
**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**

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*



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 14496-15:2022



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Contents

Page

Foreword.....	vi
Introduction.....	vii
1 Scope	1
2 Normative references	1
3 Terms, definitions, abbreviated terms and conventions.....	1
3.1 Terms and definitions.....	1
3.2 Abbreviated terms	10
3.3 Conventions.....	11
4 General definitions	12
4.1 Overview.....	12
4.2 Sample and configuration definition.....	12
4.3 Video track structure	14
4.4 Template fields used	14
4.5 Visual width and height.....	14
4.6 Decoding time (DTS) and composition time (CTS).....	15
4.7 Sample groups on random access recovery points 'roll' and random access points 'rap'	15
4.8 Hinting.....	16
4.9 On change of sample entry (informative).....	16
4.10 SEI information box	18
4.11 Post-decoder requirements scheme for signalling of SEI	18
4.12 Alternative extraction source track grouping	19
4.13 NAL unit map entry.....	19
4.14 Rectangular region group entry.....	21
4.15 Layer information sample group.....	23
5 AVC elementary streams and sample definitions.....	25
5.1 Overview.....	25
5.2 Elementary stream structure.....	25
5.3 Sample and configuration definition.....	28
5.4 Derivation from ISO base media file format	32
6 SVC elementary stream and sample definitions.....	44
6.1 Overview	44
6.2 Elementary stream structure.....	44
6.3 Use of the plain AVC file format.....	45
6.4 Sample and configuration definition.....	45
6.5 Derivation from the ISO base media file format.....	47
7 MVC and MVD elementary stream and sample definitions	53
7.1 Overview	53
7.2 Overview of MVC or MVD Storage.....	55
7.3 MVC and MVD elementary stream structures	56

7.4	Use of the plain AVC file format	57
7.5	Sample and configuration definition	58
7.6	Derivation from the ISO base media file format	61
7.7	MVC specific information boxes.....	76
8	HEVC elementary streams and sample definitions.....	86
8.1	Overview	86
8.2	Elementary stream structure	86
8.3	Sample and configuration definition	87
8.4	Derivation from ISO base media file format.....	92
9	Layered HEVC elementary stream and sample definitions	101
9.1	Overview	101
9.2	Overview of L-HEVC storage.....	102
9.3	L-HEVC elementary stream structure.....	103
9.4	Sample and configuration definition	103
9.5	Derivation from the ISO base media file format and the HEVC file format (Clause 8)	105
9.6	L-HEVC specific structures.....	116
10	Storage of tiled HEVC and L-HEVC video streams.....	122
10.1	Overview	122
10.2	NAL unit map entry	123
10.3	Tile region group entry	123
10.4	Tile sub track definition	123
10.5	HEVC and L-HEVC tile track	124
10.6	HEVC slice segment data track	129
11	VVC elementary streams and sample definitions	130
11.1	Overview	130
11.2	Sample and configuration definition	137
11.3	Derivation from ISO base media file format.....	146
11.4	Sample groups	160
11.5	Entity groups	180
11.6	Data sharing and VVC bitstream reconstruction	188
12	EVC elementary streams and sample definitions.....	199
12.1	Overview	199
12.2	Elementary stream structure	199
12.3	Sample and configuration definition	200
12.4	Derivation from ISO base media file format.....	203
Annex A (normative)	In-stream structures	210
Annex B (normative)	SVC, MVC, and MVD sample group and sub-track definitions.....	228
Annex C (normative)	Temporal metadata support.....	251
Annex D (normative)	File format toolsets and brands	260
Annex E (normative)	Sub-parameters for the MIME type 'codecs' parameter.....	264

Annex F (informative) Unspecified nal_unit_type value management for sample entry types of AVC and HEVC.....	273
Annex G (informative) Examples of VVC base and subpicture tracks.....	275

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 14496-15:2022

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 or www.iec.ch/members_experts/refdocs).

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. In the IEC, see www.iec.ch/understanding-standards.

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*.

This sixth edition cancels and replaces the fifth edition (ISO/IEC 14496-15:2019), which has been technically revised. It also incorporates the Amendment ISO/IEC 14496-15:2019/Amd 1:2020.

The main changes are as follows:

- Support for the Versatile Video Coding (ISO/IEC 23090-3) and Essential Video Coding (ISO/IEC 23094-1)
- Addition of sample entry types 'hvc3', 'hev3', 'hvt2', and 'hvt3' targeted at tile-based delivery and merging of High Efficiency Video Coding (ISO/IEC 23008-2) bitstreams

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

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 and www.iec.ch/national-committees.

Introduction

This document defines a storage format based on, and compatible with, the ISO Base Media File Format (ISO/IEC 14496-12), which is used by the MP4 file format (ISO/IEC 14496-14) and the Motion JPEG 2000 file format (ISO/IEC 15444-3) among others. This document enables video streams formatted as Network Adaptation Layer Units (NAL Units) to

- a) be used in conjunction with other media streams, such as audio,
- b) be used in an MPEG-4 systems environment, if desired,
- c) be formatted for delivery by a streaming server, using hint tracks, and
- d) inherit all the use cases and features of the ISO Base Media File Format on which MP4 and MJ2 are based.

This document may be used as a standalone document; it specifies how NAL unit structured video content shall be stored in an ISO Base Media File Format compliant format. However, it is normally used in the context of a specification, such as the MP4 file format, derived from the ISO Base Media File Format, that permits the use of NAL unit structured video such as AVC (ISO/IEC 14496-10) video and High Efficiency Video Coding (HEVC, ISO/IEC 23008-2) video.

The ISO Base Media File Format is becoming increasingly common as a general-purpose media container format for the exchange of digital media, and its use in this context should accelerate both adoption and interoperability.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents or patents.iec.ch.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 14496-15:2022

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

1 Scope

This document specifies the storage format for streams of video that is structured as NAL Units, such as AVC (ISO/IEC 14496-10) and HEVC (ISO/IEC 23008-2) video streams. In addition, Annex E specifies parameters and sub-parameters applying when sample entries specified in this document are used as the 'codecs' parameter of a MIME type, as specified in IETF RFC 6381.

2 Normative references

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

ISO/IEC 14496-12:2020, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 14496-10:2020, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*

ISO/IEC 23008-2:2020, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

ISO/IEC 23090-3:2021, *Information technology — Coded representation of immersive media — Part 3: Versatile video coding*

ISO/IEC 23094-1:2020, *Information technology — General video coding — Part 1: Essential video coding*

IETF RFC 4648, *The Base16, Base32, and Base64 data encodings*

IETF RFC 6381, *MIME codecs and profiles*

3 Terms, definitions, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 14496-10, ISO/IEC 23008-2, ISO/IEC 23090-3 or ISO/IEC 23094-1, and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1.1

3D-AVC NAL unit

3D AVC VCL NAL unit

NAL unit with type 21 with `avc_3d_extension_flag` equal to 1

3.1.2

aggregator

in-stream structure using a NAL unit header for grouping of NAL units belonging to the same sample

3.1.3

alternate region set

set of rectangular regions that are alternatives to be used as a rectangular region when reconstructing a VVC bitstream from a VVC extraction base track

3.1.4

applicable video coding standard

video coding standard for the data carried in the track

Note 1 to entry: The video coding standard can be ISO/IEC 14496-10, ISO/IEC 23008-2, ISO/IEC 23090-3, or ISO/IEC 23094-1.

3.1.5

AU- or picture-level non-VCL NAL unit

non-VCL NAL unit that applies to one or more entire AUs or one or more entire pictures

Note 1 to entry: An AU-level non-VCL NAL unit applies to one or more entire AUs. A picture-level non-VCL NAL unit applies to one or more entire pictures. In VVC, AU-level or picture-level non-VCL NAL units include: 1) all the DCI, OPI, VPS, SPS, PPS, AUD, PH, EOS, and EOB NAL units; 2) APS NAL units that apply to one or more entire AUs or pictures; and 3) SEI NAL units that only contain SEI messages that apply to one or more entire AUs or pictures.

3.1.6

AVC base layer

maximum subset of a bitstream that is AVC compatible

Note 1 to entry: The AVC base layer is represented by AVC VCL NAL units and associated non-VCL NAL units. The AVC base layer is not using any of the functionality of ISO/IEC 14496-10:2020, Annex G, Annex H, Annex I, or Annex J.

Note 2 to entry: The AVC base layer itself can be a temporal scalable bitstream.

3.1.7

AVC parameter set sample

sample in a parameter set elementary stream that consists of those parameter set NAL units that are to be considered as if present in the video elementary stream at the same instant in time

3.1.8

AVC sample

access unit as defined in ISO/IEC 14496-10

3.1.9**AVC NAL unit**

AVC VCL NAL unit and its associated non-VCL NAL units in a bitstream

3.1.10**AVC VCL NAL unit**

NAL unit with type 1 to 5 (inclusive)

3.1.11**canonical order**

order of NAL units that conforms to the applicable video standard

Note 1 to entry: When a single track carries a video bitstream, the NAL units are stored in the canonical order. When multiple tracks are used to carry a video bitstream, an implicit or explicit video bitstream reconstruction process might be applied to recover the canonical order.

3.1.12**canonical stream format**

elementary stream that contains NAL units in the canonical order and conforms to the constraints specified in this document for carrying an elementary stream of the applicable video standard in one or more tracks

3.1.13**complete subset**

minimal set of tracks that contain all the information in the original bitstream

3.1.14**cropped frame dimensions**

width and height of the decoded frame after applying the output cropping parameters

3.1.15**default sample group description index**

default_group_description_index of SampleGroupDescriptionBox with version greater than or equal to 2

3.1.16**elementary stream**

sequence of one or more bitstreams of the applicable video standard

Note 1 to entry: The term elementary stream is not directly related to the terms video elementary stream, parameter set elementary stream, and video and parameter set elementary stream.

Note 2 to entry: The applicable video standard can be included as a prefix to the term elementary stream. For example, an AVC elementary stream refers to an elementary stream that is a sequence of one or more bitstreams conforming to ISO/IEC 14496-10.

3.1.17**extractor**

in-stream structure using a NAL unit header for extraction of data from other tracks

Note 1 to entry: Extractors contain instructions on how to extract data from other tracks. Logically an Extractor can be seen as a pointer to data. While reading a track containing Extractors, the Extractor is replaced by the data it is pointing to.

3.1.18

HEVC sample

access unit as defined in ISO/IEC 23008-2

3.1.19

implicit reconstruction

reconstruction of a stream of access units from two or more tracks not using extractors

3.1.20

in-stream structure

structure residing within sample data

3.1.21

layer

scalable layer

<SVC, MVC, and MVD> set of VCL NAL units with the same values of dependency_id, quality_id, and temporal_id, and the associated non-VCL NAL units

Note 1 to entry: A scalable layer with any of dependency_id, quality_id, and temporal_id not equal to 0 enhances the video by one or more scalability levels in at least one direction (temporal, quality or spatial resolution)

Note 2 to entry: SVC uses a "layered" encoder design that results in a bitstream representing "coding layers". In some publications the 'base layer' is the first quality layer of a specific coding layer. In some publications the base layer is the scalable layer with the lowest priority. The SVC file format uses "scalable layer" or "layer" in a general way for describing nested bitstreams (using terms like AVC base layer or SVC enhancement layer).

3.1.22

layer

scalable layer

<HEVC and VVC> set of VCL NAL units with the same value of nuh_layer_id and the associated non-VCL NAL units

3.1.23

layer set

set of layers represented within a bitstream created from another bitstream by operation of the sub-bitstream extraction process

3.1.24

L-HEVC sample

picture units that are within an access unit as specified in Annex F of ISO/IEC 23008-2:2020 and are represented by the track

3.1.25

MVC NAL unit

MVC VCL NAL unit and its associated non-VCL NAL units in an MVC stream

Note 1 to entry: The association of non-VCL NAL units with MVC VCL NAL units is specified in ISO/IEC 14496-10:2020, Annex H.

3.1.26

MVC sample

one or more view components as defined in Annex H of ISO/IEC 14496-10:2020 and the associated non-VCL NAL units

3.1.27**MVC VCL NAL unit**

NAL unit with type 20, and NAL units with type 14 when the immediately following NAL units are AVC VCL NAL units

Note 1 to entry: MVC VCL NAL units do not affect the decoding process of a legacy AVC decoder.

3.1.28**MVC+D depth NAL unit****MVC+D depth VCL NAL unit**

NAL unit with type 21 containing a coded slice extension for a depth view component

3.1.29**MVD NAL unit****MVD VCL NAL unit**

NAL unit with type 21, containing a coded slice extension for a depth view component coded with MVC+D or 3D-AVC, or a 3D-AVC texture view component

3.1.30**MVD sample**

one or more view components as defined in Annex I or Annex J of ISO/IEC 14496-10:2020 and the associated non-VCL NAL units, where each view component contains a texture view component, a depth view component or both

3.1.31**NAL-unit-like structure**

data structure that is similar to NAL units in the sense that it also has a NAL unit header and a payload, with a difference that the payload might not follow the start code emulation prevention mechanism required for the NAL unit syntax

3.1.32**natively present**

not included in an aggregator or an extractor

Note 1 to entry: Data referred to by (hence not included in) an aggregator is considered as natively present. Data included in an aggregator is not considered as natively present.

3.1.33**operating point**

<SVC, MVC, and MVD> independently decodable subset of a layered bitstream

Note 1 to entry: Each operating point consists of all the data needed to decode this particular bitstream subset.

Note 2 to entry: In an SVC stream an operating point represents a particular spatial resolution, temporal resolution, and quality, and can be represented either by (i) specific values of DTQ (dependency_id, temporal_id and quality_id) or (ii) specific values of P (priority_id) or (iii) combinations of them (e.g. PDTQ). Note that the usage of priority_id is defined by the application. In an SVC file a track represents one or more operating points. Within a track tiers can be used to define multiple operating points.

Note 3 to entry: The bitstream subset of an MVC or MVD operating point represents a particular set of target output views at a particular temporal resolution, and consists of all the data needed to decode this particular bitstream subset. In MVD each target output view in the bitstream subset of an MVD operating point can contain a texture view, a depth view or both.

Note 4 to entry: An operating point is referred to as an operation point in Annex H of ISO/IEC 14496-10.

3.1.34

operating point

<HEVC> independently decodable subset of a layered bitstream, where one or more layers in the set of layers are indicated to be output layers

Note 1 to entry: Each operating point consists of all the data needed to decode this particular bitstream subset.

Note 2 to entry: An operating point is referred to as an output operation point in ISO/IEC 23008-2.

3.1.35

operating point

<VVC> temporal subset of an output layer set (OLS), identified by an output layer set (OLS) index and a highest value of TemporalId

Note 1 to entry: Each operating point consists of all the data needed to decode this particular bitstream subset.

Note 2 to entry: An operating point is referred to as an operation point in ISO/IEC 23090-3.

3.1.36

output layer set

set of layers consisting of the layers of one of the specified layer sets, where one or more layers in the set of layers are indicated to be output layers, as specified in ISO/IEC 23008-2

3.1.37

parameter set

video parameter set, sequence parameter set, picture parameter set, or adaptation parameter set as defined in the applicable video standard

Note 1 to entry: This term is used to refer to all types of parameter sets.

3.1.38

parameter set elementary stream

elementary stream containing samples made up of only sequence and picture parameter set NAL units synchronized with the video elementary stream

3.1.39

picture unit

set of VCL NAL units and their associated non-VCL NAL units

Note 1 to entry: The association of VCL NAL units and non-VCL NAL units with picture units is specified in the applicable video standard.

3.1.40

prefix NAL unit

NAL units with type 14

Note 1 to entry: Prefix NAL units provide scalability information about AVC VCL NAL units and filler data NAL units. Prefix NAL units do not affect the decoding process of a legacy AVC decoder. The behaviour of a legacy AVC file reader as a response to prefix NAL units is undefined.

3.1.41

rectangular region

rectangle that does not contain holes and does not overlap with any other rectangular region of the same picture

3.1.42**reference layer**

layer that is indicated as possibly needed for decoding of another layer

Note 1 to entry: For layered HEVC, reference layers can be indicated by the 'o_inf' sample group defined in subclause 9.6.2.

3.1.43**scalable layer representation**

bitstream subset that is required for decoding the scalable layer, consisting of the scalable layer itself and all the scalable layers on which the scalable layer depends

3.1.44**sub-picture**

<SVC> proper subset of coded slices of a scalable layer representation

3.1.45**sub-picture tier**

<SVC> tier that consists of sub-pictures that are constrained so that any coded slice that is not included in the tier representation of this sub-picture tier is not referred to in inter prediction or inter-layer prediction for decoding of this sub-picture tier

3.1.46**sub-layer****temporal sub-layer**

<HEVC> set of VCL NAL units with a particular value of TemporalId and the associated non-VCL NAL units

3.1.47**sublayer****temporal sublayer**

<VVC> set of VCL NAL units with a particular value of TemporalId and the associated non-VCL NAL units

3.1.48**SVC enhancement layer**

layer that specifies a part of a scalable bitstream that enhances the video

Note 1 to entry: An SVC enhancement layer is represented by SVC VCL NAL units and the associated non-VCL NAL units and SEI messages.

Note 2 to entry: Usually an SVC enhancement layer represents a spatial or coarse-grain scalability (CGS) coding layer (identified by a specific value of dependency_id).

3.1.49**SVC NAL unit**

SVC VCL NAL unit and its associated non-VCL NAL units in an SVC stream

Note 1 to entry: The association of non-VCL NAL units with svc VCL NAL units is specified in ISO/IEC 14496-10:2020, Annex G.

3.1.50**SVC sample**

NAL units that belong to an access unit as defined in ISO/IEC 14496-10:2020, subclause 7.4.1.2, and are represented by the track

3.1.51

SVC stream

bitstream represented by the operating point for which `dependency_id` is equal to `mDid`, `temporal_id` is the greatest `temporal_id` value among `mOpSet`, and `quality_id` is the greatest `quality_id` value among `mOpSet`, where the greatest value of `dependency_id` of all the operating points represented by DTQ (`dependency_id`, `temporal_id` and `quality_id`) combinations is equal to `mDid`, and the set of all the operating points with `dependency_id` equal to `mDid` is `mOpSet`.

Note 1 to entry: The term "SVC stream" is referenced by 'decoding/accessing the entire stream' in this document. There can be NAL units that are not required for decoding this operating point.

3.1.52

SVC VCL NAL unit

NAL unit with type 20, and NAL units with type 14 when the immediately following NAL units are AVC VCL NAL units

Note 1 to entry: SVC VCL NAL units do not affect the decoding process of a legacy AVC decoder.

3.1.53

temporal layer representation representation of a temporal layer

temporal layer and all lower temporal layers

3.1.54

tier

<SVC, MVC, and MVD> set of operating points within a track, providing information about the operating points and instructions on how to access the corresponding bitstream portions (using maps and groups)

Note 1 to entry: In SVC file format a tier represents one or more scalable layers of an SVC bitstream. In HEVC and VVC, the term tier is used to represent a part of the interoperability point representation consisting of profile, tier, and level. Readers should not be confused about these two different meanings of the word "tier".

Note 2 to entry: The term "tier" is used in SVC file format to avoid confusion with the frequently used term layer. A tier represents a subset of a track and represents an operating point of an SVC bitstream. Tiers in a track subset the entire track, no matter whether the track references another track by extractors.

Note 3 to entry: An MVC or MVD tier represents a particular set of temporal subsets of a particular set of views.

3.1.55

tier representation representation of the tier

bitstream subset that is required for decoding the tier, consisting of the tier itself and all the tiers on which the tier depends

3.1.56

video elementary stream

elementary stream containing access units made up of NAL units for coded picture data

3.1.57

video and parameter set elementary stream

elementary stream containing access units with coded pictures and with parameter sets

3.1.58**video stream**

self-contained independently decodable video bitstream

3.1.59**virtual base view**

AVC compatible representation of an independently coded non-base view

Note 1 to entry: The virtual base view of an independently coded non-base view is created according to the process specified in ISO/IEC 14496-10:2020, subclause H.8.5.5. Samples containing data units of an independently coded non-base view and samples of the virtual base view are aligned by decoding times.

3.1.60**VVC bitstream**

bitstream conforming to the VVC standard (ISO/IEC 23090-3)

Note 1 to entry: Unless otherwise scoped in the text, the term VVC bitstream refers to the entire bitstream that the file writer includes in a file, possibly stored in multiple tracks. In some occasions, Clause 11 specifically refers to the VVC bitstream contained in a track, either natively or through resolving 'subp' track references or 'recr' track references.

3.1.61**VVC extraction base track**

VVC track that references another VVC track through a 'recr' track reference

3.1.62**VVC merge base track**

VVC track that references VVC subpicture tracks through a 'subp' track reference

3.1.63**VVC non-VCL track**

track that contains only non-VCL NAL units and is referred to by a VVC track through a 'vvcN' track reference

3.1.64**VVC subpicture**

subpicture as defined in the VVC standard (ISO/IEC 23090-3)

Note 1 to entry: Unless otherwise scoped in the text, the terms VVC subpicture and subpicture are used interchangeably.

3.1.65**VVC subpicture track**

track that contains either a sequence of one or more VVC subpictures forming a rectangular region or a sequence of one or more complete slices forming a rectangular region

3.1.66**VVC track**

track that represents a VVC elementary stream by including NAL units in its samples and/or sample entries, and possibly by associating other VVC tracks containing other layers and/or sublayers of the VVC elementary stream through 'vvcB' entity group and the 'vopi' sample group or through the 'opeg' entity group, and possibly by referencing VVC subpicture tracks

3.2 Abbreviated terms

3D-AVC	Three-dimensional Advanced Video Coding [refers to ISO/IEC 14496-10:2020 when the techniques in Annex J (Multiview and Depth Video with Enhanced Non-Base View Coding) are in use]
3D-HEVC	Three-dimensional High Efficiency Video Coding [refers to ISO/IEC 23008-2:2020 when the techniques in Annex I (3D High Efficiency Video Coding) are in use]
A3D	Three-dimensional Advanced Video Coding [refers to ISO/IEC 14496-10:2020 when the techniques in Annex J (Multiview and Depth Video with Enhanced Non-Base View Coding) are in use]

NOTE 1 The abbreviation A3D is used in terminology related to syntax elements and structures, whereas the abbreviation 3D-AVC is used otherwise.

