVC Configuration Only	A plain HEVC track without NAL units with nuh_layer_id greater than 0; Extractors may be present and used to reference NAL units; constructor_type shall be equal to 0 or 2 in extractors; Aggregators may be present to contain and reference NAL units.
'hvc2' or 'hev2'	HEVC and L-HEVC Configurations	An L-HEVC track with both NAL units with nuh_layer_id equal to 0 and NAL units with nuh_layer_id greater than 0; Extractors and aggregators may be present; Extractors may reference any NAL units; constructor_type shall be equal to 0 or 2 in extractors; Aggregators may both contain and reference any NAL units.
'hvc3' or 'hev3'	HEVC Configuration Only	A plain HEVC track without NAL units with nuh_layer_id greater than 0; Extractors may be present and used to reference NAL units; constructor_type shall be equal to 0, 2, 3, 4, 5 or 6 in extractors; Aggregators may be present to contain and reference NAL units.
'hvc3' or 'hev3'	HEVC and L-HEVC Configurations	An L-HEVC track with both NAL units with nuh_layer_id equal to 0 and NAL units with nuh_layer_id greater than 0; Extractors and aggregators may be present; Extractors may reference any NAL units; constructor_type shall be equal to 0, 2, 3, 4, 5 or 6 in extractors; Aggregators may both contain and reference any NAL units.
'lhv1', 'lhe1'	L-HEVC Configuration Only	An L-HEVC track with NAL units with nuh_layer_id greater than 0 and without NAL units with nuh_layer_id equal to 0; Extractors shall not be present; Aggregators may be present to contain and reference NAL units.

9.5.3.1.3

In the second sentence, replace:

When the sample entry is 'hvc2' or 'hev2'

with:

When the sample entry is 'hvc2', 'hev2', 'hvc3' or 'hev3'

9.5.4

In the second and third list items, replace:

the sample entry type is `'hvc2'` or `'hev2'`

with:

the sample entry type is `'hvc2'`, `'hev2'`, `'hvc3'` or `'hev3'`

9.5.5

In list item 3, replace:

if the sample entry type is `'hvc2'` or `'hev2'` and the track contains extractors,

with:

if the sample entry type is `'hvc2'`, `'hev2'`, `'hvc3'` or `'hev3'` and the track contains extractors,

In list item 4, replace:

Otherwise, if the sample entry type is `'hvc2'` or `'hev2'`

with:

Otherwise, if the sample entry type is `'hvc2'`, `'hev2'`, `'hvc3'` or `'hev3'`

9.5.8

In the second sentence, replace:

When included in an `'hvc2'` or `'hev2'` track that is not the base track and contains extractors,

with:

When included in an `'hvc2'`, `'hvc3'`, `'hev2'`, or `'hev3'` track that is not the base track and that contains extractors,

In the paragraph immediately after the NOTE, replace:

The presence of the `'sync'`, `'roll'`, and `'rap'` sample groups in `'hvc2'`, `'hev2'`, `'lhv1'`, or `'lhe1'` tracks

with:

The presence of the `'sync'`, `'roll'`, and `'rap'` sample groups in `'hvc2'`, `'hvc3'`, `'hev2'`, `'hev3'`, `'lhv1'`, or `'lhe1'` tracks

10.5.1

In the first paragraph, replace:

For such cases, tile tracks may be created using the `HEVCTileSampleEntry` or `LHEVCTileSampleEntry` sample description format

with:

For such cases, tile tracks may be created using the `HEVCTileSampleEntry`, `HEVCTileSSHInfoSampleEntry` or `LHEVCTileSampleEntry` sample description format

In the second paragraph, replace:

The sample entry type for an HEVC tile track is 'hvt1'

with:

The sample entry type for an HEVC tile track is 'hvt1' or 'hvt3'

Add the following paragraph at the end of subclause 10.5.1:

When a timed metadata track is linked to a tile base track with a `'cdsc'` track reference, it describes the HEVC video bitstream carried by the tile base track and all the associated tile tracks, and in this case the timed metadata track shall not be linked to the associated tile tracks.