ALF	Adaptive Loop Filter
APS	Adaptation Parameter Set
AU	Access Unit
AUD	Access Unit Delimiter
AVC	Advanced Video Coding. Where contrasted with SVC, MVC, or MVD in this document, this term refers to the main part of ISO/IEC 14496-10:2020, including none of Annex G (Scalable Video Coding), Annex H (Multiview Video Coding), Annex I (Multiview and Depth Video Coding), and Annex J (Multiview and Depth Video with Enhanced Non-Base View Coding)
BLA	Broken Link Access
CBR	Constant Bit Rate
CLVS	Coded Layer Video Sequence
CRA	Clean Random Access
CTU	Coding Tree Unit
CVS	Coded Video Sequence
DCI	Decoding Capability Information
DRA	Dynamic Range Adjustment
EOB	End of Bitstream
EOS	End of Sequence
EVC	Essential Video Coding
FF	File Format
GDR	Gradual Decoding Refresh
HEVC	High Efficiency Video Coding
HRD	Hypothetical Reference Decoder
ID	Identifier
IDR	Instantaneous Decoding Refresh
IRAP	Intra Random Access Point
ISOBMFF	ISO Base Media File Format (ISO/IEC 14496-12)
L-HEVC	Layered High Efficiency Video Coding
LMCS	Luma Mapping with Chroma Scaling

MVC	Multiview Video Coding [refers to ISO/IEC 14496-10:2020 when the techniques in Annex H (Multiview Video Coding) are in use]
MVCD	Multiview Video Coding Plus Depth [refers to ISO/IEC 14496-10:2020 when the techniques in Annex I (Multiview and Depth Video Coding) are in use]
MVC+D	Multiview Video Coding Plus Depth [refers to ISO/IEC 14496-10:2020 when the techniques in Annex I (Multiview and Depth Video Coding) are in use]

NOTE 2 The abbreviation MVCD is used in terminology related to syntax elements and structures, whereas the abbreviation MVC+D is used otherwise.

MV-HEVC	Multiview High Efficiency Video Coding [refers to ISO/IEC 23008-2:2020 when the techniques in Annex G (Multiview High Efficiency Video Coding) are in use]
MVD	Multiview Video Coding Plus Depth [refers to ISO/IEC 14496-10:2020 when the techniques in Annex I (Multiview and Depth Video Coding) or Annex J (Multiview and Depth Video with Enhanced Non-Base View Coding) are in use]
NAL	Network Abstraction Layer
OPI	Operating Point Information
PH	Picture Header
PPS	Picture Parameter Set
PU	Picture Unit
RADL	Random Access Decodable Leading
RASL	Random Access Skipped Leading
RBSP	Raw Byte Sequence Payload
ROI	Region-Of-Interest
SAP	Stream Access Point
SEI	Supplemental Enhancement Information
SH	Slice Header
SHVC	Scalable High efficiency Video Coding [refers to ISO/IEC 23008-2:2020 when the techniques in Annex H (Scalable High Efficiency Video Coding) are in use]
SPS	Sequence Parameter Set
STSA	Step-wise Temporal Sub-layer Access [in the context of ISO/IEC 23008-2] or Step-wise Temporal Sublayer Access [in the context of ISO/IEC 23090-3]
SVC	Scalable Video Coding [refers to ISO/IEC 14496-10:2020 when the techniques in Annex G (Scalable Video Coding) are in use]
TSA	Temporal Sub-layer Access
VCL	Video Coding Layer
VPS	Video Parameter Set
VVC	Versatile Video Coding

3.3 Conventions

Mathematical functions:

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

4 General definitions

4.1 Overview

The specifications in this clause apply to all coding systems identified by chapters in this specification, unless specifically over-ridden by definitions in the clause for a specific coding system.

4.2 Sample and configuration definition

4.2.1 General

For sample formats specified in this document, a sample contains an access unit or a part of an access unit (e.g. in a track containing a part of a multi-layer video bitstream), where an access unit is as defined in the appropriate specification.

4.2.2 Canonical order and restrictions

NOTE 1 An elementary stream is stored in the ISO Base Media File Format in a manner that a single track contains the elementary stream in the canonical stream format or a reader can reconstruct the canonical stream format from multiple tracks carrying the elementary stream. The canonical stream format is as neutral as possible so that systems that need to customize the stream for delivery over different transport protocols — MPEG-2 Systems, RTP, and so on — need not remove information from the elementary stream while being free to add to the elementary stream. Furthermore, a canonical stream format allows such operations to be performed against a known initial state.

When multiple tracks are used to store an elementary stream, as may be the case for Clauses 6, 7, 9, 10, and 11, some tracks may contain canonical streams while others may need to be processed (e.g. when extractors are used or when an implicit reconstruction of access units is needed) to produce a canonical stream.

The canonical stream format is an elementary stream that satisfies the following conditions:

- **SEI NAL units:** All SEI NAL units shall be contained in the parameter set arrays, or in the sample whose decoding time is at the time, or immediately precedes the time (with no intervening samples), when the SEI messages come into effect instantaneously. In general, SEI messages for a picture shall be included in the sample containing that picture and that SEI messages pertaining to a sequence of pictures shall be included in the sample containing the first picture of the sequence to which the SEI message pertains. The order of SEI messages within a sample is as defined in the applicable video coding standard.
- **NAL unit order:** The sequence of NAL units in an elementary stream and within a single sample shall be in a valid decoding order for those NAL units as specified in the applicable video coding standard.
- **All timing information is external to stream:** Picture Timing SEI messages that define presentation or composition timestamps may be included in the video elementary stream, as these messages contain other information than timing, and may be required for conformance checking. However, all timing information is provided by the information stored in the various sample metadata tables, and this information over-rides any timing provided in the video layer. Timing provided within the video stream in this file format should be ignored as it may contradict the timing provided by the file format and may not be correct or consistent within itself.

NOTE 2 This constraint is imposed due to the fact that post-compression editing, combination, or re-timing of a stream at the file format level can invalidate or make inconsistent any embedded timing information present within the video stream.

- **No start codes:** The elementary streams shall not include start codes. As stored, each NAL unit is preceded by a length field as specified in 4.2.3; this enables easy scanning of the sample's NAL units. Systems that wish to deliver, from this file format, a stream using start codes will need to reformat the stream to insert those start codes.

4.2.3 Sample format

4.2.3.1 Definition

This subclause defines the structure of the samples. Samples are externally framed and have a size supplied by that external framing. The syntax of a sample is configured via the decoder specific configuration for the elementary stream. An example of the structure of a video sample is depicted in Figure 1.

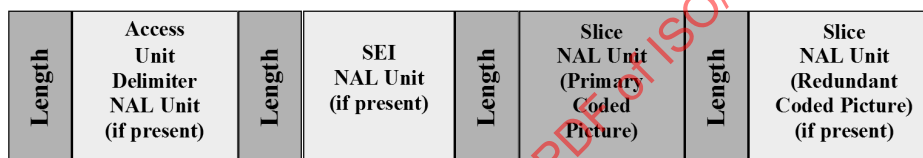


Figure 1 — Example structure of a sample

An access unit is made up of a set of NAL units. Each NAL unit is represented with a:

- *Length*: Indicates the length in bytes of the following NAL unit. The length field may be configured to be of 1, 2, or 4 bytes.
- *NAL Unit*: Contains the NAL unit data as specified in the applicable video coding standard.

4.2.3.2 Syntax

```
aligned(8) class NALUSample
{
    for (i=0; i<sample_size; )    // to end of the sample
    {
        unsigned int ((DecoderConfigurationRecord.LengthSizeMinusOne+1)*8)
            NALUnitLength;
        bit(NALUnitLength * 8) NALUnit;
        i += (DecoderConfigurationRecord.LengthSizeMinusOne+1) + NALUnitLength;
    }
}
```

4.2.3.3 Semantics

`DecoderConfigurationRecord` indicates the record in the matching sample entry (e.g. `AVCDecoderConfigurationRecord` in the case of AVC).

`NALUnitLength` indicates the size of a NAL unit measured in bytes. The length field includes the size of both the NAL header and the NAL unit payload but does not include the length field itself.

NALUnit contains a single NAL unit. The syntax of a NAL unit is defined in the appropriate specification (e.g. ISO/IEC 14496-10) and includes both the one byte NAL header and the variable length encapsulated byte stream payload.

4.2.4 Optional boxes in the sample entry

Unless otherwise explicitly specified, the optional boxes, including the BitRateBox, in the sample entry document the information for the output bitstream by the file parser reconstructed from this track and all the tracks this track depends on for decoding.

4.3 Video track structure

In the terminology of ISO/IEC 14496-12, both video and parameter set tracks are video or visual tracks. They therefore use:

- a) a handler_type of 'vide' in the HandlerBox;
- b) a video media header 'vmhd';
- c) and, as defined below, a derivative of the VisualSampleEntry.

A video stream is represented by one or more video tracks in a file.

If there is more than one track representing scalable aspects of a single stream, then they form alternatives to each other, and the field 'alternate_group' should be used, or the composition system used should select one of them, as appropriate. See ISO/IEC 14496-12:2020, subclause 8.10.3 for informative labelling of why tracks are members of alternate groups.

4.4 Template fields used

The ISO Base Media File Format defines a number of fields that have default values but that may be defined for use by specific sub-systems. Tracks containing video data may use the following template fields:

- a) alternate_group in the TrackHeaderBox (see 5.4.8 on stream switching).
- b) template field 'depth' in the VisualSampleEntry to document the presence of alpha.

depth takes one of the following values:

- 0x18 – the video sequence is in colour with no alpha
- 0x28 – the video sequence is in grayscale with no alpha
- 0x20 – the video sequence has alpha (gray or colour)

4.5 Visual width and height

The width and height fields in a VisualSampleEntry shall correctly document the maximum cropped frame dimensions in any sample of the video stream that is described by that entry.

The width and height fields do *not* reflect any changes in size caused by SEI messages such as pan-scan.

The visual handling of SEI messages such as pan-scan is both optional and terminal-dependent.

Note that the visual size in the SPS or PPS may be either frame or field size; in the sample entry, it is always the frame size.

The width and height fields in the track header may not be the same as the width and height fields in the one or more `VisualSampleEntry` in the video track. As specified in the ISO Base Media File Format, if normalized visual presentation is needed, all the sequences are normalized to the track width and height for presentation.

4.6 Decoding time (DTS) and composition time (CTS)

Samples are stored in the file format in decoding order. If picture reordering is not used and decoding and composition times are the same, then presentation is the same as decoding order and only the time-to-sample 'stts' table is used. Note that any kind of picture may be reordered, not only B-pictures.

If decoding time and composition time differ, the composition time-to-sample 'ctts' table is also used in conjunction with the 'stts' table.

4.7 Sample groups on random access recovery points 'roll' and random access points 'rap'

This version of ISO/IEC 14496-15 disallows the presence of `SampleToGroupBox` with `grouping_type` equal to 'roll' and `version` greater than 0 in tracks with sample entries specified in ISO/IEC 14496-15 other than Clause 7 and 11. Parsers according to this version of ISO/IEC 14496-15 shall ignore `SampleToGroupBox` with `grouping_type` equal to 'roll' and `version` greater than 0 in tracks with sample entries specified in ISO/IEC 14496-15 other than Clause 7 and 11.

This version of ISO/IEC 14496-15 disallows the presence of `SampleToGroupBox` with `grouping_type` equal to 'rap' and `version` greater than 0 in tracks with sample entries specified in ISO/IEC 14496-15 other than clause 7. Parsers according to this version of ISO/IEC 14496-15 shall ignore `SampleToGroupBox` with `grouping_type` equal to 'rap' and `version` greater than 0 in tracks with sample entries specified in ISO/IEC 14496-15 other than Clause 7.

NOTE 1 The video coding system can include the concept of gradual decoding refresh (GDR). For AVC and HEVC, this can be signalled in the bit-stream using the recovery point SEI message. This message is found at the beginning of the random access, and indicates how much data needs to be decoded subsequent to the access unit at the position of the SEI message before the recovery is complete. For VVC, the recovery point SEI message is not supported. Instead, the gradual decoding refresh is indicated normatively with a GDR picture that has a `ph_recovery_poc_cnt` syntax element in its picture header. The GDR picture is indicated by a GDR NAL unit type and the value of the `ph_recovery_poc_cnt` syntax element is used to identify the recovery point picture.

When all access units in output order starting from the access unit at the position of a recovery point SEI message can be successfully decoded after random access, i.e. when the `recovery_frame_cnt` syntax element of the recovery point SEI message is equal to 0, the Random Access Point ('rap') sample grouping should be used. The Random Access Point ('rap') sample grouping should not be used for VVC.

This concept of gradual recovery is supported in the file format also by using `VisualRollRecoveryEntry` Groups [4.5]. In order that the group membership marks the sample containing the recovery point SEI message, or for VVC the GDR picture, the 'roll_distance' is constrained to being only positive (i.e. a post-roll). In other words, `VisualRollRecoveryEntry` Groups can be used when the value of the `recovery_frame_cnt` syntax element of the recovery point SEI message is greater than 0 or when the value of the `ph_recovery_poc_cnt` syntax element in VVC is greater than 0.

NOTE 2 The roll-group counts samples in the file format; this does not necessarily match the way that the distances are represented in the recovery point SEI message or by the `ph_recovery_poc_cnt` syntax element in VVC.

Within a stream, it is necessary to mark the beginning of the pre-roll, so that a stream decoder may start decoding there. However, in a file, when performing random access, a deterministic search is desired for the closest preceding frame that can be decoded perfectly (either a sync sample, or the end of a pre-roll).

4.8 Hinting

Note that what the hint tracks call "B frames" are actually 'disposable' pictures or non-reference pictures, for example as defined in ISO/IEC 14496-10.

Care should be taken when the structures in Annex A (aggregators or extractors) are in use and the track is hinted. These structures are defined only for use in the file format and should not be transmitted. In particular, a hint track that points at an extractor in a video track would cause the extractor itself to be transmitted (which is probably both incorrect and not the desired behaviour), not the data the extractor references. Hint tracks should normally directly reference NAL units specified in the applicable video coding standard.

4.9 On change of sample entry (informative)

In this clause, resetting decoding can be regarded as a similar process as starting the decoding from the beginning of a video stream.

A change in the sample entry is caused, for example, by a change of parameter set(s) reusing the same parameter set ID(s) in a `VisualSampleEntry`.

For the single-layer video file formats specified in Clauses 5 and 8, the video decoder for decoding of the bitstream output from the file parser is expected to be reset at the first sample at which the sample entry changes.

For the file format specified in Clause 11, the video decoder for decoding of the bitstream output from the file parser is expected to be reset at the first sample at which the sample entry changes if the following applies.

- There is no 'dcfi' sample group, as specified in subclause 11.4.11, included in the track.

When more than one sample entry is used in a track, random accessing is enabled if a video stream is encoded and encapsulated in the track as follows:

- The first sample that a sample entry is associated with (i.e., applies to) is a sync sample.
- Decoding of samples that a sample entry is associated with does not rely on data from any sample not associated with the sample entry or data from any other sample entry.

For the multi-layer video file formats specified in Clauses 6 and 7, the video decoder for decoding of the bitstream output from the file parser is expected to be reset at the first sample in the base track or scalable base track (as defined in Clauses 6 and 7) at which the sample entry changes. When more than one sample entry is used in the base track or scalable base track, random accessing is enabled if a video stream is encoded and encapsulated in the track as follows:

- The first sample that a sample entry in the base track or scalable base track is associated with is a sync sample.
- Decoding of the set of access units from the first access unit containing the first sample in the base track associated with a sample entry to the last access unit containing the last sample in the base track associated with the sample entry, inclusive, does not rely on data from any access unit earlier than the first access unit in decoding order or data from any sample entry associated with a sample contained in an access unit earlier than the first access unit in decoding order.

NOTE When parameter sets are stored other than in the sample entry, i.e. either in a separate parameter set track, or in the samples in the video track, it is possible that there are two parameter sets with the same identifier, with the semantics (defined in the applicable video coding standard) that the second occurrence replaces the first from that time onwards in the stream. Converting such a stream to a form in which all parameter sets are in a single sample entry involves more than moving the applicable NAL units into the sample entry; one of the two occurrences will need a new identifier value, and the video NAL units that refer to it by identifier will also need updating. Alternatively, multiple sample entries can be used for the storage of parameter sets with same ID, in which case sample to sample entry association needs rewriting.

In the following, let the L-HEVC base track be the track referred to by a track reference of type 'oref' (when present) or the present track (when it contains an 'oinf' sample group). For the multi-layer video file formats specified in Clause 9, the video decoder for decoding of the bitstream output from the file parser is expected to be reset at the first sample in the L-HEVC base track at which the sample entry or the 'oinf' sample group description entry changes. When more than one sample entry or 'oinf' sample group description entry is used in the L-HEVC base track, random accessing is enabled if a video stream is encoded and encapsulated in the track as follows:

- The first sample that a sample entry or an 'oinf' sample group description entry in the L-HEVC base track is associated with has an IRAP picture (as specified in ISO/IEC 23008-2) at the lowest layer carried in the track.
- Decoding of the set of access units from the first access unit containing the first sample in the L-HEVC base track associated with a sample entry or an 'oinf' sample group description entry to the last access unit containing the last sample in the L-HEVC base track associated with the sample entry or 'oinf' sample group description entry, inclusive, does not rely on data from any access unit earlier than the first access unit in decoding order or data from any sample entry or 'oinf' sample group description entry associated with a sample contained in an access unit earlier than the first access unit in decoding order.

4.10 SEI information box

4.10.1 Definition

Box Type: 'seii'
 Container: Scheme Information box ('schi') or VisualSampleEntry
 Mandatory: Yes (in the SchemeInformationBox), No (in a VisualSampleEntry)
 Quantity: One (in the SchemeInformationBox), Zero or one (in a VisualSampleEntry)

The SEI Information box documents the SEIs in a stream. When contained in a VisualSampleEntry, numRequiredSEIs shall be 0. By inspecting the SEI Information box a player will know which SEI messages it can assume to be present, and which are deemed necessary by the file author for correct playback. There might be other SEIs present in the bitstream that are not documented by this box.

4.10.2 Syntax

```
aligned(8) class SeiInformationBox extends Box('seii') {
    unsigned int(16) numRequiredSEIs;
    for (i = 0; i < numRequiredSEIs; i++) {
        unsigned int(16) requiredSEI_ID;
    }
    unsigned int(16) numNotRequiredSEIs;
    for (i = 0; i < numNotRequiredSEIs; i++) {
        unsigned int(16) notrequiredSEI_ID;
    }
}
```

4.10.3 Semantics

requiredSEI_ID takes on the value “payloadType” of an SEI message present in the bitstream that is deemed necessary by the file author for correct playback.

notrequiredSEI_ID takes on the value “payloadType” of an SEI message present in the bitstream that is not deemed necessary by the file author for correct playback.

4.11 Post-decoder requirements scheme for signalling of SEI

4.11.1 General

In order to handle situations where the file author requires certain actions on the player or renderer, the ISO base media file format specifies the restricted-video mechanism where sample entries are hidden behind the generic sample entry 'resv'. The mechanism applies to all coding systems identified by chapters in this document. For the case of signalling of SEI, a file author can list occurring SEI message IDs (ISO/IEC 14496-10, ISO/IEC 23008-2) and classify them into two categories: those that are deemed required by the file author for correct playback, and others. The occurrence of either type of SEI messages can be signalled in the SEI Information box.

4.11.2 Definition

The scheme for signalling of SEI is defined here.

The SchemeType 'aSEI' is used.

The SEI information box is mandatory in the SchemeInformationBox under the 'aSEI' scheme. In this case, it contains information about the SEI messages present in the bitstream. Although the SEI messages are not required for decoding, the file author may require certain actions for rendering or other purposes. The box distinguishes between an SEI that is required to be understood for correct playback and an SEI that is not required for correct playback (but may enhance playback).

The SEI messages listed here should be stored either in the bitstream or in the Configuration Record. The SEI Information box does not contain the actual SEI messages, it only lists those that occur in the bitstream.

4.12 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 'subp', 'scal' or 'sabt' track reference. The value of (`flags & 1`) shall be equal to 1 in a `TrackGroupBox` of type 'alte' to indicate the uniqueness of `track_group_id` as specified in ISO/IEC 14496-12.

A 'subp', '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 VVC subpicture track selection as specified in subclause 11.6.3 or for extraction as specified in Clause A.3 and Clause A.7 or for bitstream reconstruction from tile tracks as specified in subclause 10.5.4.

NOTE Tracks in an 'alte' track group could represent different source content as long as any track in the 'alte' track group is a suitable source for AVC or HEVC extractors or for the reconstruction of a VVC access unit in a VVC merge base track. Tracks that have the same non-zero `alternate_group` value in the `TrackHeaderBox` represent the same source content. When the same 'alte' track group is referenced multiple times from a single sample, file readers are generally advised to select different tracks from this track group with different non-zero `alternate_group` values to resolve the references.

4.13 NAL unit map entry

4.13.1 Definition

The `NALUMapEntry` may be used to assign an identifier, called `groupID`, to each NAL unit. The `NALUMapEntry`, when present, shall be linked to a sample group description providing the semantics of that `groupID`. This link shall be provided by setting the `grouping_type_parameter` of the `SampleToGroupBox` of type 'nalm' to the four-character code of the associated sample grouping type. Consequently, a `SampleToGroupBox` of type 'nalm' shall never use version 0 of the box. It is not permitted to indicate an associated `grouping_type` for which the sample group description definition does not provide semantics for `groupID`.

A track shall not contain both a `SampleToGroupBox` of type 'nalm' associated with `grouping_type_parameter` equal to a particular value `groupType` and `SampleToGroupBox` of type `groupType`. When a track contains a `SampleToGroupBox` of type 'nalm' associated with `grouping_type_parameter` `groupType`, NAL units of the mapped sample are indirectly associated with the sample group description of type `groupType` through the `groupID` of the `NALUMapEntry` applicable for that sample. When a track contains a `SampleToGroupBox` of type `groupType`, each

sample is directly mapped to the sample group description of type `groupType` through the `SampleToGroupBox` of type `groupType` and all NAL units of the mapped sample are associated with the same `groupID`.

4.13.2 Syntax

```
class NALUMapEntry() extends VisualSampleGroupEntry ('nalm') {
    bit(6) reserved = 0;
    unsigned int(1) large_size;
    unsigned int(1) rle;
    if (large_size) {
        unsigned int(16) entry_count;
    } else {
        unsigned int(8) entry_count;
    }
    for (i=1; i<= entry_count; i++) {
        if (rle) {
            if (large_size) {
                unsigned int(16) NALU_start_number;
            } else {
                unsigned int(8) NALU_start_number;
            }
        }
        unsigned int(16) groupID;
    }
}
```

4.13.3 Semantics

`large_size` indicates whether the number of NAL units entries in the track samples is represented on 8 or 16 bits.

`rle` indicates whether run-length encoding is used (1) to assign `groupID` to NAL units or not (0).

`entry_count` specifies the number of entries in the map. Note that when `rle` is equal to 1, the `entry_count` corresponds to the number of runs where consecutive NAL units are associated with the same group. When `rle` is equal to 0, `entry_count` represents the total number of NAL units.

`NALU_start_number` is the 1-based NAL unit index in the sample of the first NAL unit in the current run associated with `groupID`.

`groupID` specifies the unique identifier of the group. All NAL units mapped to the same group with a particular `groupID` value have the same properties in all the sample groups that indicate a mapping to that particular `groupID` value and are contained in this track. More information about the group is provided by the sample group description entry with this `groupID` and `grouping_type` equal to the `grouping_type_parameter` of the `SampleToGroupBox` of type 'nalm'.

Each NAL-like-structure, if any, that is present in a sample is treated as follows:

- If the NAL-unit-like-structure is an aggregator, all NAL units aggregated by the aggregator by inclusion or by reference count as a single NAL unit in the indexing provided by the `NALUMapEntry`.

- Otherwise, the NAL-unit-like structure counts as one NAL unit in the indexing provided by the `NALUMapEntry`, prior to resolving or processing the NAL-unit-like structure.

When used with movie fragments, new NAL unit maps can be defined for the duration of the movie fragments, by defining a new `SampleGroupDescriptionBox` in the `TrackFragmentBox` as defined in ISO/IEC 14496-12:2020, subclause 8.9.4.

4.14 Rectangular region group entry

4.14.1 Definition

Group Type: 'trif'
 Container: `SampleGroupDescriptionBox('sgpd')`
 Mandatory: No
 Quantity: Zero or more

The `RectangularRegionGroupEntry` may be used to describe a rectangular region.

NOTE 1 The definition of a rectangular region can depend on coding systems. Clauses 10 and 11 specify the use the `RectangularRegionGroupEntry` for HEVC and VVC, respectively.

NOTE 2 The `RectangularRegionGroupEntry` used to be called `TileRegionGroupEntry`.

The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type 'trif'.

4.14.2 Syntax

```
class RectangularRegionGroupEntry() extends VisualSampleGroupEntry ('trif')
{
    unsigned int(16) groupID;
    unsigned int(1) rect_region_flag;
    if (!rect_region_flag)
        bit(7) reserved = 0;
    else {
        unsigned int(2) independent_idc;
        unsigned int(1) full_picture;
        unsigned int(1) filtering_disabled;
        unsigned int(1) has_dependency_list;
        bit(2) reserved = 0;
        if (!full_picture) {
            unsigned int(16) horizontal_offset;
            unsigned int(16) vertical_offset;
        }
        unsigned int(16) region_width;
        unsigned int(16) region_height;
        if (has_dependency_list) {
            unsigned int(16) dependency_rect_region_count;
            for (i=1; i<= dependency_rect_region_count; i++)
                unsigned int(16) dependencyRectRegionGroupID;
        }
    }
}
```

4.14.3 Semantics

`groupID` is a unique identifier for the rectangular region group described by this sample group entry. The value of `groupID` in a rectangular region group entry shall be greater than 0. The value 0 is reserved for a special use.

When there is `SampleToGroupBox` of type 'nalm' and `grouping_type_parameter` equal to 'trif', a `SampleGroupDescriptionBox` of type 'trif' shall be present, and the following applies:

- The value of `groupID` in a rectangular region group entry shall be equal to the `groupID` in one of the entries of `NALUMapEntry`.
- A NAL unit being mapped to `groupID` 0 by a `NALUMapEntry` implies that the NAL unit is required for decoding any rectangular region in the same coded picture as this NAL unit.

NOTE 1 There can be multiple rectangular region group entries with the same values of `horizontal_offset`, `vertical_offset`, `region_width` and `region_height`, respectively, but with different `groupID` values, for describing varying dependencies.

`rect_region_flag` equal to 1 specifies that the region covered by the NAL units within a picture and associated with this rectangular region group entry is a rectangular region, and further information of the rectangular region is provided by subsequent fields in this rectangular region group entry. The value 0 specifies that the region covered by the NAL units within a picture and associated with this rectangular region group entry is not a rectangular region, and no further information of the region is provided in this rectangular region group entry.

NOTE 2 The `rect_region_flag` used to be called `tile_region_flag`.

When a multi-layer bitstream is carried in one or more tracks, for any two layers `layerA` and `layerB` of the bitstream, the following constraint applies: When a NAL unit of `layerA` is associated with a `groupID` value `gldA` for which the corresponding `rect_region_flag` is equal to 1, and a NAL unit of `layerB` is associated with a `groupID` value `gldB` for which the corresponding `rect_region_flag` is equal to 1, `gldA` shall not be equal to `gldB`.

`independent_idc` specifies the coding dependencies between each rectangular region associated with this rectangular region group entry and other rectangular regions in the same picture or in reference pictures of the same layer. Inter-layer dependencies, if any, are indicated by a list of `dependencyRectRegionGroupID` (when `has_dependency_list` is equal to 1).

This field takes the following values:

- When `independent_idc` is equal to 0, the coding dependencies between this rectangular region and other rectangular regions in the same picture or in reference pictures of the same layer is either described by a list of `dependencyRectRegionGroupID` (when `has_dependency_list` is equal to 1) or unknown (when `has_dependency_list` is equal to 0).
- When `independent_idc` is equal to 1, there are no temporal dependencies between this rectangular region and the rectangular regions with different `groupID` in any reference pictures of the same layer but there can be coding dependencies between this rectangular region and a rectangular region with the same `groupID` in a reference picture of the same layer.

- When `independent_idc` is equal to 2, there are no coding dependencies between this rectangular region and any rectangular region in a reference picture of the same layer.
- The value 3 is reserved.

`full_picture`, when set, indicates that each rectangular region associated with this rectangular region group entry is a complete picture, in which case `region_width` and `region_height` shall be set to the width and height, respectively, of the complete picture, and `independent_idc` shall be set to 1 or 2.

`filtering_disabled`, when set, indicates that for each rectangular region associated with this rectangular region group entry the in-loop filtering operation does not require access to pixels adjacent to this rectangular region, i.e., bit-exact reconstruction of the rectangular region is possible without decoding the adjacent rectangular regions.

`has_dependency_list`, when set to 1, indicates that `dependency_rect_region_count` and, when `dependency_rect_region_count` is greater than 0, a list of `dependencyRectRegionGroupID` are present. When set to 0, `dependency_rect_region_count` is not present and no `dependencyRectRegionGroupID` is present.

`horizontal_offset` and `vertical_offset` give respectively the horizontal and vertical offsets of the top-left pixel of the rectangular region that is covered by the NAL units in each rectangular region associated with this rectangular region group entry, relative to the top-left pixel of the base region, in luma samples. The base region used in the `RectangularRegionGroupEntry` is the picture to which the NAL units in a rectangular region associated with this rectangular region group entry belongs.

NOTE 3 The definition of the base region can be overridden by a specific coding system. Clause 10 overrides the definition of the base region for HEVC and L-HEVC.

`region_width` and `region_height` give respectively the width and height of the rectangular region that is covered by the NAL units in each rectangular region associated with this rectangular region group entry, in luma samples.

NOTE 4 For coded streams using spatial scalability and rectangular regions on both the base and enhancement layers, when each layer is carried in its own track, the `RectangularRegionGroupEntry` sample descriptions of the base layer will give coordinates expressed in luma samples of the base layer, while the `RectangularRegionGroupEntry` sample descriptions of an enhancement layer will give coordinates expressed in luma samples of the enhancement layer.

`dependency_rect_region_count` indicates the number of rectangular regions each rectangular region associated with this rectangular region group entry depends on.

`dependencyRectRegionGroupID` gives the `groupID` of a rectangular region (as defined by a `RectangularRegionGroupEntry`) that this rectangular region depends on. For a particular rectangular region associated with this rectangular region group entry, the rectangular regions it depends on may be from the same layer or the reference layers.

4.15 Layer information sample group

4.15.1 Definition

The list of layers and sub layers a track carries is signalled in the Layer Information sample group. In the context of subclause 4.15, the term sub-layer is a collective term for the term sub-layer as defined in ISO/IEC 23008-2 and the term sublayer as defined in ISO/IEC 23090-3.

The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type `'linf'`.

4.15.2 Syntax

```
class LayerInfoGroupEntry extends VisualSampleGroupEntry ('linf') {
    bit(2) reserved = 0;
    unsigned int(6) num_layers_in_track;
    for (i=0; i<num_layers_in_track; i++) {
        bit(2) reserved = 0;
        unsigned int(1) irap_gdr_pics_in_layer_only_flag;
        unsigned int(1) completeness_flag;
        unsigned int(6) layer_id;
        unsigned int(3) min_TemporalId;
        unsigned int(3) max_TemporalId;
        bit(1) reserved = 0;
        unsigned int(7) sub_layer_presence_flags;
    }
}
```

4.15.3 Semantics

`irap_gdr_pics_in_layer_only_flag`: equal to 1 specifies that the *i*-th layer with `nuh_layer_id` equal to `layer_id` contains only IRAP pictures and GDR pictures with `ph_recovery_poc_cnt` equal to 0 and all IRAP and GDR pictures with `ph_recovery_poc_cnt` equal to 0 for the *i*-th layer are present in the track. `irap_gdr_pics_in_layer_only_flag` equal to 0 specifies that the *i*-th layer with `nuh_layer_id` equal to `layer_id` may contain pictures other than IRAP pictures or GDR pictures with `ph_recovery_poc_cnt` equal to 0.

`completeness_flag`: equal to 0 specifies that when `min_TemporalId` is equal to 0, all pictures of the *i*-th layer with `TemporalId` value equal to 0 are present in the track. `completeness_flag` equal to 1 specifies that when `min_TemporalId` is equal to 0, all pictures of the *i*-th layer with `TemporalId` value equal to 0 may not be present in the track and all pictures of the *i*-th layer with `TemporalId` value equal to 0 that are neither IRAP pictures nor GDR pictures with `ph_recovery_poc_cnt` equal to 0 are present in the track. When `min_TemporalId` is not equal to 0 for the *i*-th layer, the meaning of `completeness_flag` is unspecified for the *i*-th layer.

`num_layers_in_track`: The number of layers carried in any sample of this track associated with this sample group.

`layer_id`: `nuh_layer_id` of the layer carried in the associated samples. The instances of this field shall be in ascending order of `layer_id` in the loop.

`min_TemporalId`: The minimum `TemporalId` value for the sub-layers in the layer within the track. When `irap_gdr_pics_in_layer_only_flag` is equal to 1 for the *i*-th layer, `min_TemporalId` shall be equal to 0 for the *i*-th layer.

`max_TemporalId`: The maximum `TemporalId` value for the sub-layers in the layer within the track. When `irap_gdr_pics_in_layer_only_flag` is equal to 1 for the *i*-th layer, `max_TemporalId` shall be equal to 0 for the *i*-th layer.

`sub_layer_presence_flags`: Each bit of this field at bit position `bitPos` in the range of `min_TemporalId` to `max_TemporalId` indicates that the sub-layer with `TemporalId` equal to `bitPos` is natively present (when the bit is equal to 1) or present by extractors (when the bit is

equal to 0) in the track. Bits of this field at bit positions less than `min_TemporalId` or greater than `max_TemporalId` are unspecified.

5 AVC elementary streams and sample definitions

5.1 Overview

The Advanced Video Coding (AVC) standard (ISO/IEC 14496-10), jointly developed by the ITU-T and ISO/IEC, offers not only increased coding efficiency and enhanced robustness, but also many features for the systems that use it. To enable the best visibility of, and access to, those features, and to enhance the opportunities for the interchange and interoperability of media, this clause defines a storage format for video streams compressed using AVC.

This clause defines the storage for plain AVC streams, where 'plain AVC' refers to the main part of ISO/IEC 14496-10, excluding any multi-layer extension of ISO/IEC 14496-10:2020 such as Annex G (Scalable Video Coding), Annex H (Multiview Video Coding), Annex I (Multiview and Depth Video Coding), and Annex J (Multiview and Depth Video with Enhanced Non-Base View Coding).

This clause specifies the elementary stream and sample structure used to store AVC visual content.

The storage of AVC content uses the existing capabilities of the ISO base media file format but also defines extensions to support the following features of the AVC codec.

- a) Switching pictures enable switching between different coded streams and substitution of pictures within the same stream.
- b) Sub-sequences and layers provide a structuring of the dependencies of a group of pictures to provide for a flexible stream structure (e.g. in terms of temporal scalability and layering).
- c) The sequence and picture parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded macroblock data. Each slice containing the coded macroblock data references the picture parameter set containing its decoding parameters. In turn, the picture parameter set references a sequence parameter set that contains sequence level decoding parameter information.

5.2 Elementary stream structure

Two types of elementary streams are defined for storing AVC content (see also Figure 2):

- **A video elementary stream** contains all video coding related NAL units (i.e. those NAL units containing video data or signalling video structure) and may contain non-video coding related NAL units such as SEI messages and access unit delimiter NAL units. Aggregators and Extractors, when present, shall not be directly output by file parsers.
- **Parameter set elementary streams** shall not contain video coding related NAL units (i.e. those NAL units containing video data or signalling video structure), and would normally contain only sequence parameter sets, picture parameter sets and sequence parameter set extension NAL units.

Using these stream types, AVC content shall be stored in one of these configurations:

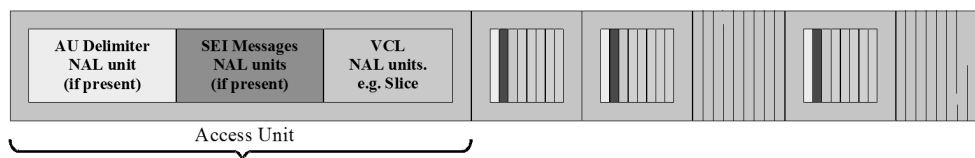
- **Video elementary stream with no parameter sets:** In this case, sequence and picture parameter set NAL units shall be stored in the sample entries of this track. Sequence and picture parameter set NAL units shall not be part of AVC samples within the stream itself.
- **Video elementary stream possibly including parameter sets:** In this case, the sample entry indicates whether the stream may contain parameter sets of given types, in addition to other parameters provided in the sample entry. Sequence and picture parameter set NAL units may therefore be part of AVC samples within the stream itself.
- **Video elementary stream and parameter set elementary stream:** In this case, sequence and picture parameter set NAL units shall be transmitted only in the parameter set elementary stream and shall neither be present in the sample entries nor the AVC samples of the video elementary stream.

The types of NAL units that are allowed in each of the video and parameter set elementary streams are specified in Table 1.

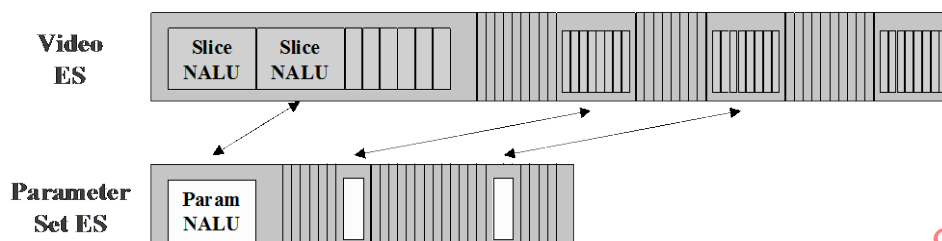
Table 1 — NAL Unit types in elementary Streams

Value of nal_unit_type	Description	Video elementary stream (sample entry 'avc1' or 'avc2')	Video elementary stream (sample entry 'avc3' or 'avc4')	Parameter set elementary stream
0	Unspecified	See Annex F	See Annex F	See Annex F
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp()	Yes	Yes	No
2	Coded slice data partition A slice_data_partition_a_layer_rbsp()	Yes	Yes	No
3	Coded slice data partition B slice_data_partition_b_layer_rbsp()	Yes	Yes	No
4	Coded slice data partition C slice_data_partition_c_layer_rbsp()	Yes	Yes	No
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp()	Yes	Yes	No
6	Supplemental enhancement information(SEI) sei_rbsp()	Yes. Except for the Sub- sequence, layering or Filler SEI messages	Yes Except for the Sub- sequence, or layering SEI messages	Only 'declarative' SEIs should be present
7	Sequence parameter set (SPS) seq_parameter_set_rbsp()	No. If parameter set elementary stream is not used, SPS shall be stored	Yes Parameter set elementary stream shall not be used	Yes

		in the Decoder Specific Information.		
8	Picture parameter set (PPS) pic_parameter_set_rbsp()	No. If parameter set elementary stream is not used, PPS shall be stored in the Decoder Specific Information.	Yes Parameter set elementary stream shall not be used	Yes
9	Access unit delimiter (AU Delimiter) access_unit_delimiter_rbsp()	Yes	Yes	No
10	End of sequence end_of_seq_rbsp()	Yes	Yes	No
11	End of stream end_of_stream_rbsp()	Yes	Yes	No
12	Filler data (FD) filler_data_rbsp()	No	Yes	No
13	Sequence parameter set extension seq_parameter_set_extension_rbsp()	No. If parameter set elementary stream is not used, Sequence Parameter Set Extension shall be stored in the Decoder Specific Information.	Yes Parameter set elementary stream shall not be used	Yes
14...18	Reserved	Not specified by this document	Not specified by this document	Not specified by this document
19	Coded slice of an auxiliary coded picture without partitioning slice_layer_without_partitioning_rbsp()	Yes	Yes	No
20...23	Reserved	Not specified by this document	Not specified by this document	Not specified by this document
24 – 31	Unspecified	See Annex F	See Annex F	See Annex F



(a) Single video elementary stream containing NAL units



(b) Synchronized video and parameter sets with arrows denoting synchronization between streams

Figure 2 — AVC elementary stream structure

5.3 Sample and configuration definition

5.3.1 Canonical order and restrictions

The canonical stream format is an AVC elementary stream that satisfies the following conditions in addition to the general conditions in 4.2.2:

- Video data NAL units (Coded Slice, Coded Slice Data Partition A, Coded Slice Data Partition B, Coded Slice Data Partition C, Coded Slice IDR Pictures): All slice and data partition NAL units for a single picture shall be contained within the sample whose decoding time and composition time are those of the picture. Each sample shall contain at least one VCL NAL unit of the primary coded picture.
- **Parameter sets:** If a parameter set elementary stream is used, then the sample in the parameter stream shall have a decoding time equal or prior to when the parameter set(s) comes into effect instantaneously.

NOTE 1 This means that for a parameter set to be used in a picture it is to be sent prior to the sample containing that picture or in the sample for that picture.

NOTE 2 When the sample entry name is 'avc1' or 'avc2', parameter sets are stored either in the sample entries of the video stream or in the parameter set stream, but never in both. This ensures that it is not necessary to examine every part of the video elementary stream to find relevant parameter sets. It also avoids dependencies of indefinite duration between the sample that contains the parameter set definition and the samples that use it. Storing parameter sets in the sample entries of a video stream provides a simple and static way to supply parameter sets. Parameter set elementary streams on the other hand are more complex but allow for more dynamism in the case of updates. Parameter sets can be inserted into the video elementary stream when the file is streamed over a transport that permits such parameter set updates. When the sample entry name is 'avc3' or 'avc4', then parameter sets can be present both the sample entries and as part of the samples.

- **Parameter set track:** A sync sample in a parameter set track indicates that all parameter sets needed from that (decoding) time forward in the video elementary stream are in that or

succeeding parameter stream samples. Also there shall be an AVC parameter set sample at each point a parameter set is updated. Each AVC parameter set sample shall contain exactly the sequence and picture parameter sets needed to decode the relevant section of the video elementary stream.

NOTE 3 The use of a parameter set track in the file format does not require that a system delivering video content use a separate elementary stream for parameter sets. Instead, implementations can choose to map parameter sets to in-band parameter set NAL units in the video elementary stream or use some out-of-band delivery mechanism defined by the transport layer.

- **SEI message NAL units:** The order of SEI messages within a sample is as defined in ISO/IEC 14496-10:2020, subclause 7.4.1.2.
- **Access unit delimiter NAL units:** The constraints obeyed by access unit delimiter NAL units are defined in ISO/IEC 14496-10:2020, subclause 7.4.1.2.3.
- **Sub-sequence and layering SEI messages.** Sub-sequence or layering SEI messages should not occur in the AVC elementary stream. Specifically, the sub-sequence information, sub-sequence layer characteristics, and sub-sequence characteristics SEI messages should not occur in the stored AVC elementary stream. Instead, all such information is stored as external metadata as described in 5.4.7.
- **Redundant picture:** NAL units within a single access unit shall be ordered in non-decreasing order of redundant picture count (`redundant_pic_cnt`).
- **Slice groups:** NAL units within a primary coded picture or a redundant coded picture shall be ordered in non-decreasing order of slice group identifier. Within the same slice group, slices shall be ordered by their first Macroblock location (`first_mb_in_slice` in the slice header).

NOTE 4 Slice groups are stored in a canonical order to ease hinting, and to make it easier to find a primary coded picture within a sample.

- **Filler data.** Video data is naturally represented as variable bit rate in the file format and should be filled for transmission if needed. Filler Data NAL units and Filler Data SEI messages shall not be present in the file format stored stream when the sample entry does not also permit parameter sets.

NOTE 5 The removal or addition of Filler Data NAL units, start codes, SEI messages or Filler Data SEI messages can change the bit-stream characteristics with respect to conformance with the HRD when operating the HRD in CBR mode as specified in ISO/IEC 14496-10:2020, Annex C.

5.3.2 Decoder configuration information

5.3.2.1 AVC decoder configuration record

5.3.2.1.1 Definition

This record contains the size of the length field used in each sample to indicate the length of its contained NAL units as well as the initial parameter sets. This record is externally framed (its size is supplied by the structure that contains it).

This record contains a version field. This version of the specification defines version 1 of this record. Incompatible changes to the record will be indicated by a change of version number. Readers shall not attempt to decode this record or the streams to which it applies if the version number is unrecognized.

Compatible extensions to this record will extend it and will not change the configuration version code. Readers should be prepared to ignore unrecognized data beyond the definition of the data they understand (e.g. after the parameter sets in this document).