10.5.4

in the first paragraph, add the following text at the end of the first sentence:

When a `'sabt'` track reference points to a `track_group_id` of an `'alte'` track group, any single track of the `'alte'` track group is a valid tile track to be used in the bitstream reconstruction.

10.5.5

At the end of 10.5.5, add new subclauses as follows:

10.5.6 HEVC Tile Track with Slice Segment Header Info

10.5.6.1 Definition

Sample Entry Type: `'hvt3'`

Container: Sample Description Box (`'stsd'`)

Mandatory: No

Quantity: Zero or more sample entries may be present

An `'hvt3'` track shall have a `'tbas'` track reference to an HEVC tile base track. The specifications for HEVC tile track specified in 10.5 apply to the `'hvt3'` track. The width and height of the `VisualSampleEntry` for an HEVC tile track (sample entry type `'hvt3'`) shall be set to the width and height of the minimum bounding box enclosing all tile regions contained in the track. The layout information in the track header (i.e. layer, matrix, width and height) of an HEVC tile track shall be ignored by file parsers. `CleanApertureBox` and `PixelAspectRatioBox` shall not be present in an `'hvt3'` sample description.

For each VCL NAL unit in `'hvt3'` tracks there shall be a preceding `SliceSegmentHeaderInfo` NAL-unit-like structure that documents its slice segment header length.

NOTE Even though `SliceSegmentHeaderInfo` `NAL`-unit-like structures are informational in nature, client implementations can rely on their presence for correct behaviour and performance reasons. Signalling a track as `'hvt3'` allows such clients to check compatibility.

10.5.6.2 Syntax

```
class HEVCTileSSHInfoSampleEntry() extends VisualSampleEntry ('hvt3'){
   HEVCTileConfigurationBox   config(); // optional
}
```

10.5.6.3 Semantics

The constraints and semantics of `HEVCTileSSHInfoSampleEntry` are identical to those of `HEVCTileSampleEntry` as specified in subclause 10.5.2.3.

10.6 HEVC slice segment data track

10.6.1 Overview

The general definition of sample format as provided in subclause 4.3.3 does not apply to the definition of 'hvt2' tracks.

The sample format of an `'hvt2'` track consists of one and only one instance of the HEVC syntax elements slice_segment_data( ) and rbsp_slice_segment_trailing_bits( ) of an independent slice segment. No other data is present in samples of `'hvt2'` tracks.

`'hvt2'` tracks avoid the need to have a slice segment header redundantly present for applications where the slice segment header is adjusted depending on which composition of tracks is merged to a bitstream to be decoded. Appropriate slice segment headers for an `'hvt2'` track are present in extractor tracks that include samples from the `'hvt2'` tracks by reference of type `'scal'`. It is not possible to process an `'hvt2'` track without an `'hvc2'`, `'hev2'`, `'hvc3'`, or `'hev3'` track that contains slice segment headers natively and the respective slice segment data by reference from the `'hvt2'` track through extractors.

`track_in_movie` shall be equal to 0 in the `TrackHeaderBox` of `'hvt2'` tracks.

10.6.2 Sample entry name and format for HEVC slice segment data tracks

10.6.2.1 Definition

Sample Entry Type: `'hvt2'`

Container: Sample Description Box (`'stsd'`)

Mandatory: No

Quantity: Zero or more sample entries may be present

This sample entry describes the media samples of an HEVC slice segment data track. The `width` and `height` of the `VisualSampleEntry` for an HEVC slice segment data track (sample entry type `'hvt2'`) shall be set to the width and height of the minimum bounding box enclosing the independent slice segments contained in the track. The layout information in the track header (i.e. `layer`, `matrix`, `width` and `height`) of an HEVC slice segment data track shall be ignored by file parsers. `CleanApertureBox` and `PixelAspectRatioBox` shall not be present in an `'hvt2'` sample description.