When used to provide the configuration of

- a parameter set elementary stream,
- a video elementary stream used in conjunction with a parameter set elementary stream,

the configuration record shall contain no sequence or picture parameter sets (`numOfSequenceParameterSets` and `numOfPictureParameterSets` shall both have the value 0).

When used to provide the configuration of a video elementary stream used without a parameter set elementary stream, the configuration record may or may not contain sequence or picture parameter sets (`numOfSequenceParameterSets` or `numOfPictureParameterSets` may or may not have the value 0).

The values for `AVCProfileIndication`, `AVCLevelIndication`, and the flags that indicate profile compatibility shall be valid for all parameter sets of the stream described by this record. The level indication shall indicate a level of capability equal to or greater than the highest level indicated in the included parameter sets; each profile compatibility flag may only be set if all the included parameter sets set that flag. The profile indication shall indicate a profile to which the entire stream associated with this configuration record conforms. If the sequence parameter sets are marked with different profiles, and the relevant profile compatibility flags are all zero, then the stream may need examination to determine which profile, if any, the entire stream conforms to. If the entire stream is not examined, or the examination reveals that there is no profile to which the entire stream conforms, then the stream shall be split into two or more sub-streams with separate configuration records in which these rules are met.

Explicit indication can be provided in the AVC Decoder Configuration Record about the chroma format and bit depth used by the AVC video elementary stream. The parameter '`chroma_format_idc`' present in the sequence parameter set in AVC specifies the chroma sampling relative to the luma sampling. Similarly the parameters '`bit_depth_luma_minus8`' and '`bit_depth_chroma_minus8`' in the sequence parameter set specify the bit depth of the samples of the luma and chroma arrays. The values of `chroma_format_idc`, `bit_depth_luma_minus8` and '`bit_depth_chroma_minus8`' shall be identical in all sequence parameter sets in a single AVC configuration record. If two sequences differ in any of these values, two different AVC configuration records will be needed. If the two sequences differ in color space indications in their VUI information, then two different configuration records are also required.

The array of sequence parameter sets, and the array of picture parameter sets, may contain SEI messages of a 'declarative' nature, that is, those that provide information about the stream as a whole. An example

of such an SEI is a user-data SEI. Such SEIs may also be placed in a parameter set elementary stream. NAL unit types that are reserved in ISO/IEC 14496-10 and in this document may acquire a definition in future, and readers should ignore NAL units with reserved values of NAL unit type when they are present in these arrays.

NOTE This 'tolerant' behaviour is designed so that errors are not raised, allowing the possibility of backwards-compatible extensions to these arrays in future specifications.

When Sequence Parameter Set Extension NAL units occur in this record with `AVCProfileIndication` equal to 66, 77, or 88, they should be placed in the Sequence Parameter Set Array.

5.3.2.1.2 Syntax

```
aligned(8) class AVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) AVCProfileIndication;
    unsigned int(8) profile_compatibility;
    unsigned int(8) AVCLevelIndication;
    bit(6) reserved = '111111'b;
    unsigned int(2) lengthSizeMinusOne;
    bit(3) reserved = '111'b;
    unsigned int(5) numOfSequenceParameterSets;
    for (i=0; i< numOfSequenceParameterSets; i++) {
        unsigned int(16) sequenceParameterSetLength;
        bit(8*sequenceParameterSetLength) sequenceParameterSetNALUnit;
    }
    unsigned int(8) numOfPictureParameterSets;
    for (i=0; i< numOfPictureParameterSets; i++) {
        unsigned int(16) pictureParameterSetLength;
        bit(8*pictureParameterSetLength) pictureParameterSetNALUnit;
    }
    if( AVCProfileIndication != 66 && AVCProfileIndication != 77 &&
        AVCProfileIndication != 88 )
    {
        bit(6) reserved = '111111'b;
        unsigned int(2) chroma_format;
        bit(5) reserved = '11111'b;
        unsigned int(3) bit_depth_luma_minus8;
        bit(5) reserved = '11111'b;
        unsigned int(3) bit_depth_chroma_minus8;
        unsigned int(8) numOfSequenceParameterSetExt;
        for (i=0; i< numOfSequenceParameterSetExt; i++) {
            unsigned int(16) sequenceParameterSetExtLength;
            bit(8*sequenceParameterSetExtLength) sequenceParameterSetExtNALUnit;
        }
    }
}
```

5.3.2.1.3 Semantics

`AVCProfileIndication` contains the profile code as defined in ISO/IEC 14496-10.

`profile_compatibility` is a byte defined exactly the same as the byte that occurs between the `profile_IDC` and `level_IDC` in a sequence parameter set (SPS), as defined in ISO/IEC 14496-10.

`AVCLevelIndication` contains the level code as defined in ISO/IEC 14496-10.

`lengthSizeMinusOne` indicates the length in bytes of the `NALUnitLength` field in an AVC video sample or AVC parameter set sample of the associated stream minus one. For example, a size of

one byte is indicated with a value of 0. The value of this field shall be one of 0, 1, or 3 corresponding to a length encoded with 1, 2, or 4 bytes, respectively.

`numOfSequenceParameterSets` indicates the number of SPSs that are used as the initial set of SPSs for decoding the AVC elementary stream.

`sequenceParameterSetLength` indicates the length in bytes of the SPS NAL unit as defined in ISO/IEC 14496-10.

`sequenceParameterSetNALUnit` contains a SPS NAL unit, as specified in ISO/IEC 14496-10.

SPSs shall occur in order of ascending parameter set identifier with gaps being allowed.

`numOfPictureParameterSets` indicates the number of picture parameter sets (PPSs) that are used as the initial set of PPSs for decoding the AVC elementary stream.

`pictureParameterSetLength` indicates the length in bytes of the PPS NAL unit as defined in ISO/IEC 14496-10.

`pictureParameterSetNALUnit` contains a PPS NAL unit, as specified in ISO/IEC 14496-10.

PPSs shall occur in order of ascending parameter set identifier with gaps being allowed.

`chroma_format` contains the `chroma_format` indicator as defined by the `chroma_format_idc` parameter in ISO/IEC 14496-10.

`bit_depth_luma_minus8` indicates the bit depth of the samples in the Luma arrays. For example, a bit depth of 8 is indicated with a value of zero ($\text{BitDepth} = 8 + \text{bit_depth_luma_minus8}$). The value of this field shall be in the range of 0 to 4, inclusive.

`bit_depth_chroma_minus8` indicates the bit depth of the samples in the Chroma arrays. For example, a bit depth of 8 is indicated with a value of zero ($\text{BitDepth} = 8 + \text{bit_depth_chroma_minus8}$). The value of this field shall be in the range of 0 to 4, inclusive.

`numOfSequenceParameterSetExt` indicates the number of Sequence Parameter Set Extensions that are used for decoding the AVC elementary stream.

`sequenceParameterSetExtLength` indicates the length in bytes of the SPS Extension NAL unit as defined in ISO/IEC 14496-10.

`sequenceParameterSetExtNALUnit` contains a SPS Extension NAL unit, as specified in ISO/IEC 14496-10.

5.4 Derivation from ISO base media file format

5.4.1 AVC file type and identification

Conformance with this document is indicated by the presence of the brand of a specification that permits the inclusion of AVC content, in the compatible brands list of the `FileTypeBox` as defined in ISO/IEC 14496-12. The file extension normally matches the major brand.

AVC content may be used in an MPEG-4 context; in a file with extension “.mp4”, the major brand may be ‘avc1’.

Readers conformant to this document should read the file if a suitable brand occurs in the compatible-brands list. Other structures and/or track types, defined in specifications other than that identified by the brand, may be present, and these may be ignored by a reader conformant with the specification identified by the brand.

5.4.2 AVC video stream definition

5.4.2.1 Sample entry name and format

5.4.2.1.1 Definition

Sample Entry and Box Types: 'avc1', 'avc2', 'avc3', 'avc4', 'avcC', 'm4ds', 'btrt'

Container: Sample Description Box ('stsd')

Mandatory: An 'avc1', 'avc2', 'avc3' or 'avc4' sample entry is mandatory

Quantity: One or more sample entries may be present

An AVC visual sample entry shall contain an AVC Configuration Box, as defined below. This includes an AVCDerivedConfigurationRecord, as defined in 5.3.2.1.

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

Multiple sample entries may be used, as permitted by the ISO Base Media File Format specification, to indicate sections of video that use different configurations or parameter sets.

The sample entry name 'avc1' or 'avc3' may only be used when the stream to which this sample entry applies is a compliant and AVC stream as viewed by an AVC decoder operating under the configuration (including profile and level) given in the AVConfigurationBox. The file format specific structures that resemble NAL units (see Annex A) may be present but shall not be used to access the AVC base data; that is, the AVC data shall not be contained in Aggregators (though they may be included within the bytes referenced by the additional_bytes field) nor referenced by Extractors.

The sample entry name 'avc2' or 'avc4' may only be used when Extractors or Aggregators (Annex A) are required to be supported, and an appropriate Toolset is required (for example, as indicated by the file-type brands). This sample entry type indicates that, in order to form the intended AVC stream, Extractors shall be replaced with the data they are referencing, and Aggregators shall be examined for contained NAL Units. The syntax and semantics of Extractors and Aggregators are specified in Annex A. Tier grouping may be present.

5.4.2.1.2 Syntax

```
// Visual Sequences
class AVCCConfigurationBox extends Box('avcC') {
    AVCDecoderConfigurationRecord() AVCConfig;
}

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

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

class AVC2SampleEntry() extends VisualSampleEntry (type) {
    // type is 'avc2' or 'avc4'
    AVCCConfigurationBox avcconfig;
    MPEG4ExtensionDescriptorsBox descr; // optional
}
```

5.4.2.1.3 Semantics

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\012AVC Coding" being recommended; the first byte is a count of the remaining bytes, here represented by \012, which (being octal 12) is 10 (decimal), the number of bytes in the rest of the string.

config is defined in 5.3.2. If a separate parameter set stream is used, numOfSequenceParameterSets and numOfPictureParameterSets shall both be zero.

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.

5.4.3 AVC parameter set stream definition

5.4.3.1 Sample entry name and format

5.4.3.1.1 Definition

Sample Entry Type: 'avcp'
 Container: Sample Description Box ('stsd')
 Mandatory: Yes
 Quantity: One or more sample entries may be present

An AVC parameter stream sample entry shall contain an AVC Parameter Stream Configuration Box, as defined below.

5.4.3.1.2 Syntax

```
class AVCPParameterSampleEntry() extends VisualSampleEntry ('avcp'){
    AVCConfigurationBox config;
}
```

5.4.3.1.3 Semantics

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\016AVC Parameters" being recommended (\016 is 14, the length of the string as a byte).

config is defined in 5.3.2. numOfSequenceParameterSets and numOfPictureParameterSets shall both be zero.

5.4.3.2 Sample format

5.4.3.2.1 Definition

This subclause defines the sample format for AVC Parameter set streams. An AVC parameter set sample contains only one or more sequence, picture parameter set, or sequence parameter set extension NAL units.

5.4.3.2.2 Syntax

```
aligned(8) class AVCPParameterSample
{
    unsigned int PictureLength = sample_size;
    //Size of AVCPParameterSample from SampleSizeBox
    for (i=0; i<PictureLength; ) // to end of the picture
    {
        unsigned int ((AVCDecoderConfigurationRecord.LengthSizeMinusOne+1)*8)
            NALUnitLength;
        bit(NALUnitLength * 8) NALUnit;
        i += (AVCDecoderConfigurationRecord.LengthSizeMinusOne+1) + NALUnitLength;
    }
}
```

5.4.3.2.3 Semantics

NALUnitLength indicates the size of a NAL unit measured in bytes. The length field includes the size of both the one byte NAL header and the EBSP payload but does not include the length field itself.

NALUnit contains a single NAL unit. The syntax of a NAL unit is defined in ISO/IEC 14496-10 and includes both the one byte NAL header and the variable length encapsulated byte stream payload.

5.4.3.3 Track reference

A track reference of type 'avcp' in the video elementary stream track reference table, referencing the parameter set stream, is used to connect from the video elementary stream to the parameter set elementary stream.

5.4.4 Parameter sets

This subclause applies when a separate parameter set stream is not used.

Each AVC sample entry, which contains the AVC video stream decoder specific information, includes a group of SPSs and PPSs. This group of parameter sets functions much like a codebook. Each parameter set has an identifier, and each slice references the parameter set it was coded against using the parameter set's identifier.

When the sample entry name is 'avc1' or 'avc2', the following applies:

- In the file format each configuration of parameter sets is represented separately. A parameter set cannot be updated without causing a different sample entry to be used. For example, suppose that samples 1 to 4 use PPSs identified as 1, 2, 3 and a single SPS identified as 1. At sample 5 a new value of PPS 2 is required but PPSs 1 and 3 remain unaltered and are used until sample 10. In this case, the sample entry for samples 1 through 4 is the same and contains the initial values of PPSs 1, 2, 3 and SPS 1. At sample 5 the sample entry needs to change to a second sample entry, which contains the updated value for PPS 2 *as well as* the original values of PPSs 1 and 3 and SPS 1. This second sample entry is used for samples 5 through 10.
- Systems wishing to send SPS or PPS updates will need to compare the two configurations to find the differences in order to send the appropriate parameter set updates.

NOTE 1 When several parameter sets are used and parameter set updating is desired, a separate parameter set elementary stream can be used.

NOTE 2 Decoders conforming to this document need to support both parameter sets stored in separate elementary streams as well as parameter sets stored in the AVC sample entries, unless restricted by another specification using this one.

When the sample entry name is 'avc3' or 'avc4', parameter sets may be present in both sample entry and as part of samples, and an update of a parameter set by a parameter set of the same type that is stored as part of a sample is possible.

5.4.5 Sync sample

A sample is considered as a sync sample if the video data NAL units in the sample indicate that the primary coded picture contained in the sample is an instantaneous decoding refresh (IDR) picture.

When the sample entry name is 'avc1' or 'avc2', all SPSs and PPSs needed to decode the video data NAL units in the sample of the IDR picture and the following samples in decode order are contained in the decoder configuration of the video elementary stream or in a separate parameter set elementary stream sample.

When the sample entry name is 'avc3' or 'avc4', the following applies:

- a) If the sample is a sync sample, all parameter sets needed for decoding the sample shall be included either in the sample entry or in the sample itself.
- b) Otherwise (the sample is not a sync sample), all parameter sets needed for decoding the same 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 parameter set elementary stream sample is a sync sample if and only if all parameter sets required by the associated video elementary stream from the time of the AVC parameter set sample forward are supplied, in the parameter set stream, before they are required by the associated video elementary stream.

5.4.6 Shadow sync

The use of the shadow sync table to indicate alternate encodings of a sample for random access are supported as defined in the ISO Base Media File Format. A shadow sync shall indicate a sample that is a random access point as specified in the general requirements and for the specific coding format in the track.

While the use of shadow sync is supported for backward compatibility reasons, this use is deprecated and use of the mechanisms defined in 5.4.8 is recommended.

5.4.7 Layering and sub-sequences

5.4.7.1 Overview

Streams may be constructed so that the referential dependencies between samples allow only subsets of the samples to be sent to the decoder. This mechanism is called *thinning* a stream. Thinning discards entire sets of samples using knowledge of what other sets of pictures this set of pictures depends on and what picture sets in turn depend on it.

The referential dependencies between samples in a stream are structured into *layers* and *sub-sequences*. Samples in higher layers only depend on samples in lower layers. Layers are numbered, and the samples are organized such that a sample in layer *N* has no dependencies on samples in layers greater than *N*.

Sub-sequences are as defined in ISO/IEC 14496-10:2020, Annex D. Dependency relations between sub-sequences represent the dependency structure of a stream. Each sub-sequence belongs to one and only one layer. A sample shall reside in one layer and in one sub-sequence only.

Layering and sub-sequence information is represented in the file format to allow systems reading the files to understand the ways in which stream thinning may be achieved without having to examine the dependency structure of every sample.

Layer and sub-sequences are represented in the AVC file format as Sample Group. An AVC file shall contain zero or one instance of a SampleToGroupBox (per track) with a `grouping_type` equal to 'avll'. This SampleToGroupBox instance represents the assignment of samples in a track to layers. An accompanying instance of the SampleGroupDescriptionBox with the same grouping type shall, if it exists, contain AVCLayerEntry sample group entries describing the layers. Similarly, an AVC file shall contain zero or one instance of a SampleToGroupBox (per track) with a `grouping_type` equal to 'avss'. This SampleToGroupBox instance represents the assignment of samples in a track to sub-sequences. An accompanying instance of the SampleGroupDescriptionBox with the same grouping type shall, if it exists, contain AVCSubSequenceEntry sample group entries describing the sub-sequences.

5.4.7.2 Sub-sequence description entry

5.4.7.2.1 Definition

Group Type: 'avss'
 Container: Sample Group Description Box ('sgpd')
 Mandatory: No
 Quantity: Zero or more.

A sub-sequence description entry is a sample group entry that describes a sub-sequence. A sub-sequence is a set of samples in a track belonging to the same layer. A sub-sequence depends on another sub-sequence if and only if there exists a sample in the sub-sequence that is directly referentially dependent on some sample in the other sub-sequence. All samples in a sub-sequence shall directly depend only on (i.e., refer to) other samples within the same sub-sequence or samples in the sub-sequences on which is it dependent. A sub-sequence depends on zero or more sub-sequences in the lower layers. A sub-sequence shall not depend on any other sub-sequence in the same or higher layer.

At most one partition of an AVC stream into layers shall exist in the AVC file format; that is, there is either zero or one instances of the sample group boxes (SampleToGroupBox, SampleGroupDescriptionBox) per track concerning the grouping of samples into layers and sub-sequences.

The grouping_type_parameter is not defined for the SampleToGroupBox with grouping type 'avss'.

5.4.7.2.2 Syntax

```
aligned(8) class DependencyInfo
{
    unsigned int(8)    subSeqDirectionFlag;
    unsigned int(8)    layerNumber;
    unsigned int(16)   subSequenceIdentifier;
}

class AVCSubSequenceEntry() extends VisualSampleGroupEntry('avss')
{
    unsigned int(16) subSequenceIdentifier;
    unsigned int(8)  layerNumber;
    unsigned int(1)  durationFlag;
    unsigned int(1)  avgRateFlag;
    unsigned int(6)  reserved = 0;
    if (durationFlag)
        unsigned int(32) duration;
    if (avgRateFlag)
    {
        unsigned int(7)  reserved = 0;
        unsigned int(1)  accurateStatisticsFlag;
        unsigned int(16) avgBitRate;
        unsigned int(16) avgFrameRate;
    }
    unsigned int(8) numReferences;
    DependencyInfo dependency[numReferences];
}
}
```

5.4.7.2.3 Semantics

`subSeqDirectionFlag`, `layerNumber` and `subSequenceIdentifier` within the `DependencyInfo` class identify a sub-sequence that is used as a reference for this sub-sequence. Only direct, not indirect, referential dependencies shall be identified. The identified sub-sequence has sub-sequence identifier equal to `subSequenceIdentifier` and resides in the layer having the layer number equal to `layerNumber`. If `subSeqDirectionFlag` is 0, the sub-sequence used as a reference for this sub-sequence is the closest sub-sequence among all the candidate sub-sequences whose first picture precedes the first picture of this sub-sequence in decoding order and that resides in the indicated layer and has the indicated sub-sequence identifier; 'closest' means that among all the candidate sub-sequences the first picture of the referenced sub-sequence is the closest to the first picture of this sub-sequence in decoding order. If `subSeqDirectionFlag` is equal to 1, the sub-sequence used as a reference for this sub-sequence is the closest sub-sequence among all the candidate sub-sequences whose first picture succeeds the first picture of this sub-sequence in decoding order and that resides in the indicated layer and has the indicated sub-sequence identifier; 'closest' has the same meaning as above.

`subSequenceIdentifier` gives the identifier for the sub-sequence.

`layerNumber` gives the layer number to which the sub-sequence belongs.

`durationFlag` equal to 0 indicates that the duration of the target sub-sequence is not specified.

Otherwise, a value of 1 indicates that the `duration` field indicates the duration of this sub-sequence.

`avgRateFlag` equal to 0 indicates that the average bit rate and the average frame rate of the target sub-sequence are unspecified. Otherwise, a value of 1 indicates that the average rate characteristics are described by the `accurateStatisticsFlag`, `avgBitRate`, and `avgFrameRate` fields.

`duration` indicates the duration of the target sub-sequence in clock ticks of a 90-kHz clock.

`accurateStatisticsFlag` indicates how reliable the values of `avgBitRate` and `avgFrameRate` are. `accurateStatisticsFlag` equal to 1 indicates that `avgBitRate` and `avgFrameRate` are rounded from statistically correct values. `accurateStatisticsFlag` equal to 0 indicates that `avgBitRate` and `avgFrameRate` are estimates and may deviate somewhat from the correct values.

`avgBitRate` gives the average bit rate in (1000 bits)/second of this sub-sequence. All NAL units of this sub-sequence are taken into account in the calculation. In the following, B is the number of bits in all NAL units in the sub-sequence. t_1 is the decoding timestamp of the first picture of the sub-sequence (in decoding order), and t_2 is the decoding timestamp of the last picture of the sub-sequence (in decoding order). Then, the `avgBitRate` is calculated as follows provided that $t_1 \neq t_2$: $\text{avgBitRate} = \text{round}(B \div ((t_2 - t_1) * 1000))$. If $t_1 = t_2$, `avgBitRate` shall be 0.

`avgFrameRate` gives the average frame rate in units of frames/(256 seconds) of this sub-sequence. All NAL units of this sub-sequence are taken into account in the calculation. The average frame rate is calculated according to the presentation timestamp of the frame. In the following, C is the number of frames in the sub-sequence. t_1 is the presentation timestamp of the first picture of the sub-sequence (in decoding order), and t_2 is the presentation timestamp (in seconds) of the last picture of the sub-sequence (in decoding order). Then, the `avgFrameRate` is calculated as follows provided that $t_1 \neq t_2$: $\text{avgFrameRate} = \text{round}(C * 256 \div (t_2 - t_1))$. If $t_1 = t_2$, `avgFrameRate` shall be 0. Value zero indicates an unspecified frame rate.

`numReferences` gives the number of sub-sequences directly referenced in this sub-sequence. `dependency` is an array of `DependencyInfo` structures giving the identifying referenced sub-sequences.