The sample format of an `'hvt2'` track shall consist of one and only one instance of the HEVC syntax elements slice_segment_data( ) and rbsp_slice_segment_trailing_bits( ) of a typically independent slice segment. No other data shall be present in samples of `'hvt2'` tracks.

10.6.2.2 Syntax

```
class HEVCSliceSegmentDataSampleEntry() extends VisualSampleEntry ('hvt2'){
   HEVCTileConfigurationBox   config(); // optional
}
```


10.6.2.3 Semantics

The `HEVCSliceSegmentDataSampleEntry` shall not contain any `HEVCConfigurationBox`, `LHEVCConfigurationBox` or `MPEG4ExtensionDescriptorsBox`; these boxes are found in the sample description of the track containing extractors for including the slice segment data by reference. Other optional boxes may be included.

Optionally, the `HEVCSliceSegmentDataSampleEntry` may contain one `HEVCTileConfigurationBox`, used to indicate the tier and level information in the case the slice segment data in this track is for a motion-constrained tile or tile set.

> `Compressorname` in the base class `VisualSampleEntry` indicates the name of the compressor used with the value `"\025HEVC Slice Data Coding"` being recommended; the first byte is a count of the remaining bytes, here represented by \025, which (being octal 25) is 21 (decimal), the number of bytes in the rest of the string.

All `'hvt2'` tracks referenced by the same extractor track and the extractor track shall share the same timescale.

NOTE If an `'hvt2'` track is removed from a file, all extractor tracks that refence the `'hvt2'` track have to be removed too. If an extractor track is removed from a file, all `'hvt2'` tracks that the extractor track references ought to be removed too provided that there is no other extractor track referencing them.

A.3.1

In the final paragraph, add the following text at the end of the first sentence:

When a `'scal'` track reference points to a `track_group_id` of an `'alte'` track group, any single track of the `'alte'` track group is a valid source for extraction.

A.7.1

Add the following items in the list in the second paragraph:

c) A sample constructor from a track group extracts, by reference, NAL unit data (either entire NAL unit or NAL unit payload) from a sample of another track or track in a track group.

d) A reference constructor allows referencing default constructors declared in a list in sample entry, with optional override of default constructor fields.

e) A NAL unit start constructor marks the beginning of a reconstructed NAL unit of possibly variable size (depending on the current track selection).

In the final paragraph, add the following text at the end of the first sentence:

When a `'scal'` track reference points to a `track_group_id` of an `'alte'` track group, any single track of the `'alte'` track group is a valid source for extraction.

A.7.2

Replace the syntax with the following:

```
class aligned(8) Extractor () {
   NALUnitHeader();
   do {
      unsigned int(8)constructor_type;
      if( constructor_type == 0 )
         SampleConstructor();
      else if( constructor_type == 2 )
         InlineConstructor();
      else if( constructor_type == 3 )
         SampleConstructorFromTrackGroup();
      else if( constructor_type == 4 )
         ReferenceConstructor();
      else if( constructor_type == 5 )
         DefaultReferenceConstructor();
      else if( constructor_type == 6 )
         NALUStartInlineConstructor();
   } while( !EndOfNALUnit() )
}
```

A.7.3

Replace the definition of `constructor_type` with the following:

`constructor_type` specifies the constructor that follows. `SampleConstructor`, `InlineConstructor`, `SampleConstructorFromTrackGroup`, `ReferenceConstructor`, `DefaultReferenceConstructor` and `NALUStartInlineConstructor` correspond to `constructor_type` equal to 0, 2, 3, 4, 5 and 6, respectively. Other values of `constructor_type` are reserved.