5.4.7.3 Layer description entry

5.4.7.3.1 Definition

Group Type: 'av11'
 Container: Sample Group Description Box ('sgpd')
 Mandatory: No
 Quantity: Zero or more.

A layer sample group entry defines the layer information for all samples in a layer. Layers are numbered with non-negative integers. Layers are ordered hierarchically based on their dependency on each other: A layer having a larger layer number is a higher layer than a layer having a smaller layer number. The layers are ordered hierarchically based on their dependency on each other so that a layer does not depend on any higher layer and may depend on lower layers. The lowest layer is numbered as zero and other layers are given consecutive numbers. In other words, layer 0 is independently decodable, pictures in layer 1 may be predicted from layer 0, pictures in layer 2 may be predicted from layers 0 and 1, etc.

The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type 'av11'.

5.4.7.3.2 Syntax

```
class AVCLayerEntry() extends VisualSampleGroupEntry ('av11')
{
    unsigned int(8)    layerNumber;
    unsigned int(7)    reserved = 0;
    unsigned int(1)    accurateStatisticsFlag;
    unsigned int(16)   avgBitRate;
    unsigned int(16)   avgFrameRate;
}
```

5.4.7.3.3 Semantics

`layerNumber` gives the number of this layer with the base layer being numbered as zero and all enhancement layers being numbered as one or higher with consecutive numbers.

`accurateStatisticsFlag` indicates how reliable the values of `avgBitRate` and `avgFrameRate` are. `accurateStatisticsFlag` equal to 1 indicates that `avgBitRate` and `avgFrameRate` are rounded from statistically correct values. `accurateStatisticsFlag` equal to 0 indicates that `avgBitRate` and `avgFrameRate` are estimates and may deviate somewhat from the correct values.

`avgBitRate` gives the average bit rate in units of 1000 bits per second. All NAL units in this and lower sub-sequence layers are taken into account in the calculation. The average bit rate is calculated according to the decoding timestamp. In the following, B is the number of bits in all NAL units in this and lower sub-sequence layers. t_1 is the decoding timestamp of the first picture in this and lower sub-sequence layers in the presentation order, and t_2 is the decoding timestamp of the latest picture in this and lower sub-sequence layers in the presentation order. Then, `avgBitRate` is calculated as follows provided that $t_1 \neq t_2$: $\text{avgBitRate} = \text{round}(B \div ((t_2 - t_1) * 1000))$. If $t_1 = t_2$, `avgBitRate` shall be 0. Value zero indicates an unspecified bit rate.

`avgFrameRate` gives the average frame rate in units of frames/(256 seconds). All NAL units in this and lower sub-sequence layers are taken into account in the calculation. In the following, C is the number of frames in this and lower sub-sequence layers. t_1 is the presentation timestamp of the first picture in this and lower sub-sequence layers in presentation order, and t_2 is the

presentation timestamp of the latest picture in this and lower sub-sequence layers in the presentation order. Then, the `avgFrameRate` is calculated as follows provided that $t_1 \neq t_2$:

$$\text{avgFrameRate} = \text{round}(C * 256 \div (t_2 - t_1))$$
 If $t_1 = t_2$, `avgFrameRate` shall be 0. Value zero indicates an unspecified frame rate.

5.4.8 Alternate streams and switching pictures

5.4.8.1 Switching pictures

In typical streaming scenarios, one of the key requirements is to scale the bit rate of the compressed data in response to changing network conditions. The simplest way to achieve this is to encode multiple streams with different bandwidths and quality settings for representative network conditions. The server can then switch amongst these pre-coded streams in response to network conditions. In earlier standards, switching between streams is only possible at I-pictures, because the pictures may only be switched when there are no dependencies on prior pictures for reconstruction.

AVC includes supports for SP-pictures and SI-pictures ("switching pictures") that allow switching from one stream to another while still supporting inter coding of switching pictures. Figure 3 shows how SP pictures are used to switch between two different bit streams.

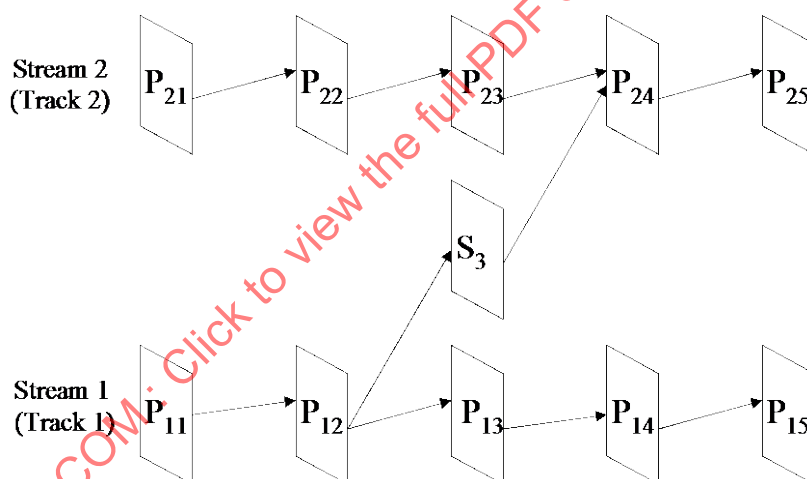


Figure 3 — Stream switching

In the file format, switching pictures are stored in *switching picture* tracks, which are tracks separate from the track that is being switched from and the track being switched to. Switching picture tracks can be identified by the existence of a specific required track reference in that track. A switching picture is an alternative to the sample in the destination track that has exactly the same decoding time. If all switching pictures are SI pictures, then no further information is needed.

If any of the pictures in the switching track are SP pictures, then two extra pieces of information may be needed. First, the source track that is being switched from shall be identified by using a track reference (the source track may be the same track as the destination track). Second, the dependency of the switching picture on the samples in the source track may be needed, so that a switching picture is only used when the pictures on which it depends have been supplied to the decoder.

This dependency is represented by means of an optional extra sample table. There is one entry per sample in the switching track. Each entry records the relative sample numbers in the source track on which the switching picture depends. If this array is empty for a given sample, then that switching sample contains an SI picture. If the dependency box is not present, then only SI-frames shall be present in the track.

A switching sample may have multiple coded representations with different dependencies. For AVC video, the multiple representations of a switching sample are stored in different switching tracks (i.e. access unit). For example, one switch track might contain a SP-picture representation dependent on some earlier samples, used for stream switching, while another switch track may contain another representation as an SI-picture, used for random access.

5.4.8.2 Alternate group

The ISO Base Media File Format (but not the version one specification of the MPEG-4 file format, which is branded as 'mp41') supports the use of what is called *alternate tracks*. Each track can optionally specify an *alternate group* (in the track header box) that groups together alternate encodings of the same content. Thus, each alternate bit-stream can be stored as a separate track and related together as alternate tracks. All the tracks that form a group that may be switched between, but not the tracks containing the switching pictures, shall be a member of an *alternate_group* with a non-zero group identifier.

An alternate group is not needed if there is only one primary track, with a switching track. This switching track may contain SI pictures, or SP pictures for trick modes or error resilience, which predict both from and to the same track.

5.4.8.3 Track references

The switching track shall be linked to the track into which it switches (the destination track) by a track reference of type 'swto' in the switching picture track.

If the switching track contains SP pictures, the switching track shall be linked to the track from which it switches (the source track) by a track reference of type 'swfr' in the switching picture track.

5.4.8.4 Sample dependency

5.4.8.4.1 Definition

Box Type: 'sdep'
 Container: Sample Table 'stbl'
 Mandatory: No
 Quantity: Zero or exactly one.

This subclause defines the dependencies of each switching sample on sample(s) in the source track. This table is only needed in a switching track that has a source ('swfr') track dependency.

This box contains the sample dependencies for each switching sample. The dependencies are stored in the table, one record for each sample. The size of the table, `sample_count` is taken from the `sample_count` in the Sample Size Box ('stsz') or Compact Sample Size Box ('stz2').

5.4.8.4.2 Syntax

```
aligned(8) class SampleDependencyBox
    extends FullBox('sdep', version = 0, 0) {
    for (i=0; i < sample_count; i++){
        unsigned int(16) dependency_count;
        for (k=0; k < dependency_count; k++) {
            signed int(16) relative_sample_number;
        }
    }
}
```

5.4.8.4.3 Semantics

`dependency_count` is an integer that counts the number of samples in the source track on which this switching sample directly depends.

`relative_sample_number` is an integer that identifies a sample in the source track. The relative sample numbers are encoded as follows. If there is a sample in the source track with the same decoding time, it has a relative sample number of 0. Whether or not this sample exists, the sample in the source track that immediately precedes the decoding time of the switching sample has relative sample number -1, the sample before that -2, and so on. Similarly, the sample in the source track that immediately follows the decoding time of the switching sample has relative sample number +1, the sample after that +2, and so on.

5.4.9 Definition of a sub-sample for AVC

For the use of the sub-sample information box (ISO/IEC 14496-12:2020, subclause 8.7.7) in an AVC stream, a sub-sample is defined as one or more contiguous NAL units within a sample and having the same value of the following fields; `RefPicFlag`, `RedPicFlag` and `VclNalUnitFlag`. Each sub-sample includes both NAL unit(s) and their preceding NAL unit length field(s). The presence of this box is optional; however, if present in a track containing AVC data, it shall have the semantics defined here.

The `subsample_priority` field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12.

The `discardable` field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded (e.g. the sub-sample consists of an SEI NAL unit, or a redundant coded picture).

The `codec_specific_parameters` field of the Subsample Information box is defined for AVC as follows:

```
unsigned int(1) RefPicFlag;
unsigned int(1) RedPicFlag;
unsigned int(1) VclNalUnitFlag;
bit(29) reserved = 0;
```

`RefPicFlag` equal to 0 indicates that all the NAL units in the sub-sample have `nal_ref_idc` equal to 0. `RefPicFlag` equal to 1 indicates that all the NAL units in the sub-sample have `nal_ref_idc` greater than 0.

`RedPicFlag` equal to 0 indicates that all the NAL units in the sub-sample have `redundant_pic_cnt` equal to 0. `RedPicFlag` equal to 1 indicates that all the NAL units in the sub-sample have `redundant_pic_cnt` greater than 0.

`VclNalUnitFlag` equal to 0 indicates that all NAL units in the sub-sample are non-VCL NAL units.
Value 1 indicates that all NAL units in the sub-sample are VCL NAL units.

6 SVC elementary stream and sample definitions

6.1 Overview

This clause specifies the storage format of SVC data. It extends the definitions of the storage format of AVC in Clause 5.

The file format for storage of SVC content, as defined in this clause, Annex A, Annex B, Annex C, and Annex D, uses the existing capabilities of the ISO base media file format and the plain AVC file format (i.e. the file format specified in Clause 5). In addition, the following new extensions, among others, to support SVC-specific features are specified.

- a) Scalable or multiview grouping (see Annex B) indicates the association of NAL units with different types and hierarchy levels of scalability.
- b) Aggregators enable efficient scalable grouping of NAL units by changing irregular patterns of NAL units into regular patterns of aggregated data units.
- c) Extractors enable efficient extraction of NAL units from other tracks than the one containing the media data.
- d) Temporal metadata statements (see Annex C) store time-aligned information of media samples.
- e) `SVCConfigurationBox` in track having a sample entry specified in Clause 5 enables the storage of an SVC bitstream in a manner that the AVC compatible base layer can be used by any plain AVC file format compliant reader.

6.2 Elementary stream structure

SVC streams are stored in accordance with 5.2, with the following definition of an SVC video elementary stream:

- **An SVC Video Elementary Streams** contains all video coding related NAL units (i.e. those NAL units containing video data or signalling video structure, possibly after resolution of extractors and aggregators) and may contain non-video coding related NAL units such as SEI messages and access unit delimiter NAL units. Aggregators and Extractors, when present, shall not be directly output by file parsers.

SVC streams may also be stored using associated parameter set streams, if needed.

For SVC streams, Table 1 is updated as follows in Table 2; only entries where the definition for SVC differs from AVC, are shown.

Table 2 — NAL Unit types in SVC and AVC Streams

Value of nal_unit_ type	Description	AVC video elementary stream	SVC video elementary stream (sample entry 'avc1', 'avc2', or 'svc1')	SVC video elementary stream (sample entry 'avc3', 'avc4', or 'svc2')	Parameter set elementary stream
14	Prefix NAL unit in scalable extension prefix_nal_unit_rbsp()	Not specified	Yes	Yes	No
15	Subset sequence parameter set subset_seq_paramete r_set_rbsp()	Not specified	No. If parameter set elementary stream is not used, Subset SPS shall be stored in the Decoder Specific Information.	Yes. Parameter set elementary stream shall not be used.	Yes
20	Coded slice in scalable extension slice_layer_ extension_rbsp()	Not specified	Yes	Yes	No
24 – 31	Not specified	See Annex F	See Annex F	See Annex F	See Annex F

NOTE slice_layer_extension_rbsp was previously called slice_layer_in_scalable_extension_rbsp.

There may be AVC VCL NAL units, SVC VCL NAL units and other NAL units, i.e. non-VCL NAL units, present in an SVC video elementary stream. Additionally, there may be Aggregators and Extractors present in an SVC video elementary stream.

An AVC VCL NAL unit in an SVC video elementary stream conforming to one or more profiles specified in ISO/IEC 14496-10:2020, Annex G, shall be immediately preceded by a prefix NAL unit containing the scalability information for the AVC VCL NAL unit. In this file format an AVC VCL NAL unit and the immediately preceding prefix NAL unit are logically seen as one NAL unit: the prefix NAL unit provides the scalability information and the AVC VCL NAL unit provides the NAL unit type and payload.

6.3 Use of the plain AVC file format

The SVC file format is an extension of the plain AVC file format defined in Clause 5 of this document.

Subclause 5.4.7 is defined for use with plain AVC streams. Its use with SVC streams is deprecated.

6.4 Sample and configuration definition

6.4.1 Canonical order and restrictions

The following restrictions apply to SVC data in addition to the requirements in subclause 5.3.1.

- **SVC coded slice NAL units** (Coded slices in scalable extension): All SVC coded slice NAL units for a single instant in time shall be contained in the sample whose composition time is that of the picture represented by the access unit. An SVC sample shall contain at least one AVC or SVC VCL NAL unit.

- **Prefix NAL units** (Prefix NAL unit in scalable extension): Each prefix NAL unit is placed immediately before the corresponding AVC VCL NAL unit, providing scalability information about the AVC VCL NAL unit.

NOTE Prefix NAL units can also be associated with filler data NAL units.

- **Aggregators/Extractors**: The order of all NAL units included in an Aggregator or referenced by an Extractor is exactly the decoding order as if these NAL units were present in a sample not containing aggregators or extractors. After processing the Aggregator or the Extractor, all NAL units shall be in valid decoding order as specified in ISO/IEC 14496-10.

6.4.2 Decoder configuration record

When the decoder configuration record defined in 5.3.2.1 is used for a stream that can be interpreted as either an SVC or AVC stream, the AVC decoder configuration record shall reflect the properties of the AVC compatible base layer, e.g. it shall contain only parameter sets needed for decoding the AVC base layer.

If the sample entry name is 'svc1', a parameter set stream may be used with SVC streams, as with AVC streams, in which case, parameter sets shall not be included in the decoder configuration record. Otherwise (the sample entry name is 'svc2'), parameter sets may be stored in both the decoder configuration record or as part of samples while a parameter set elementary stream shall not be used.

Sequence or picture parameter sets are numbered in order of storage from 1 to `numOfSequenceParameterSets` or `numOfPictureParameterSets` respectively. Sequence and Picture parameter sets stored in this record in a file may be referenced using this 1-based index by the `InitialParameterSetBox`.

The `SVCD decoderConfigurationRecord` is structurally identical to an `AVC decoderConfigurationRecord`. However, the reserved bits preceding and succeeding the `lengthSizeMinusOne` field are re-defined. The syntax is as follows:

```
aligned(8) class SVCD decoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) AVCProfileIndication;
    unsigned int(8) profile_compatibility;
    unsigned int(8) AVCLevelIndication;
    unsigned int(1) complete_representation;
    bit(5) reserved = '11111'b;
    unsigned int(2) lengthSizeMinusOne;
    bit(1) reserved = '0'b;
    unsigned int(7) numOfSequenceParameterSets;
    for (i=0; i< numOfSequenceParameterSets; i++) {
        unsigned int(16) sequenceParameterSetLength;
        bit(8*sequenceParameterSetLength) sequenceParameterSetNALUnit;
    }
    unsigned int(8) numOfPictureParameterSets;
    for (i=0; i< numOfPictureParameterSets; i++) {
        unsigned int(16) pictureParameterSetLength;
        bit(8*pictureParameterSetLength) pictureParameterSetNALUnit;
    }
}
```

The semantics of the fields `AVCProfileIndication`, `profile_compatibility`, and `AVCLevelIndication` differ from the `AVCDecoderConfigurationRecord` as follows:

The fields `AVCProfileIndication`, `AVCLevelIndication` carry the profile and level indications, respectively, indicating the profile and level of the entire scalable stream in this track. They, and the `profile_compatibility` field, shall have values such that an SVC decoder supporting these values is able to decode bitstreams conforming to the profile, level and profile compatibility flags indicated in any of the sequence parameter sets or subset sequence parameter sets contained in this record.

The semantics of other fields are as follows, or are as defined for an `AVCDecoderConfigurationRecord`:

`complete_representation` is set on a minimal set of tracks that contain a portion of the original encoded scalable stream, as defined in 6.5.1. Other tracks may be removed from the file without loss of any portion of the original encoded bitstream, and, once the set of tracks has been reduced to only those in the complete subset, any further removal of a track removes a portion of the encoded information.

`numOfSequenceParameterSets` indicates the number of SPSs and subset SPSs that are used for decoding the SVC elementary stream. The value of `numOfSequenceParameterSets` shall be in the range of 0 to 64, inclusive.

`SequenceParameterSetLength` indicates the length in bytes of the SPS or subset SPS NAL unit.

`SequenceParameterSetNALUnit` contains a SPS or subset SPS NAL unit. SPSs shall occur in order of ascending parameter set identifier with gaps being allowed. Subset SPSs shall occur in order of ascending parameter set identifier with gaps being allowed. Any SPS shall occur before all the subset SPSs, if any.

6.5 Derivation from the ISO base media file format

6.5.1 SVC track structure

A scalable video stream is represented by one or more video tracks in a file. Each track represents one or more operating points of the scalable stream. A scalable stream may, of course, be further thinned, if desired.

There is a minimal set of one or more tracks that, when taken together, contain the complete set of encoded information. All these tracks shall have the flag “`complete_representation`” set in all their sample entries. This group of tracks that form the complete encoded information are called the “complete subset”.

Let the lowest operating point be the one of all the operating points represented by DTQ (`dependency_id`, `temporal_id` and `quality_id`) combinations that has the least values of `dependency_id`, `temporal_id` and `quality_id`, respectively. The track that has the flag “`complete_representation`” set and contains the lowest operating point shall be nominated as the ‘scalable base track’. All the other tracks that are part of the same scalable encoded information shall be linked to this base track by means of a track reference of type ‘`sbas`’ (scalable base). The complete encoded information can be retained when the tracks included in the “complete subset” are retained; all other tracks shall represent subsets, copies (i.e. the same media data) or re-orderings of the complete subset.

NOTE 1 An alternate group can also include completely independent bitstreams, as well as alternative operating points of the same bitstream. The SVC tracks in the alternate group are to be examined to see how many scalable base tracks are identified.

NOTE 2 "A scalable bitstream" can require more than one track to represent it (consider a stream with a low-resolution, low-frame-rate base layer, and a high resolution enhancement layer, and a high frame-rate enhancement layer, but missing the data for high resolution high frame-rate). However, such a scalable bitstream is typically a non-conforming bitstream.

All the tracks sharing the same scalable base track shall share the same timescale as the scalable base track.

6.5.2 Data sharing and extraction

Different tracks may logically share data. This sharing can take one of the following two forms:

- a) The sample data is copied (i.e., the same media data is present) in different tracks.
- b) There may be instructions on how to perform this copy at the time that the file is read.

For the second case, Extractors (defined in A.3) are used.

6.5.3 SVC video stream definition

6.5.3.1 Sample entry name and format

6.5.3.1.1 Definition

Sample Entry and Box Types: 'svc1', 'svc2', 'svcC', 'seib'

Container: Sample Description Box ('stsd')

Mandatory: One of the 'avc1', 'avc2', 'avc3', 'avc4', 'svc1', and 'svc2' sample entries is mandatory.

Quantity: One or more sample entries may be present

If an SVC elementary stream contains an AVC compatible base layer, then an AVC visual sample entry ('avc1', 'avc2', 'avc3', or 'avc4') shall be used. Here, the entry shall contain initially an AVC Configuration Box, possibly followed by an SVC Configuration Box as defined below. The AVC Configuration Box documents the Profile, Level and Parameter Set information pertaining to the AVC compatible base layer as defined by the `AVCDecoderConfigurationRecord`. The SVC Configuration Box documents the Profile, Level, and possibly also parameter sets pertaining to the entire stream containing the SVC compatible enhancement layers as defined by the `SVCDecoderConfigurationRecord`, stored in the `SVCConfigurationBox`.

If the SVC elementary stream does not contain an AVC base layer, then an SVC visual sample entry ('svc1' or 'svc2') shall be used. The SVC visual sample entry shall contain an SVC Configuration Box, as defined below. This includes an `SVCDecoderConfigurationRecord`, as defined in this document.

The `lengthSizeMinusOne` field in the SVC and AVC configurations in any given sample entry shall have the same value.

A priority assignment URI provides the name (in the URI space) of a method used to assign `priority_id` values. When it occurs in an AVC or SVC sample entry, exactly one URI shall be present, that documents the `priority_id` assignments in the stream. The URI is treated here as a name only; it should be de-

referenceable, though this is not required. File readers may be able to recognize some methods and thereby know what stream extraction operations based on `priority_id` would do.

Extractors or aggregators may be used for SVC VCL NAL units in `'avc1'`, `'avc2'`, `'avc3'`, `'avc4'`, `'svc1'` or `'svc2'` tracks.

NOTE When `'avc1'` or `'avc3'` sample entry is used and both AVC and SVC configurations are present, it can be necessary to indicate an unrealistic level for the AVC base layer, to accommodate the bit rate of the entire stream, because all the NAL units are considered as included in the AVC base layer and hence can be fed to the decoder, which is expected to discard those NAL unit it does not recognize.

Either or both of a `ScalabilityInformationSEIBox` or `SVCConfigurationBox` may be present in an `'avc1'` or `'avc3'` sample entry. In this case the `AVCSVCSampleEntry` definition below applies.

The parameter sets required to decode a NAL unit that is present in the sample data of a video stream, either directly or by reference from an Extractor, shall be present in the decoder configuration of that video stream or in the associated parameter set stream (if used).

Table 3 shows for a video track all the possible uses of sample entries, configurations and the SVC tools (excluding timed metadata, which is always used in another track):

Table 3 — Use of sample entries for AVC and SVC tracks

sample entry name		with configuration records	meaning
<code>'avc1'</code> <code>'avc3'</code>	or	AVC Configuration Only	A plain AVC track with AVC NAL units only; Extractors, aggregators, and tier grouping shall not be present.
<code>'avc1'</code> <code>'avc3'</code>	or	AVC and SVC Configurations	An SVC track with both AVC and SVC NAL units; Extractors and aggregators shall not be present.
<code>'avc2'</code> <code>'avc4'</code>	or	AVC Configuration Only	A plain AVC track with AVC NAL units only; Extractors may be present and used to reference AVC NAL units; Aggregators may be present to contain and reference AVC NAL units; Tier grouping may be present.
<code>'avc2'</code> <code>'avc4'</code>	or	AVC and SVC Configurations	An SVC track with both AVC and SVC NAL units; Extractors and aggregators may be present; Extractors may reference both AVC and SVC NAL units; Aggregators shall not contain but may reference AVC NAL units, and may both contain and reference SVC NAL units; Tier grouping may be present.
<code>'svc1'</code> <code>'svc2'</code>	or	SVC Configuration	An SVC track without AVC NAL units; Extractors shall be present to reference AVC NAL units (from a different track) and may also be used to reference SVC NAL units; Aggregators may be present to contain and reference SVC NAL units; Tier grouping may be present.

6.5.3.1.2 Syntax

```

class SVCConfigurationBox extends Box('svcC') {
    SVCDecoderConfigurationRecord() SVCConfig;
}

class ScalabilityInformationSEIBox extends Box('seib', size)
{
    unsigned int(8*size-64) scalinfosei;
}

class SVCPriorityAssignmentBox extends Box('svcP')
{
    unsigned int(8)    method_count;
    string PriorityAssignmentURI[method_count];
}

class AVCSVCSampleEntry() extends AVCSampleEntry ('avc1' or 'avc3') {
    SVCConfigurationBox  svcconfig;          // optional
    ScalabilityInformationSEIBox  scalability; // optional
    SVCPriorityAssignmentBox  method;          // optional
}

class AVC2SVCSampleEntry() extends AVC2SampleEntry('avc2' or 'avc4') {
    SVCConfigurationBox  svcconfig;          // optional
    ScalabilityInformationSEIBox  scalability; // optional
    SVCPriorityAssignmentBox  method;          // optional
}

// Use this if the track is NOT AVC compatible
class SVCSampleEntry() extends VisualSampleEntry ('svc1' or 'svc2') {
    SVCConfigurationBox  svcconfig;
    MPEG4ExtensionDescriptorsBox descr; // optional
    ScalabilityInformationSEIBox  scalability; // optional
    SVCPriorityAssignmentBox  method;          // optional
}

```

6.5.3.1.3 Semantics

When the sample entry is 'svc1' or 'svc2', Compressorname in the base class VisualSampleEntry indicates the name of the compressor used, with the value "\012SVC Coding" being recommended (\012 is 10, the length of the string "SVC coding" in bytes).

scalinfosei contains an SEI NAL unit containing only a scalability information SEI message as specified in ISO/IEC 14496-10:2020 Annex G. The 'size' field of the container box ScalabilityInformationSEIBox shall not be equal to 0 or 1.

method_count provides a count of the number of following URIs. This field shall take the value 1 in an 'avc1', 'avc2', 'avc3', 'avc4', 'svc1' or 'svc2' sample entry.

PriorityAssignmentURI provides a unique name of the method used to assign priority_id values. In the case of absence of this box, the priority assignment method is unknown.

6.5.4 SVC visual width and height

The visual width and height documented in a VisualSampleEntry of a stream containing SVC VCL NAL unit are the cropped frame dimensions of the AVC base layer, if the stream is described by a sample entry of type 'avc1', 'avc2', 'avc3' or 'avc4'; otherwise they are the cropped frame dimensions of decoded pictures by decoding the entire stream.

6.5.5 Sync sample

For video data described by a sample entry of type 'avc1', 'avc2', 'avc3', or 'avc4', the sync sample table identifies IDR access units for both an AVC decoder, and an SVC decoder (if any) operating on the entire bitstream.

For video data described by a sample entry of type 'svc1', the sync sample table identifies IDR access units in the entire SVC bitstream.

For video data described by a sample entry of type 'svc2', the sync sample table identifies IDR access units in the entire SVC bitstream, and additionally the following applies:

- a) If the sample is a sync sample, all parameter sets needed for decoding that sample shall be included either in the sample entry or in the sample itself.
- b) Otherwise (the sample is not a sync sample), all parameter sets 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.

NOTE The sync sample table, if present, documents only access units that are IDR access units for both the AVC compatible base layer and the layer corresponding to decoding the entire bitstream contained in the track. In case the documenting of layer-specific IDR access units is desired, the stream can be stored in separate tracks, e.g. two tracks, one containing the AVC base layer with a sample entry of type 'avc1' or 'avc3', and the other containing the SVC enhancement layers with a sample entry of type 'svc1' or 'svc2'. However, extractors are then used for tracks that are not the scalable base track.

6.5.6 Shadow sync

A shadow sync box shall not be used for video data described by an 'svc1' or 'svc2' sample entry. Its use for SVC is deprecated.

6.5.7 Independent and disposable samples box

If the SampleDependencyBox is used in a track that is both AVC and SVC compatible, then care should be taken that the information provided by this box is true no matter what valid subset of the SVC data (possibly only the AVC data) is used. The 'unknown' values (value 0 of the fields sample_depends_on, sample_is_depended_on, and sample_has_redundancy) may be needed if the information varies.

6.5.8 Sample groups on random access recovery points 'r011' and random access points 'rap'

For video data described by a sample entry of type 'avc1', 'avc2', 'avc3' or 'avc4', the random access recovery sample group and the random access point sample group identify random access recovery points and random access points, respectively, for both an AVC decoder, and an SVC decoder (if any) operating on the entire bitstream.

NOTE If the random access recovery points or the random access points for the AVC decoder and the SVC decoder operating on the entire bitstream are not all aligned, the random access recovery points table or the random access point table, respectively, will not document all of them. In this case, the stream can be stored in multiple tracks, e.g. two tracks, one containing the AVC base layer with a sample entry of type 'avc1' or 'avc3', and the other containing the SVC enhancement layers with a sample entry of type 'svc1' or 'svc2'.

For video data described by a sample entry of type 'svc1' or 'svc2', the information provided by the random access recovery sample group and the random access point sample group is true for any valid subset of the entire SVC bitstream.

6.5.9 Definition of a sub-sample for SVC

This subclause extends the definition of a sub-sample for AVC in 5.4.9.

For the use of the sub-sample information box (ISO/IEC 14496-12:2020, subclause 8.7.7) in an SVC stream, a sub-sample is defined as one or more contiguous whole NAL units having the same values of the following fields: RefPicFlag, RedPicFlag, VclNalUnitFlag, IdrFlag, PriorityId, DependencyId, QualityId, TemporalId, UseRefBasePicFlag, DiscardableFlag and StoreBaseRepFlag, specified subsequently. Each sub-sample includes both NAL unit(s) and their preceding NAL unit length field(s). The presence of this box is optional; however, if present in a track containing SVC data, it shall have the semantics defined here.

As required in 5.4.9, the subsample_priority field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12.

The codec_specific_parameters field of the Subsample Information box is defined for SVC as follows:

```

unsigned int(1) RefPicFlag;
unsigned int(1) RedPicFlag;
unsigned int(1) VclNalUnitFlag;
bit(6) reserved = 0;
unsigned int(1) IdrFlag;
unsigned int(6) PriorityId;
bit(1) reserved = 0; // corresponding to no_inter_layer_pred_flag
unsigned int(3) DependencyId;
unsigned int(4) QualityId;
unsigned int(3) TemporalId;
unsigned int(1) UseRefBasePicFlag;
unsigned int(1) DiscardableFlag;
bit(1) reserved = 0; // corresponding to output_flag
unsigned int(1) StoreBaseRepFlag;
bit(1) reserved = 0;

```

For an AVC VCL NAL unit in an SVC context, the prefix NAL unit shall be grouped with the AVC VCL NAL unit in the same sub-sample, and its fields values apply to the AVC VCL NAL unit.

RefPicFlag equal to 0 indicates that all the NAL units in the sub-sample have nal_ref_idc equal to 0. RefPicFlag equal to 1 indicates that all the NAL units in the sub-sample have nal_ref_idc greater than 0.

RedPicFlag equal to 0 indicates that all the NAL units in the sub-sample have redundant_pic_cnt equal to 0. RedPicFlag equal to 1 indicates that all the NAL units in the sub-sample have redundant_pic_cnt greater than 0.

VclNalUnitFlag equal to 0 indicates that all NAL units in the sub-sample are non-VCL NAL units. Value 1 indicates that all NAL units in the sub-sample are VCL NAL units.

IdrFlag indicates the idr_flag value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same value of idr_flag.

PriorityId indicates the **priority_id** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same value of **priority_id**.

NoInterLayerPredFlag indicates the **no_inter_layer_pred_flag** of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same value of **no_inter_layer_pred_flag**.

DependencyId indicates the **dependency_id** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same **dependency_id** value.

QualityId indicates the **quality_id** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same **quality_id** value.

TemporalId indicates the **temporal_id** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same **temporal_id** value.

UseRefBasePicFlag indicates the **use_ref_base_pic_flag** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same value of **use_ref_base_pic_flag**.

DiscardableFlag indicates the **discardable_flag** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same **discardable_flag** value.

NOTE This is not the same definition as the **discardable** field in the sub-sample information box.

StoreBaseRepFlag indicates the **store_base_rep_flag** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same value of **store_base_rep_flag**.

7 MVC and MVD elementary stream and sample definitions

7.1 Overview

This clause specifies the storage format of MVC data. It extends the definitions of the storage format of AVC in Clause 5.

The file format for storage of MVC and MVD content, as defined in this clause, Annex A, Annex B, Annex C, and Annex D uses the existing capabilities of the ISO base media file format and the plain AVC file format (i.e. the file format specified in Clause 5). In addition, the following new extensions, among others, to support MVC- and MVD-specific features are specified.

- a) Multiview grouping (see Annex B) indicates the association of NAL units with different types and hierarchy levels of scalability.
- b) Aggregators enable efficient scalable grouping of NAL units by changing irregular patterns of NAL units into regular patterns of aggregated data units.
- c) Extractors enable efficient extraction of NAL units from other tracks than the one containing the media data.
- d) Temporal metadata statements (see Annex C) store time-aligned information of media samples.
- e) **MVCConfigurationBox** in a track having a sample entry specified in Clause 5 enables the storage of an MVC or MVD bitstream in a manner that the AVC compatible base layer can be used by any plain AVC file format compliant reader.

The support for MVC or MVD includes a number of tools, and there are various 'models' of how they might be used. In particular, an MVC or MVD stream can be placed in tracks in a number of ways, among which are the following:

- a) all the views in one track, labelled with sample groups;
- b) each view, including both texture views and depth views when both are present, in its own track, labelled in the sample entries;
- c) a hybrid, one track containing all views, and one or more single-view tracks each containing a view that can be independently coded;
- d) the expected operating points each in a track (e.g. the AVC base, a stereo pair, a multiview scene, or an MVD scene).
- e) (for MVD only) each texture or depth view in its own track, labelled in the sample entries.

The MVC and MVD file format allows storage of one or more views into a track, similarly to the support for SVC in Clause 6. Storage of multiple views per track can be used, e.g. when a content provider wants to provide a multiview bitstream that is not intended for subsetting or when the bitstream has been created for a few pre-defined sets of output views (such as 1, 2, 5, or 9 views) where tracks can be created accordingly. If more than one view is stored in a track and there are several tracks (more than one) representing the MVC or MVD bitstream, the use of the sample grouping mechanism is recommended. The sample grouping mechanism is used to define tiers identifying the views present in the track and to extract required NAL units for certain operating points conveniently. The sample grouping mechanism is usually used with aggregators to form regular NAL unit patterns within samples. Thus, SVC-like sample grouping, aggregators, and view definitions for sample groups are specified for MVC or MVD.

The Multiview Information box ('mvci') is specified to indicate information that applies to more than one view, such as the target output views in one or more Multiview Group boxes. Characteristics (such as camera parameters) of the respective bitstream subset can also be indicated within the Multiview Group box using the Multiview Relation Attributes box ('mvra'), which is similar to the Track Selection box.

A player should have means to determine which views are preferred for displaying, and select one or more tracks that provide the data for the desired operating point, preferring a track that is specific to that operating point over tracks that also contain other data. The display characteristics of players may differ; for example, the number of simultaneously displayed views and the optimal angle between views can be different. In order to guide a player for selection of output views, alternative groups of output views and the common and differentiating characteristics between them can be indicated with the Multiview Group Relation box ('swtc'), which also includes the Multiview Relation Attributes box ('mvra').

When an MVC or MVD bitstream is represented by multiple tracks and a player uses an operating point that contains data in multiple tracks, the player shall reconstruct MVC or MVD access units before passing them to the MVC or MVD decoder. An MVC or MVD operating point may be explicitly represented by a track, i.e., an access unit is reconstructed simply by resolving all extractors and aggregators of a sample. If the number of operating points is large, it may be space-consuming and impractical to create a track for each operating point. In such a case, MVC or MVD access units are reconstructed as specified in 7.6.2. The MVC or MVD Decoder Configuration record contains a field indicating whether the associated samples use explicit or implicit access unit reconstruction (see the explicit_au_track field).

7.2 Overview of MVC or MVD Storage

The storage of MVC and MVD streams can be supported by a number of structures, including information in the sample entry, the media information box, and sample groups. Table 4 provides an overview of the structures provided, their names, and a brief description of their functions.

NOTE Each group of rows starting with an entry in the left column (e.g. 'minf', '?vc?') document a containment structure within that container; the higher level containment is not shown.

Table 4 — Box, sample entry and group structures for MVC and MVD Streams

				Box Name	Brief Description
minf				Media Information Box	
	mvci			Multiview Information Box	
		mvcg		Multiview Group Box	Specifies a multiview group for the views of the multiview video stream that are output
			buff	Buffering Information Box	Contains the buffering information of the bitstream subset specified by the multiview group
			mvra	Multiview Relation Attribute Box	Indicates the relation of the tracks or tiers of the respective multiview group with each other (when contained in a Multiview Group box)
			tibr	Tier Bit Rate Box	Provides information about the bit rate values of the bitstream subset specified by the multiview group
			tiri	Tier Information Box	Provides information about the profile, level, frame size, discardability, and frame-rate of the bitstream subset specified by the multiview group
			vwidi	Multiview Scene Information Box	Indicates the maximum disparity in a scene with multiple views
		swtc		Multiview Group Relation Box	Specifies a set of multiview groups from which one multiview group is decoded and played at any time
			mvra	Multiview Relation Attribute Box	Indicates the relation of the multiview groups with each other (when contained in a Multiview Group Relation box)
?vc?				Sample Entry	(Note: various codes are used for sample entries)
	vsib			View Scalability Information SEI Message Box	Contains an SEI NAL unit containing only a view scalability information SEI message as specified in ISO/IEC 14496-10:2020 Annex H
	ecam			Extrinsic Camera Parameters Box	Contains camera parameters that define the location and orientation of the camera reference frame with respect to a known world reference frame
	icam			Intrinsic Camera Parameters Box	Contains camera parameters that link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame
	vwidi			View Identifier Box	Indicates the views included in the track (when included in a sample entry)
	mvcp			MVC View Priority Assignment Box	Provides a URI containing a unique name of the method used to assign content_priority_id values for the View Priority sample grouping
	mvcc			MVC Configuration Box	
	mvdc			MVCD Configuration Box	Contains the MVD decoder configuration record and the MVD depth resolution box (for MVD streams only)
		3dpr		MVD Depth Resolution Box	Provides the resolution of depth views (for MVD streams only)
	3sib			MVD Scalability Information SEI Message Box	Contains an SEI NAL unit containing only an MVCD view scalability information SEI message as specified in ISO/IEC 14496-10:2020 Annex I
sgpd				Sample Group Description Box	
	mvif			Multiview Group Entry	Contains the following boxes
		buff		Buffering Information Box	Contains the buffer information of the tier

		ldep		Tier Dependency Box	Identifies the tiers that the current tier is dependent on
		svip		Initial Parameter Sets Box	Contains parameter sets needed for decoding this tier and all the tiers it depends on
		svpr		Priority Range Box	Reports the minimum and maximum priority_id of the NAL units mapped to this tier
		tibr		Tier Bit Rate Box	Provides information about the bit rate values of a tier
		tiri		Tier Information Box	Provides information about the profile, level, frame size, discardability, and frame-rate of a tier
		vipr		View Priority Box	Labels views with priorities based on content
		vwid		View Identifier Box	Indicates the views included in the tier (when included in a Multiview Group entry,)
	dtrt			Decode Re-timing Group Entry	Provides adjusted decoding times when high temporal layers are discarded
	scnm			Sample Map Group Entry	Provides the mapping of NAL units to multiview groups for all samples in the track

The structures within a sample entry provide information for the decoding or use of the samples (video information) that are associated with that sample entry. Sample groups provide time-varying information about the track as a whole, assisting (for example) with the extraction of subsets of the media within a track. Information in the Multiview Information Box (appearing in the media information box) can span several tracks and is descriptive of collections of tracks, even though the Multiview Information Box resides in the track containing the base view of the stream.

7.3 MVC and MVD elementary stream structures

MVC and MVD streams are stored in accordance with 5.2, with the following definition of an MVC or MVD video elementary stream:

- **An MVC and MVD Video Elementary Stream** contains all video coding related NAL units (i.e. those NAL units containing video data or signalling video structure, possibly after resolution of extractors and aggregators) and may contain non-video coding related NAL units such as SEI messages and access unit delimiter NAL units. Aggregators and Extractors, when present, shall not directly output by file parsers.

MVC and MVD streams may also be stored using associated parameter set streams, when needed.

For MVC and MVD streams, Table 1 is amended by the following table Table 5.

Table 5 — NAL Unit types in MVC, MVD, and AVC Streams

Value of nal_unit_type	Description	AVC video elementary stream	MVC video elementary stream (sample entry 'avc1', 'avc2', 'mvc1', or 'mvc2')	MVC video elementary stream (sample entry name 'avc3', 'avc4', 'mvc3', or 'mvc4')	MVD video elementary stream (sample entry name 'mvd1', 'mvd2', 'a3d1', or 'a3d2')	MVD video elementary stream (sample entry name 'mvd3', 'mvd4', 'a3d3', or 'a3d3')	Parameter set elementary stream
14	Prefix NAL unit prefix_nal_unit_rbsp()	Not specified	Yes	Yes	Yes	Yes	No
15	Subset sequence parameter set subset_seq_parameter_set_rbsp()	Not specified	No	Yes Parameter set elementary stream shall not be used	No	Yes Parameter set elementary stream shall not be used	Yes
20	Coded slice extension slice_layer_extension_rbsp()	Not specified	Yes	Yes	Yes	Yes	No
21	Coded slice extension for a depth view component or a 3D-AVC texture view component slice_layer_extension_rbsp()	Not specified	Not specified	Not specified	Yes	Yes	No
24 – 31	Not specified	Not specified	See Annex F	See Annex F	See Annex F	See Annex F	See Annex F

There may be AVC VCL NAL units, MVC VCL NAL units and other NAL units, i.e. non-VCL NAL units, present in an MVC video elementary stream. There may be AVC VCL NAL units, MVC VCL NAL units, MVD VCL NAL units, and non-VCL NAL units present in an MVD video elementary stream. Additionally, there may be Aggregators or Extractors present in an MVC or MVD video elementary stream.

An AVC VCL NAL unit in an MVC or MVD video elementary stream conforming to one or more profiles specified in Annex H, Annex I, or Annex J of ISO/IEC 14496-10:2020 shall be immediately preceded by a prefix NAL unit. In this document, an AVC VCL NAL unit and the immediately preceding prefix NAL unit are logically seen as one NAL unit.

7.4 Use of the plain AVC file format

The MVC or MVD file format is an extension of the plain AVC file format defined in Clause 5 of this document.

Subclause 5.4.7 is defined for use with plain AVC streams. Its use with MVC and MVD streams is deprecated.

7.5 Sample and configuration definition

7.5.1 Canonical order and restriction

The following restrictions apply to MVC and MVD data in addition to the requirements in subclause 5.3.1.

- **MVC coded slice NAL units** (Coded slice extension): All MVC coded slice NAL units for a single instant in time shall be contained in the sample whose composition time is that of the picture represented by the access unit. An MVC sample shall contain at least one AVC or MVC VCL NAL unit.
- **MVD VCL NAL units** (Coded slice extension): All MVD VCL NAL units for a single instant in time shall be contained in the sample whose composition time is that of the picture represented by the access unit. An MVD sample shall contain at least one AVC, MVC or MVD VCL NAL unit.
- **Prefix NAL units**: Each prefix NAL unit is placed immediately before the corresponding AVC VCL NAL unit.
- **Aggregators/Extractors**: The order of all NAL units included in an Aggregator or referenced by an Extractor is exactly the decoding order as if these NAL units were present in a sample not containing aggregators or extractors. After processing the Aggregator or the Extractor, all NAL units shall be in valid decoding order as specified in ISO/IEC 14496-10.

7.5.2 Decoder configuration record

7.5.2.1 MVC decoder configuration record

When the AVC decoder configuration record (as defined in subclause 5.3.2.1) is used for a stream that can be interpreted as either an MVC or AVC stream, the AVC decoder configuration record shall reflect the properties of the AVC compatible base view, e.g. it shall contain only parameter sets needed for decoding the AVC base view.

If the sample entry name is 'mvc1' or 'mvc2', a parameter set stream may be used with MVC streams, as with AVC streams. In that case, parameter sets shall not be included in the decoder configuration record. Otherwise (the sample entry name is 'mvc3' or 'mvc4'), parameter sets may be stored in both the decoder configuration record or as part of samples while a parameter set elementary stream shall not be used.