A.7.4.1.2

Change NOTE to NOTE 1 and add the following as NOTE 2:

NOTE 2 When `track_ref_index` references a track group, file writers are expected to select the values of `data_offset` and `data_length` carefully. For example, when the referenced samples in all the tracks of the track group consist of one and only one VCL NAL unit and the slice segment headers in all these VCL NAL units have the same length, it is possible to use a non-zero `data_offset` to point to the first byte of the slice segment data. When the referenced samples in all tracks of the track group contain one and only one NAL unit, it is possible to use a `data_length` value that points beyond the sample size of any referenced sample to extract bytes from `data_offset` until the end of the sample from any track of the track group.

A.7.5.2

At the end of A.7.5.2, add new subclauses as follows:

A.7.6 Sample constructor from a track group

A.7.6.1 Syntax

```
class aligned(8) SampleConstructorFromTrackGroup () {
      unsigned int(8) track_ref_index;
      signed   int(8) sample_offset;
      unsigned int(2) copy_mode; // sample, NALU, NALU payload
      if (copy_mode != 0) {
            unsigned int(1) nalu_idx_field_size;
            unsigned int(5) reserved;
            unsigned int((nalu_idx_field_size + 1) * 8) nalu_idx;
      } else {
            unsigned int(6) reserved;
      }
}
```


A.7.6.2 Semantics

`track_ref_index` specifies the index of the track reference of type `'scal'` to use to find the `track_ID` or the `track_group_id` from which to extract data. When the `track_ref_index` resolves to a `track_group_id`, it is up to the parser or player to select the most appropriate track in the corresponding track group depending on the `track_group_type`. A default behaviour is to select the first track in the file having the specified `track_group_id`.

`copy_mode`: specifies the copy operation to be performed when resolving the extractor:

— When set to 0, it means a sample copy, i.e. a copy of bytes from the first byte of the sample until the end of the sample, inclusive.

— When set to 1, it means a NAL unit copy, i.e. a copy from the first byte of the i-th NAL unit to the last byte of this same NAL unit, where i corresponds to the `nalu_idx` field.

— When set to 2, it means a NAL unit payload copy; i.e. a copy from the first byte immediately following the NAL unit header in the i-th NAL unit payload to the last byte of this same NAL unit, where i corresponds to the `nalu_idx` field.

— `copy_mode` 3 is reserved for future use.

NOTE `copy_mode` 2 is useful when some header rewriting is performed. In such case, only NALU payload is extracted and combined with rewritten NALU header, e.g. when some NALUs from different IRAP and non-IRAP pictures are merged in one single picture, there can be a need to rewrite `nal_unit_type` in NALU headers.

`nalu_idx`: 1-based index of the NAL unit from where to extract. Value 0 is reserved. NAL-unit-like structures and NAL units that are present in the sample and have `nal_unit_type` value in the range of 48 to 63, inclusive, shall not be accounted for. NAL units included or referenced by an Aggregator shall be accounted for.

A.7.7 Reference constructors

A.7.7.1 Overview

In typical usage with slice or tile extraction, fields in the constructor(s) of an extractor stay constant. For example, the same `track_ref_index` value is used in respective extractors in all samples and `sample_offset` is typically 0.

This subclause specifies a mechanism to carry default syntax element values that are used instead of including the respective syntax elements in constructors.

Reference constructors provide a mechanism to include a constructor by reference in an extractor rather than having the constructor data in band. This allows for sharing fields or complete constructors repeated across extractors.