Sequence or picture parameter sets, including subset sequence parameter sets, are numbered in order of storage from 1 to `numOfSequenceParameterSets` or `numOfPictureParameterSets`, respectively. Sequence and picture parameter sets stored in this record in a file may be referenced using this 1-based index by the `InitialParameterSetBox`.

The `MVCDecoderConfigurationRecord` is structurally identical to an `AVCDecoderConfigurationRecord`. However, the reserved bits preceding and succeeding the `lengthSizeMinusOne` field are re-defined. The syntax is as follows:

```

aligned(8) class MVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) AVCProfileIndication;
    unsigned int(8) profile_compatibility;
    unsigned int(8) AVCLevelIndication;
    unsigned int(1) complete_representation;
    unsigned int(1) explicit_au_track;
    bit(4) reserved = '1111'b;
    unsigned int(2) lengthSizeMinusOne;
    bit(1) reserved = '0'b;
    unsigned int(7) numOfSequenceParameterSets;
    for (i=0; i< numOfSequenceParameterSets; i++) {
        unsigned int(16) sequenceParameterSetLength ;
        bit(8*sequenceParameterSetLength) sequenceParameterSetNALUnit;
    }
    unsigned int(8) numOfPictureParameterSets;
    for (i=0; i< numOfPictureParameterSets; i++) {
        unsigned int(16) pictureParameterSetLength;
        bit(8*pictureParameterSetLength) pictureParameterSetNALUnit;
    }
}

```

The semantics of the fields `AVCProfileIndication`, `profile_compatibility`, and `AVCLevelIndication` differ from the `AVCDecoderConfigurationRecord` as follows:

The fields `AVCProfileIndication`, `AVCLevelIndication` carry the profile and level indications, respectively, indicating the profile and level for the bitstream represented by this track, i.e., the bitstream that contains all the views of this track and the views required for decoding of this track and wherein all the views in this track are the target output views. If `AVCLevelIndication` is equal to 0, the level that applies to the bitstream defined above operating with all the views of this track being the target output views is unspecified. `AVCProfileIndication`, `profile_compatibility`, and `AVCLevelIndication`, if non-zero, shall have values such that an MVC decoder supporting these values is able to decode bitstreams conforming to the profile, level and profile compatibility flags indicated in any of the sequence parameter sets or subset sequence parameter sets contained in this record.

The semantics of other fields are as follows, or, if not present in the following, are as defined for an `AVCDecoderConfigurationRecord`:

`complete_representation` is set on a minimal set of tracks that contain a portion of the original encoded stream, as defined in 7.6.1. Other tracks may be removed from the file without loss of any portion of the original encoded bitstream, and, once the set of tracks has been reduced to only those in the complete subset, any further removal of a track removes a portion of the encoded information.

`explicit_au_track` is set on a track that is 'complete'; it is not necessary to determine the view dependencies, nor calculate whether views not present in this track should be found from other tracks. However, subject to the rules for the sample entry types, extractors may be present and need to be followed to gather all the NAL units needed.

`numOfSequenceParameterSets` indicates the number of SPSs and subset SPSs that are used for decoding the MVC elementary stream.

`SequenceParameterSetLength` indicates the length in bytes of the SPS or subset SPS NAL unit.

`SequenceParameterSetNALUnit` contains a SPS or subset SPS NAL unit as specified in Annex H of ISO/IEC 14496-10:2020. SPSs shall occur in order of ascending parameter set identifier with gaps being allowed. Subset SPSs shall occur in order of ascending parameter set identifier with gaps being allowed. Any SPS shall occur before all the subset SPSs, if any.

7.5.2.2 MVD decoder configuration record

The syntax structure of `MVDDecoderConfigurationRecord` is exactly the same as `MVCDecoderConfigurationRecord`.

When the AVC decoder configuration record (as defined in subclause 5.3.2.1) is used for a stream that can be interpreted as an MVD stream, the AVC decoder configuration record shall reflect the properties of the AVC compatible base view, e.g. it may contain only parameter sets needed for decoding the AVC base view.

When the MVC decoder configuration record (as defined in subclause 7.5.2.1) is used for a stream that can be interpreted as an MVC or MVD stream, the MVC decoder configuration record shall reflect the properties of the MVC compatible bitstream subset (i.e. the bitstream subset with only the texture views), e.g. it may contain only parameter sets needed for decoding the MVC compatible bitstream subset.

If the sample entry name is 'mvd1', 'mvd2', 'a3d1', or 'a3d2', a parameter set stream may be used with MVD streams, as with AVC and MVC streams. In that case, parameter sets shall not be included in the decoder configuration record. Otherwise (the sample entry name is 'mvd3', 'mvd4', 'a3d3' or 'a3d4'), parameter sets may be stored in both the decoder configuration record or as part of samples while a parameter set elementary stream shall not be used.

Sequence or picture parameter sets, including subset sequence parameter sets, are numbered in order of storage from 1 to `numOfSequenceParameterSets` or `numOfPictureParameterSets`, respectively. Sequence and picture parameter sets stored in this record in a file may be referenced using this 1-based index by the `InitialParameterSetBox`.

The semantics of the fields `AVCProfileIndication`, `profile_compatibility`, and `AVCLevelIndication` differ from the `MVCDecoderConfigurationRecord` as follows. `AVCProfileIndication`, `profile_compatibility`, and `AVCLevelIndication`, if non-zero, shall have values such that a MVD decoder supporting these values is able to decode bitstreams conforming to the profile, level and profile compatibility flags indicated in any of the sequence parameter sets or subset sequence parameter sets contained in this record.

The semantics of other fields are as follows, or, if not present, are as defined for an `MVCDecoderConfigurationRecord`:

- `numOfSequenceParameterSets` indicates the number of SPSs and subset SPSs that are used for decoding the MVD elementary stream.
- `sequenceParameterSetNALUnit`, when contained in the MVCD Configuration Box, contains a SPS or subset SPS NAL unit as specified in Annex I of ISO/IEC 14496-10:2020.
- `sequenceParameterSetNALUnit`, when contained in the A3D Configuration Box, contains a SPS or subset SPS NAL unit as specified in Annex J of ISO/IEC 14496-10:2020. SPSs shall occur in order of ascending parameter set identifier with gaps being allowed. Subset SPSs shall occur in

order of ascending parameter set identifier with gaps being allowed. Any SPS shall occur before all the subset SPSs, if any.

7.6 Derivation from the ISO base media file format

7.6.1 MVC and MVD track structures

An MVC or MVD stream is represented by one or more video tracks in a file. Each track represents one or more views of the stream. For a track in a file storing an MVD video stream, the track may contain texture only, depth only or both texture and depth.

There is a minimal set of one or more tracks that, when taken together, contain the complete set of encoded information. All these tracks shall have the flag “complete_representation” set in all their sample entries. This group of tracks that form the complete encoded information are called the “complete subset”.

The track that has the flag “complete_representation” set and contains NAL units of the base view with temporal_id equal to 0 shall be nominated as the ‘base view track’. All the other tracks that are part of the same stream shall be linked to this base track by means of a track reference of type 'sbas' (view base). The complete encoded information can be retained when the tracks included in the “complete subset” are retained; all other tracks shall represent subsets, copies (i.e. the same media data) or reorderings of the complete subset.

All the tracks sharing the same base view track shall share the same timescale as the scalable base track. For MVD streams, all the tracks containing the texture view and the depth view of a particular view shall share the same timescale. Note that the texture view and the depth view of a particular view have the same value of view_id.

If a view represented by a track uses another view represented by another track as an inter-view prediction reference, a track reference of type 'scal' shall be included in the track referring to the source track for inter-view prediction.

For MVD streams, if a depth view is stored in a different track than the track containing the texture view associated with the depth view, a track reference of type 'deps' shall be included in the track containing the depth view, referring to the track containing the texture view. The presence of this track reference indicates that the current track contains the depth view that is associated with a texture view in the referenced track.

NOTE If a track containing a part of an MVC or MVD bitstream is removed from a file, care needs to be taken to remove also those tracks that contain 'scal' and 'sbas' track references to the removed track and references to the multiview groups that include the removed track.

7.6.2 Reconstruction of an access unit

In order to reconstruct an access unit from samples of one or more MVC or MVD tracks, the target output views may need to be determined first, by examining the Multiview Group box (7.7.3) and the Multiview Group Relation box (7.7.4). The explicit_au_track flag equal to 1 states that this track is a complete operating point; nonetheless, the track should be examined to determine which views delivered by this track are the output views.

If the target output views are not exactly represented by any track marked with `explicit_au_track` equal to 1 in the MVC decoder configuration record, access units are reconstructed as follows.

The views that are required for decoding the determined target output views can be concluded from reference view identifiers included in the View Identifier box, the '`scal`' track references, or Tier Dependency boxes.

If several tracks contain data for the access unit, the alignment of respective samples in tracks is performed on decoding time, i.e. using the time-to-sample table only without considering edit lists.

An access unit is reconstructed from the respective samples in the required tracks and tiers by arranging their NAL units in an order conforming to ISO/IEC 14496-10. Duplicate copies of NAL units, when present, need to be removed in order to conform to ISO/IEC 14496-10. The following order provides an outline of the procedure to form a conforming access unit:

- All parameter set NAL units (from the associated parameter set tracks and from the associated elementary stream tracks).
- All SEI NAL units (from the associated parameter set tracks and from the associated elementary stream tracks).
- View components in ascending order of view order index value.
- Within a view component, if both texture and depth are present, then the texture view component precedes the depth view component.
- NAL units within a texture or depth view component are in their appearance order within the sample.

7.6.3 Sample entry

7.6.3.1 Boxes for sample entry

7.6.3.1.1 Intrinsic camera parameters box

7.6.3.1.1.1 Definition

Box Type: '`icam`'
 Container: Sample Entry ('`avc1`', '`avc2`', '`avc3`', '`avc4`',
 '`mvc1`', '`mvc2`', '`mvc3`', '`mvc4`',
 '`mvd1`', '`mvd2`', '`mvd3`', '`mvd4`',
 '`a3d1`', '`a3d2`', '`a3d3`', '`a3d4`')
 Mandatory: No
 Quantity: Zero or more

This subclause specifies intrinsic camera parameters that link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame. A specification of focal length and parameters related to the geometric distortion due to camera optics is given in Annex H of ISO/IEC 14496-10:2020.

7.6.3.1.1.2 Syntax

```
class IntrinsicCameraParametersBox extends FullBox ('icam', version=0, flags) {
    unsigned int(6)    reserved=0;
    unsigned int(10)   ref_view_id;
    unsigned int(32)   prec_focal_length;
    unsigned int(32)   prec_principal_point;
    unsigned int(32)   prec_skew_factor;
    unsigned int(8)    exponent_focal_length_x;
    signed   int(64)   mantissa_focal_length_x;
    unsigned int(8)    exponent_focal_length_y;
    signed   int(64)   mantissa_focal_length_y;
    unsigned int(8)    exponent_principal_point_x;
    signed   int(64)   mantissa_principal_point_x;
    unsigned int(8)    exponent_principal_point_y;
    signed   int(64)   mantissa_principal_point_y;
    unsigned int(8)    exponent_skew_factor;
    signed   int(64)   mantissa_skew_factor;
}
```

7.6.3.1.1.3 Semantics

reserved this field shall be equal to zero

ref_view_id indicates the **view_id** identifying a view for which intrinsic camera parameters are indicated in this Intrinsic Camera Parameters Box

prec_focal_length specifies the exponent of the maximum allowable truncation error for **focal_length_x** and **focal_length_y** as given by $2^{-\text{prec_focal_length}}$. The value of **prec_focal_length** shall be in the range of 0 to 31, inclusive.

prec_principal_point specifies the exponent of the maximum allowable truncation error for **principal_point_x** and **principal_point_y** as given by $2^{-\text{prec_principal_point}}$. The value of **prec_principal_point** shall be in the range of 0 to 31, inclusive.

prec_skew_factor specifies the exponent of the maximum allowable truncation error for skew factor as given by $2^{-\text{prec_skew_factor}}$. The value of **prec_skew_factor** shall be in the range of 0 to 31, inclusive.

exponent_focal_length_x specifies the exponent part of the focal length in the horizontal direction. The value of **exponent_focal_length_x** shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified focal length.

mantissa_focal_length_x specifies the mantissa part of the focal length of the *i*-th camera in the horizontal direction.

exponent_focal_length_y specifies the exponent part of the focal length in the vertical direction. The value of **exponent_focal_length_y** shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified focal length.

mantissa_focal_length_y specifies the mantissa part of the focal length in the vertical direction.

mantissa_principal_point_x specifies the mantissa part of the principal point in the horizontal direction.

exponent_principal_point_y specifies the exponent part of the principal point in the vertical direction. The value of **exponent_principal_point_y** shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified principal point.

`mantissa_principal_point_y` specifies the mantissa part of the principal point in the vertical direction.

`exponent_skew_factor` specifies the exponent part of the skew factor. The value of `exponent_skew_factor` shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified skew factor.

`mantissa_skew_factor` specifies the mantissa part of the skew factor.

The intrinsic matrix A for the camera associated to the view indicated by `ref_view_id` is represented as follows:

$$\begin{bmatrix} \text{focalLengthX} & \text{skewFactor} & \text{principalPointX} \\ 0 & \text{focalLengthY} & \text{principalPointY} \\ 0 & 0 & 1 \end{bmatrix}$$

Each component of the intrinsic matrix is obtained from the variables specified in Table 6 as the variable x computed as follows.

- If $0 < e < 63$, $x = 2^{e-31} * (1 + n \div 2v)$, with $v = \max(0, e + p - 31)$ [Eq. F-1]
- If e is equal to 0, $x = 2^{-(30+v)} * n$, with $v = \max(0, p - 30)$ [Eq. F-2]

Table 6 — Association between camera parameter variables and syntax elements

x	e	n	p
<code>focalLengthX</code>	<code>exponent_focal_length_x</code>	<code>mantissa_focal_length_x</code>	<code>prec_focal_length</code>
<code>focalLengthY</code>	<code>exponent_focal_length_y</code>	<code>mantissa_focal_length_y</code>	<code>prec_focal_length</code>
<code>principalPointX</code>	<code>exponent_principal_point_x</code>	<code>mantissa_principal_point_x</code>	<code>prec_principal_point</code>
<code>principalPointY</code>	<code>exponent_principal_point_y</code>	<code>mantissa_principal_point_y</code>	<code>prec_principal_point</code>
<code>skewFactor</code>	<code>exponent_skew_factor</code>	<code>mantissa_skew_factor</code>	<code>prec_skew_factor</code>

7.6.3.1.2 Extrinsic camera parameters box

7.6.3.1.2.1 Definition

Box Type: 'ecam'

Container: Sample Entry ('avc1', 'avc2', 'avc3', 'avc4',
'mve1', 'mvc2', 'mvc3', 'mvc4',
'mvd1', 'mvd2', 'mvd3', 'mvd4',
'a3d1', 'a3d2', 'a3d3', 'a3d4')

Mandatory: No

Quantity: Zero or more

This subclause specifies extrinsic camera parameters that define the location and orientation of the camera reference frame with respect to a known world reference frame. A specification of extrinsic camera parameters including translation vector and rotation matrix is given in Annex H of ISO/IEC 14496-10:2020.

The extrinsic camera parameters are specified according to a right-handed coordinate system, where the upper left corner of the image is the origin, i.e., the (0, 0) coordinate, with the other corners of the image having non-negative coordinates. With these specifications, a 3-dimensional world point, $wP = [x \ y \ z]$ is mapped to a 2-dimensional camera point, $cP = [u \ v \ 1]$, according to:

$$s * cP = A * R^{-1} * (wP - T)$$

where A denotes the intrinsic camera parameter matrix that can be indicated by an intrinsic camera parameters box (see 7.6.3.1.1), R^{-1} denotes the inverse of the rotation matrix R , T denotes the translation vector, and s (a scalar value) is an arbitrary scale factor chosen to make the third coordinate of cP equal to 1. The elements of A , R , T are determined according the syntax elements signalled in this box and as specified below.

7.6.3.1.2.2 Syntax

```
class ExtrinsicCameraParametersBox extends FullBox ('ecam', version=0, flags) {
    unsigned int(6)    reserved=0;
    unsigned int(10)   ref_view_id;
    unsigned int(8)    prec_rotation_param;
    unsigned int(8)    prec_translation_param;
    for (j=1; j<=3; j++) { /* row */
        for (k=1; k<=3; k++) { /* column */
            unsigned int(8)    exponent_r[j][k];
            signed   int(64)    mantissa_r [j][k];
        }
        unsigned int(8)    exponent_t[j];
        signed   int(64)    mantissa_t[j];
    }
}
```

7.6.3.1.2.3 Semantics

`reserved` this field shall be equal to zero

`ref_view_id` indicates the `view_id` identifying a view for which intrinsic camera parameters are indicated in this Intrinsic Camera Parameters Box

`prec_rotation_param` specifies the exponent of the maximum allowable truncation error for $r[j][k]$ as given by $2^{-\text{prec_rotation_param}}$. The value of `prec_rotation_param` shall be in the range of 0 to 31, inclusive.

`prec_translation_param` specifies the exponent of the maximum allowable truncation error for $t[j]$ as given by $2^{-\text{prec_translation_param}}$. The value of `prec_translation_param` shall be in the range of 0 to 31, inclusive.

`exponent_r[j][k]` specifies the exponent part of (j, k) component of the rotation matrix. The value of `exponent_r[j][k]` shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified rotation matrix.

`mantissa_r[j][k]` specifies the mantissa part of (j, k) component of the rotation matrix.

`exponent_t[j]` specifies the exponent part of the j -th component of the translation vector. The value of `exponent_t[j]` shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU T | ISO/IEC. Decoders shall treat the value 63 as indicating an unspecified translation vector.

`mantissa_t[j]` specifies the mantissa part of the j -th component of the translation vector.

The rotation matrix R is represented as follows:

$$\begin{bmatrix} rE[0][0] & rE[0][1] & rE[0][2] \\ rE[1][0] & rE[1][1] & rE[1][2] \\ rE[2][0] & rE[2][1] & rE[2][2] \end{bmatrix}$$

The translation vector T is represented as follows:

$$\begin{bmatrix} tE[0] \\ tE[1] \\ tE[2] \end{bmatrix}$$

Each component of the rotation matrix and the translation vector is obtained from the variables specified in Table 7 as the variable x computed as follows.

– If $0 < e < 63$, $x = 2^{e-31} * (1 + n \div 2^v)$, with $v = \max(0, e + p - 31)$ [Eq. F-3]

– If e is equal to 0, $x = 2^{-(30+v)} * n$, with $v = \max(0, p - 30)$ [Eq. F-4]

Table 7 — Association between camera parameter variables and syntax elements

x	e	n	p
$rE[j][k]$	$exponent_r[j][k]$	$mantissa_r[j][k]$	$prec_rotation_param$
$tE[j]$	$exponent_t[j]$	$mantissa_t[j]$	$prec_translation_param$

7.6.3.1.3 View identifier box

7.6.3.1.3.1 Definition

Box Type: 'vwid'

Container: Sample Entry ('avc1', 'avc2', 'avc3', 'avc4',
'mvc1', 'mvc2', 'mvc3', 'mvc4',
'mvd1', 'mvd2', 'mvd3', 'mvd4',
'a3d1', 'a3d2', 'a3d3', 'a3d4')
or MultiviewGroupEntry

Mandatory: Yes (for sample entries and the primary group definition in Multiview Group entries)

Quantity: Exactly one (for sample entries and the primary group definition in Multiview Group entries)

Zero for non-primary group definitions in Multiview Group entries

When included in a sample entry, this box indicates the views included in the track. When included in a Multiview Group entry, this box indicates the views included in the respective tier. This box also indicates the view order index for each listed view. Additionally, the box includes the minimum and maximum values of temporal_id included in the track or tier when the View Identifier box is included in a sample entry or Multiview Group entry, respectively. Moreover, the box indicates the referenced views required for decoding the views included in the track or tier. Moreover, for MVD streams, the box indicates, for each of the view included in the track, the presence of texture and/or depth in the track and in the stream.

7.6.3.1.3.2 Syntax

```
class ViewIdentifierBox extends FullBox ('vwid', version=0, flags)
{
    unsigned int(2)    reserved6 = 0;
    unsigned int(3)    min_temporal_id;
    unsigned int(3)    max_temporal_id;
    unsigned int(16)   num_views;
    for (i=0; i<num_views; i++) {
        unsigned int(6)    reserved1 = 0;
        unsigned int(10)   view_id[i];
        unsigned int(6)    reserved2 = 0;
        unsigned int(10)   view_order_index;
        unsigned int(1)    texture_in_stream[i];
        unsigned int(1)    texture_in_track[i];
        unsigned int(1)    depth_in_stream[i];
        unsigned int(1)    depth_in_track[i];
        unsigned int(2)    base_view_type;
        unsigned int(10)   num_ref_views;
        for (j = 0; j < num_ref_views; j++) {
            unsigned int(4)    reserved5 = 0;
            unsigned int(2)    dependent_component_idc[i][j];
            unsigned int(10)   ref_view_id[i][j];
        }
    }
}
```

7.6.3.1.3.3 Semantics

`min_temporal_id`, `max_temporal_id` take the minimum and maximum value, respectively, of the `temporal_id` syntax element that is present in the NAL unit header extension of the NAL units mapped to the track or tier when the View Identifier box is included in a sample entry or Multiview Group entry, respectively. For AVC streams this takes the value that is, or would be, in the prefix NAL unit.

`num_views`, when the View Identifier box is present in a sample entry, indicates the number of views included in the track. When the View Identifier box is present in a Multiview Group entry, `num_views` indicates the number of views included in the respective tier.

`view_id[i]` indicates the value of the `view_id` syntax element in the NAL unit header extension of a view included in the track or tier when the View Identifier box is included in a sample entry or Multiview Group entry, respectively.

`view_order_index` indicates the value of the `VOIdx` variable, as specified in Annex H of ISO/IEC 14496-10:2020, for a view included in the track or tier when the View Identifier box is included in a sample entry or Multiview Group entry, respectively.

`texture_in_stream[i]` equal to 1 indicates that the texture view for the view with `view_id[i]` is present in the stream. The value indicates that the texture view for the view with `view_id[i]` is not present in the stream.

`texture_in_track[i]` equal to 1 indicates that the texture view for the view with `view_id[i]` is present in the track. The value indicates that the texture view for the view with `view_id[i]` is not present in the track. When `texture_in_stream[i]` is equal to 0, the value of `texture_in_track[i]` shall be equal to 0.