A.7.7.2 Syntax

```
class aligned(8) ReferenceConstructor
   BaseReferenceConstructor(0);

class aligned(8) DefaultReferenceConstructor
   BaseReferenceConstructor(1);

class aligned(8) BaseReferenceConstructor (defaultIndexFlag) {
   if (!defaultIndexFlag)
      unsigned int(8) ref_index;
```

```
    if (flags) {
       switch (ref_type) {
    case 0: //sample constructor
       if (flags & 1)
          unsigned int(8) track_ref_index;
       if (flags & 2)
          signed int(8) sample_offset;
       if (flags & 4)
          unsigned int((lengthSizeMinusOne+1)*8) data_offset;
       if (flags & 8)
          unsigned int((lengthSizeMinusOne+1)*8) data_length;
    break;

    case 2: //inline constructor
    case 6: //inline constructor
       if (flags & 1) {
          unsigned int(8) length;
          unsigned int(8) inline_data[length];
       }
       break;

    case 3: // constructor from track group
          if (flags & 1)
             unsigned int(8) track_ref_index;
          if (flags & 2)
             signed int(8) sample_offset;
          if (flags & 4) {
             unsigned int(2) copy_mode;
             if (copy_mode != 0) {
                unsigned int(1) nalu_idx_field_size;
                unsigned int(5) reserved;
                unsigned int((nalu_idx_field_size + 1) * 8) nalu_idx;
             }
             else
                unsigned int(6) reserved;
          }
          break;
       }
}
```

A.7.7.3 Semantics

`ref_index` specifies the index of the constructor to use in the list of constructors of the `DefaultHevcExtractorConstructorBox` in sample entry of the track extracting the data. A value of 0 indicates the first entry. When `ref_index` is not present in the syntax structure, `ref_index` is inferred to be equal to 1 + `ref_index` of the previous `ReferenceConstructor` or `DefaultReferenceConstructor` in the same extractor, if any, or 0 otherwise (when the containing `ReferenceConstructor` or `DefaultReferenceConstructor` is the first among all the `ReferenceConstructors` and `DefaultReferenceConstructors` in the containing extractor).

`ref_type` is the type of the constructor (i.e. the value of `constructor_type`) referenced by `ref_index`.

`flags` are the flags defined for the constructor referenced by `ref_index` in `DefaultHevcExtractorSampleBox`.

`track_ref_index`, `sample_offset`, `data_offset`, `data_length`, `length`, `inline_data`, `copy_mode`, `nalu_idx_field_size`, and `nalu_idx`, if present, override the value of the field with the same name in the referenced constructor for this occurrence.

A.7.8 Default HEVC extractor constructor box

A.7.8.1 Definition

Box Type: 'dhec'

Container: HEVCSampleEntry

Mandatory: No

Quantity: Zero or one for sample entry types 'hvc3' and 'hev3'.

Zero for other sample entry types specified in ISO/IEC 14496-15.

DefaultHevcExtractorConstructorBox provides a list of constructors to be used in place of ReferenceConstructor or DefaultReferenceConstructor present in a sample. The constructors given in this box shall only be resolved as a replacement of a ReferenceConstructor or DefaultReferenceConstructor in a sample.

A.7.8.2 Syntax

```
aligned(8) class DefaultHevcExtractorConstructorBox extends FullBox('dhec'){
   unsigned int(32) num_entries;
   for (i=1; i<= num_entries; i++) {
      unsigned int(8) constructor_type;
      unsigned int(8) flags;
      if( constructor_type == 0 )
         SampleConstructor();
      else if( constructor_type == 2 )
         InlineConstructor();
      else if( constructor_type == 3 )
         SampleConstructorFromTrackGroup();
      else if( constructor_type == 6 )
         NALUStartInlineConstructor ();
   }
}
```

A.7.8.3 Semantics

num_entries gives the number of default constructors defined in this box

constructor_type gives the type of the constructor for this entry. Value 4 (ReferenceConstructor) and 5 (DefaultReferenceConstructor) shall not be present.

flags gives the flags associated with the constructor. These flags are used when processing the reference to override some of the constructor fields.

A.7.9 NALU start inline constructor

A.7.9.1 Definition

The NALUStartInlineConstructor is used to indicate that a NAL unit starts at this constructor, and expands until but excluding the next NALUStartInlineConstructor, if any, or to the end of the extractor, otherwise. The inline_data contains the beginning of the NAL unit, starting with NAL unit header, but does not contain any NALUnitLength field. This field shall be inserted by the file reader according to the track LengthSizeMinusOne field, and set to the complete NAL unit size after processing this constructor and the following constructors until but excluding the next NALUStartInlineConstructor, if any, or to the end of the extractor, otherwise.

A.7.9.2 Syntax

```
class aligned(8) NALUStartInlineConstructor () {
   unsigned int(8) length;
   unsigned int(8) inline_data[length];
}
```

A.7.9.3 Semantics

`length`: the number of bytes that belong to the `NALUStartInlineConstructor` following this field. The value of `length` shall be greater than 0. The value of `length` equal to 0 is reserved.

`inline_data`: the data bytes to be returned when resolving the constructor. These bytes shall contain either exactly one complete NAL unit if this is the last constructor in the extractor, or the beginning of a NAL unit.

A.10 Slice segment header information NAL-unit-like structure

A.10.1 General

A `SliceSegmentHeaderInfo` NAL-unit-like structure is a file format internal that assists file readers in the parsing and rewriting of subsequent NAL units present in the samples of the track.

Like Aggregators and Extractors, it uses a syntax that is similar to the NAL unit syntax but does not follow the start code emulation prevention mechanism required for the NAL unit syntax as specified in ISO/IEC 23008-2. These NAL-unit-like structures are seen as NAL units in the context of the sample structure. `SliceSegmentHeaderInfo` shall not be output by file parsers.

`SliceSegmentHeaderInfo` NAL-unit-like structures use a NAL unit type reserved for the application/transport layer by ISO/IEC 23008-2, as indicated in Annex F.

NOTE 1 See ISO/IEC 23008-2:2017, subclause 7.4.2.2 on exercising care in the design in encoders that generate NAL units with nal_unit_type_values UNSPEC48..UNSPEC63.

A.10.2 Definition

This subclause describes `SliceSegmentHeaderInfo` NAL-unit-like structures, which enable signaling of various properties of slice_segment_header( ) of N immediately following HEVC VCL NAL units, respectively, in order to assist in slice manipulation.

When present in an ISO/IEC 23008-2 video, `SliceSegmentHeaderInfo` NAL-unit-like structures use the NAL unit header as defined in ISO/IEC 23008-2, which has the same syntax for plain HEVC and layered HEVC.

Each `SliceSegmentHeaderInfo` NAL-unit-like structure contains one or more information elements. Each information element contains the properties of the slice_segment_header( ) of a single subsequent VCL NAL unit. The number of information elements in a single `SliceSegmentHeaderInfo` NAL-unit-like structure, and thus the number of VCL NAL units to which it applies, is indicated by the `number_of_vcl_nals` field. To determine to which VCL NAL unit an information element in a `SliceSegmentHeaderInfo` NAL-unit-like structure applies, the following rules apply to the bitstream that is contained in the samples of the track (i.e. not the reconstructed bitstream obtained by resolving extractors, if any, or any implicit reconstruction process):

— The first information element in a `SliceSegmentHeaderInfo` NAL-unit-like structure applies to the first subsequent VCL NAL unit in the bitstream.

— If the `SliceSegmentHeaderInfo` contains more than one information element, the information elements apply to any subsequent VCL NAL units in the order in which they occur in the bitstream.

— `SliceSegmentHeaderInfo` NAL-unit-like structures exclusively apply to VCL NAL units. Non-VCL NAL units are not considered when determining the NAL unit to which the information in a `SliceSegmentHeaderInfo` NAL-unit-like structure applies. As an example, if a `SliceSegmentHeaderInfo` NAL-unit-like structure contains information for eight VCL NAL units, this information is applied to the first eight VCL NAL units after it in the bitstream, regardless of any non-VCL NAL units that may be present.

— A subsequent `SliceSegmentHeaderInfo` NAL-unit-like structure always takes precedence over and discards any information elements provided by earlier `SliceSegmentHeaderInfo` NAL-unit-like structure. As an example, if a bitstream contains a `SliceSegmentHeaderInfo` NAL-unit-like