`depth_in_stream[i]` equal to 1 indicates that the depth view for the view with `view_id[i]` is present in the stream. The value indicates that the depth view for the view with `view_id[i]` is not present in the stream. When `texture_in_stream[i]` is equal to 0, the value of `depth_in_stream[i]` shall be equal to 1.

`depth_in_track[i]` equal to 1 indicates that the depth view for the view with `view_id[i]` is present in the track. The value indicates that the depth view for the view with `view_id[i]` is not present in the track. When `depth_in_stream[i]` is equal to 0, the value of `depth_in_track[i]` shall be equal to 0. When `texture_in_track[i]` is equal to 0, the value of `depth_in_track[i]` shall be equal to 1.

`base_view_type` indicates whether the view is a base view (virtual or not). It takes the following values:

0 indicates that the view is neither a base view nor virtual base view.

1 shall be used to label the non-virtual base view of the MVC bitstream.

2 is a reserved value and shall not be used.

3 indicates that the view with `view_id[i]` is a virtual base view. The respective independently coded non-base view with `view_id[i]` resides in another track. When `base_view_type` is equal to 3, the subsequent `num_ref_views` shall be equal to 0.

`num_ref_views` indicates the number of views that may be directly or indirectly referenced by the view with `view_id[i]`.

`dependent_component_idc[i][j]` indicates how the texture view and depth view of the *j*-th reference view are required for decoding the view with `view_id[i]`. If the value is equal to 0, only the texture view of the reference view is required. If the value is equal to 1, only the depth view of the reference view is required. If the value is equal to 2, both texture view and depth view of the reference view are required. The value of 3 is reserved.

`ref_view_id[i][j]` indicates the view identifier of the *j*-th view that may be directly or indirectly referenced by the view with `view_id[i]`, i.e., that may be required for decoding of the view with `view_id[i]`. If a view is required for decoding the view with `view_id[i]`, it shall be listed as one of `ref_view_id[i][j]`. When the View Identifier box is included in a sample entry, it is recommended to indicate the referenced views for both anchor and non-anchor access units in the same sample entry.

7.6.3.2 MVC and MVD sample entry definitions

7.6.3.2.1 Definition

Sample Entry and Box Types:

'mvc1', 'mvc2', 'mvc3', 'mvc4',
'mvd1', 'mvd2', 'mvd3', 'mvd4',
'a3d1', 'a3d2', 'a3d3', 'a3d4'

Container: Sample Description Box ('std')

Mandatory: One of the above listed sample entries is mandatory

Quantity: One or more sample entries may be present

When present, the AVC Configuration Box documents the Profile, Level, and possibly also parameter sets pertaining to the AVC compatible base view as defined by the `AVCDecoderConfigurationRecord`. When present, the MVC Configuration Box documents the Profile, Level and Parameter Set information pertaining to the entire MVC stream as defined by the `MVCDecoderConfigurationRecord`. When present, the MVCD Configuration Box documents the Profile, Level and Parameter Set information pertaining to the entire MVC+D stream as defined by the `MVDDecoderConfigurationRecord`. When present, the A3D Configuration Box documents the Profile, Level and Parameter Set information pertaining to the entire 3D-AVC stream as defined by the `MVDDecoderConfigurationRecord`.

For the AVC sample entries 'avc1', 'avc2', 'avc3' and 'avc4', the width and height fields in the sample entry document the cropped frame dimensions of the AVC base layer. For the MVC sample entries

'mvc1', 'mvc2', 'mvc3', and 'mvc4', and for the MVD sample entries 'mvd1', 'mvd2', 'mvd3', 'mvd4', 'a3d1', 'a3d2', 'a3d3', and 'a3d4', the width and height fields in the sample entry document the cropped frame dimensions achieved by decoding any single texture view of the entire MVC or MVD stream. Furthermore, for MVD sample entries 'mvd1', 'mvd2', 'mvd3', 'mvd4', 'a3d1', 'a3d2', 'a3d3', and 'a3d4', the depth_width and depth_height in the MVDDepthResolutionBox document the cropped frame dimensions achieved by decoding any single depth view of the entire MVD stream.

The `lengthSizeMinusOne` field in the AVC, MVC, MVCD, and A3D configurations in any given sample entry shall have the same value.

A priority assignment URI provides the name (in the URI space) of a method used to assign `priority_id` values. When it occurs in an AVC or MVC sample entry, exactly one URI shall be present, that documents the `priority_id` assignments in the entire AVC or MVC stream. An MVCD or A3D sample entry shall not contain an `MVCViewPriorityAssignmentBox` (and consequently priority assignment URI can be used neither for MVC+D streams nor for 3D-AVC streams).

The URI is treated here as a name only; it should be de-referenceable, though this is not required. File readers may be able to recognize some methods and thereby know what stream extraction operations based on `priority_id` would do.

The requirements for the sample entry types 'avc1' and 'avc2' as documented in 6.5.3.1.1 also apply here.

When present in an AVC, MVC, MVCD, or A3D sample entry, `ViewScalabilityInfoSEIBox`, `ViewIdentifierBox`, `IntrinsicCameraParametersBox`, `ExtrinsicCameraParametersBox`, `MVDScalabilityInformationSEIBox`, `BitRateBox` and `MPEG4ExtensionDescriptorsBox`, apply to the entire AVC, MVC, MVC+D, or 3D-AVC stream, respectively.

The parameter sets required to decode a NAL unit that is present in the sample data of a video stream, either directly or by reference from an Extractor, shall be present in the decoder configuration of that video stream or in the associated parameter set stream (if used).

Table 8 shows for a video track all the possible uses of sample entries, configurations, and the MVC tools (excluding timed metadata, which is always used in another track).

Table 8 — Use of sample entries for AVC, MVC, MVC+D and 3D-AVC tracks

sample entry name	with configuration records	meaning
'avc1' or 'avc3'	AVC Configuration Only	A plain AVC track with AVC NAL units only; Extractors, aggregators, and tier grouping shall not be present.
'avc2' or 'avc4'	AVC Configuration Only	A plain AVC track with AVC NAL units only; Extractors may be present; Aggregators may be

		present to contain and reference AVC NAL units; Tier grouping may be present.
'avc1' or 'avc3'	AVC and MVC Configurations	An MVC track with both AVC and MVC NAL units; Aggregators and Extractors shall not be present.
'avc2' or 'avc4'	AVC and MVC Configurations	An MVC track with both AVC NAL units and MVC NAL units; Extractors may be present and used to reference both AVC and MVC NAL units; Aggregators may be present to contain and reference both AVC and MVC NAL units; Tier grouping may be present.
'mvc1' or 'mvc3'	MVC Configuration Only	An MVC track without AVC NAL units; Aggregators may be present to contain and reference MVC NAL units; Tier grouping may be present.
'mvc2' or 'mvc4'	MVC Configuration Only	An MVC track without AVC NAL units; Extractors may be present and used to reference MVC NAL units; Aggregators may be present to contain and reference MVC NAL units; Tier grouping may be present.
'avc1' or 'avc3'	AVC, MVC, and MVCD Configurations	An MVC+D track with AVC, MVC and MVC+D depth NAL units; Aggregators and extractors shall not be present; Tier grouping may be present.
'mvc1' or 'mvc3'	MVC and MVCD Configurations	An MVC+D track without AVC NAL units but with MVC and MVC+D depth NAL units; Aggregators and tier grouping may be present.
'mvc2' or 'mvc4'	MVC and MVCD Configurations	An MVC+D track without AVC NAL units but with MVC and MVC+D depth NAL units; Extractors, aggregators and tier grouping may be present.
'mvd1' or 'mvd3'	MVCD Configuration Only	An MVC+D track with MVC+D depth NAL units only; Aggregators and tier grouping may be present.
'mvd2' or 'mvd4'	MVCD Configuration Only	An MVC+D track with MVC+D depth NAL units only; Extractors, aggregators and tier grouping may be present.
'avc1' or 'avc3'	AVC, MVC, MVCD, and A3D Configurations	A 3D-AVC track with AVC, MVC, MVC+D depth, and 3D-AVC NAL units; Aggregators and extractors shall not be present; Tier grouping may be present.

		If MVC Configuration were not present in this example, the track would not contain or refer to MVC NAL units. If MVCD Configuration were not present in this example, the track would not contain or refer to MVC+D depth NAL units.
'mvc1' or 'mvc3'	MVC, MVCD, and A3D Configurations	A 3D-AVC track without AVC NAL units but with MVC, MVC+D depth NAL units, and 3D-AVC NAL units; Aggregators and tier grouping may be present. If MVCD Configuration were not present in this example, the track would not contain or refer to MVC+D depth NAL units.
'mvc2' or 'mvc4'	MVC, MVCD, and A3D Configurations	A 3D-AVC track without AVC NAL units but with MVC, MVC+D depth NAL units, and 3D-AVC NAL units; Extractors, aggregators and tier grouping may be present. If MVCD Configuration were not present in this example, the track would not contain or refer to MVC+D depth NAL units.
'mvd1' or 'mvd3'	MVCD and A3D Configurations	A 3D-AVC track without AVC or MVC NAL units but with MVC+D depth NAL units and 3D-AVC NAL units; Aggregators and tier grouping may be present.
'mvd2' or 'mvd4'	MVCD and A3D Configurations	A 3D-AVC track without AVC or MVC NAL units but with MVC+D depth NAL units and 3D-AVC NAL units; Extractors, aggregators and tier grouping may be present.
'a3d1' or 'a3d3'	A3D Configuration Only	A 3D-AVC track with 3D-AVC NAL units only; Aggregators and tier grouping may be present.
'a3d2' or 'a3d4'	A3D Configuration Only	A 3D-AVC track with 3D-AVC NAL units only; Extractors, aggregators and tier grouping may be present.

7.6.3.2.2 Syntax

```

class MVConfigurationBox extends Box('mvcC') {
    MVCDConfigurationRecord() MVCDConfig;
}

class ViewScalabilityInformationSEIBox extends Box('vsib', size)
{
    unsigned int(8*size-64) mvscalinfosei;
}

```

```

class MVDDepthResolutionBox extends Box('3dpr')
{
    unsigned int(16) depth_width;
    unsigned int(16) depth_height;
    /* The following 5 fields are collectively optional; they are either all present
    or all absent. When grid_pos_num_views is not present, the for loop is not
    present, equivalent to grid_pos_num_views equal to 0. These fields may be present
    or absent whenever the box is present (e.g., in MVCDConfigurationBox or
    A3DConfigurationBox). */
    unsigned int(16) depth_hor_mult_minus1; // optional
    unsigned int(16) depth_ver_mult_minus1; // optional
    unsigned int(4) depth_hor_rsh; // optional
    unsigned int(4) depth_ver_rsh; // optional
    unsigned int(16) grid_pos_num_views; // optional
    for(i = 0; i < grid_pos_num_views; i++) {
        bit(6) reserved=0;
        unsigned int(10) grid_pos_view_id[i];
        signed int(16) grid_pos_x[grid_pos_view_id[i]];
        signed int(16) grid_pos_y[grid_pos_view_id[i]];
    }
}

class MVCDConfigurationBox extends Box('mvdC') {
    MVDDecoderConfigurationRecord MVDConfig;
    MVDDepthResolutionBox mvdDepthRes; //Optional
}

class A3DConfigurationBox extends Box('a3dC') {
    MVDDecoderConfigurationRecord MVDConfig;
    MVDDepthResolutionBox mvdDepthRes; //Optional
}

class MVDSscalabilityInformationSEIBox extends Box('3sib', size)
{
    unsigned int(8*size-64) mvdscaleinfosei;
}

class AVCMVCSampleEntry() extends AVCSampleEntry ('avc1' or 'avc3') {
    ViewScalabilityInformationSEIBox scalability; // optional
    ViewIdentifierBox view_identifiers; // optional
    MVCConfigurationBox mvccconfig; // optional
    MVCViewPriorityAssignmentBox view_priority_method; // optional
    IntrinsicCameraParametersBox intrinsic_camera_params; // optional
    ExtrinsicCameraParametersBox extrinsic_camera_params; // optional
    MVCDConfigurationBox mvcdconfig; // optional
    MVDSscalabilityInformationSEIBox mvdscaleinfosei; // optional
    A3DConfigurationBox a3dconfig; // optional
}

class AVC2MVCSampleEntry() extends AVC2SampleEntry ('avc2' or 'avc4') {
    ViewScalabilityInformationSEIBox scalability; // optional
    ViewIdentifierBox view_identifiers; // optional
    MVCConfigurationBox mvccconfig; // optional MVCViewPriorityAssignmentBox
    view_priority_method; // optional
    IntrinsicCameraParametersBox intrinsic_camera_params; // optional
    ExtrinsicCameraParametersBox extrinsic_camera_params // optional
    MVCDConfigurationBox mvcdconfig; // optional
    MVDSscalabilityInformationSEIBox mvdscaleinfosei; // optional
    A3DConfigurationBox a3dconfig; // optional
}

```

```

// Use this if the track is NOT AVC compatible
class MVCSampleEntry() extends VisualSampleEntry ('mvc1', 'mvc2',
                                                    'mvc3', or 'mvc4') {
    MVCSConfigurationBox mvccconfig; // mandatory
    ViewScalabilityInformationSEIBox scalability; // optional
    ViewIdentifierBox view_identifiers; // mandatory
    MPEG4ExtensionDescriptorsBox descr; // optional
    MVCViewPriorityAssignmentBox view_priority_method; // optional
    IntrinsicCameraParametersBox intrinsic_camera_params; // optional
    ExtrinsicCameraParametersBox extrinsic_camera_params // optional
    MVCDConfigurationBox mvcdconfig; // optional
    MVDSscalabilityInformationSEIBox mvdscaleinfosei; // optional
    A3DConfigurationBox a3dconfig; // optional
}

class MVCDSampleEntry() extends VisualSampleEntry ('mvd1', 'mvd2',
                                                    'mvd3', or 'mvd4') {
    MVCDConfigurationBox mvcdconfig; // mandatory
    MVDSscalabilityInformationSEIBox mvdscaleinfosei; // optional
    ViewIdentifierBox view_identifiers; // mandatory
    MPEG4ExtensionDescriptorsBox descr; // optional
    IntrinsicCameraParametersBox intrinsic_camera_params; // optional
    ExtrinsicCameraParametersBox extrinsic_camera_params // optional
    A3DConfigurationBox a3dconfig; // optional
}

class A3DSampleEntry() extends VisualSampleEntry ('a3d1', 'a3d2',
                                                    'a3d3', or 'a3d4') {
    A3DConfigurationBox a3dconfig; // mandatory
    MVDSscalabilityInformationSEIBox mvdscaleinfosei; // optional
    ViewIdentifierBox view_identifiers; // mandatory
    MPEG4ExtensionDescriptorsBox descr; // optional
    IntrinsicCameraParametersBox intrinsic_camera_params; // optional
    ExtrinsicCameraParametersBox extrinsic_camera_params // optional
}

```

7.6.3.2.3 Semantics

When the sample entry is 'mvc1', 'mvc2', 'mvc3', or 'mvc4', Compressorname in the base class VisualSampleEntry indicates the name of the compressor used, with the value "\012MVC Coding" being recommended (\012 is 10, the length of the string "MVC coding" in bytes).

When the sample entry is 'mvd1', 'mvd2', 'mvd3', or 'mvd4', Compressorname in the base class VisualSampleEntry indicates the name of the compressor used, with the value "\013MVCD Coding" being recommended (\013 is 11, the length of the string "MVCD coding" in bytes).

When the sample entry is 'a3d1', 'a3d2', 'a3d3', or 'a3d4', Compressorname in the base class VisualSampleEntry indicates the name of the compressor used, with the value "\012A3D Coding" being recommended (\012 is 10, the length of the string "A3D coding" in bytes).

depth_width and depth_height give the values of the width and height, respectively, of cropped frame dimensions of the coded depth view components, in pixels, in the stream to which the sample entry containing the MVDDepthResolutionBox() is included applies.

`depth_hor_mult_minus1`, `depth_hor_rsh`, `depth_ver_mult_minus1`, and `depth_ver_rsh` are specified identically to the semantics of the syntax elements with the same names in Annex J of ISO/IEC 14496-10:2020. When not present, `depth_hor_mult_minus1` and `depth_ver_mult_minus1` are inferred to be equal to 1, and `depth_hor_rsh` and `depth_ver_rsh` are inferred to be equal to 0.

NOTE 1 In MVC+D bitstreams, the depth sampling information SEI message carries information corresponding to `depth_hor_mult_minus1`, `depth_hor_rsh`, `depth_ver_mult_minus1`, and `depth_ver_rsh`.

NOTE 2 `depth_hor_mult_minus1` and `depth_hor_rsh` specify the ratio of the width of the luma sample in a texture view component relative to the width of the luma sample in a depth view component. `depth_ver_mult_minus1` and `depth_ver_rsh` specify the ratio of the height of the luma sample in a texture view component relative to the height of the luma sample in a depth view component.

`grid_pos_num_views` specifies the number of views for which `grid_pos_view_id[i]`, `grid_pos_x[grid_pos_view_id[i]]` and `grid_pos_y[grid_pos_view_id[i]]` are present.

`grid_pos_view_id[i]` specifies a `view_id` value of a texture view.

`grid_pos_x[grid_pos_view_id[i]]` specifies a horizontal offset of a depth sampling grid relative to the luma texture sampling grid in texture luma sample units.

`grid_pos_y[grid_pos_view_id[i]]` specifies a vertical offset of a depth sampling grid relative to the luma texture sampling grid in texture luma sample units. When no value of `grid_pos_view_id[i]` is equal to a `view_id` value of a texture view, `grid_pos_x[view_id]` and `grid_pos_y[view_id]` are inferred to be equal to 0.

NOTE 3 `grid_pos_num_views`, `grid_pos_view_id[i]`, `grid_pos_x[grid_pos_view_id[i]]` and `grid_pos_y[grid_pos_view_id[i]]` are specified identically to the semantics of the syntax elements with the same names in Annex J of ISO/IEC 14496-10:2020. In MVC+D bitstreams, the depth sampling information SEI message carries information corresponding to `depth_hor_mult_minus1`, `depth_hor_rsh`, `depth_ver_mult_minus1`, and `depth_ver_rsh`.

`mvdDepthRes` contains the width and height of the coded depth view components in the stream to which the sample entry applies. When not present, the width and height of the coded depth view components are inferred to be the same as the width and height of the coded texture view components.

`mvcscalinfosei` contains an SEI NAL unit containing only a view scalability information SEI message as specified in ISO/IEC 14496-10:2020 Annex H. The 'size' field of the container box `ViewScalabilityInformationSEIBox` shall not be equal to 0 or 1.

`mvdscalinfosei` contains an SEI NAL unit containing only a MVCD scalability information SEI message as specified in ISO/IEC 14496-10:2020 Annex I. The 'size' field of the container box `MVDSscalabilityInformationSEIBox` shall not be equal to 0 or 1.

7.6.4 Sync sample

A sync sample identifies the presence of an IDR access unit of the MVC or MVD bitstream for any sample entry that includes an MVC or MVD configuration record, respectively.

For video data described by a sample entry of type 'mvc3', 'mvc4', 'mvd3', 'mvd4', 'a3d3', or 'a3d4', the following applies:

- a) If the sample is a sync sample, all parameter sets needed for decoding that sample shall be included either in the sample entry or in the sample itself.
- b) Otherwise (the sample is not sync sample), all parameter sets 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.

7.6.5 Shadow sync

A shadow sync box shall not be used for video data described by any MVC or MVD sample entry. Its use for MVC or MVD is deprecated.

7.6.6 Independent and disposable samples box

If the SampleDependencyTypeBox is used in a track that is both AVC and MVC compatible, then care should be taken that the information provided by this box is true no matter what valid subset of the MVC data (possibly only the AVC data) is used. The 'unknown' values (value 0 of the fields sample_depends_on, sample_is_depended_on, and sample_has_redundancy) may be needed if the information varies.

If the SampleDependencyTypeBox is used in a track that is compatible to all of AVC, MVC, and MVD, then care should be taken that the information provided by this box is true no matter what valid subset of the MVD data (possibly only the AVC data or only the MVC data) is used. The 'unknown' values (value 0 of the fields sample_depends_on, sample_is_depended_on, and sample_has_redundancy) may be needed if the information varies.

7.6.7 Sample groups on random access recovery points 'rol1' and random access points 'rap'

When version 0 of the SampleToGroupBox is used, the following applies:

- For video data described by a sample entry of type 'avc1', 'avc2', 'avc3' or 'avc4', the random access recovery sample group and the random access point sample group identify random access recovery points and random access points, respectively, for all of an AVC decoder, an MVC decoder (if any), and an MVD decoder (if any) operating on the entire bitstream.

NOTE If the random access recovery points or the random access points for the AVC decoder, the MVC decoder, and the MVD decoder operating on the entire bitstream are not all aligned, the random access recovery points table or the random access point table, respectively, will not document all of them. In this case, the stream can be stored in multiple tracks, e.g. two tracks, one containing the AVC base layer with a sample entry of type 'avc1' or 'avc3', and the other containing other layers with an MVC or MVD sample entry type.

- For video data described by an MVC sample entry type, the information provided by the random access recovery sample group and the random access point sample group is true for any valid subset of the entire MVC bitstream.

- For video data described by an MVD sample entry type, the information provided by the random access recovery sample group and the random access point sample group is true for any valid subset of in the entire MVD bitstream.

When version 1 of the `SampleToGroupBox` is used, the `grouping_type_parameter` specifies the `tier_id` value of the layer(s) or view(s) that are refreshed in the associated sample.

7.7 MVC specific information boxes

7.7.1 Overview

The following boxes specify information that relate to more than one output view of an MVC or MVD elementary stream. As any subset of views of an MVC or MVD elementary stream can be chosen for output, the information carried in these boxes is not necessarily specific to any track and thus contained separately. The information can be specified for different groups of output views.

7.7.2 Multiview information box

7.7.2.1 Definition

Box Type: `mvci`
 Container: Media Information Box ('minf')
 Mandatory: No
 Quantity: Zero or one

Located in the Media Information Box of the base view track indicated by the 'sbas' track reference, this box contains Multiview Group boxes, and Multiview Group Relation boxes.

7.7.2.2 Syntax

```
aligned(8) class MultiviewInformationBox
    extends FullBox('mvci', version = 0, flags) {
}
```

7.7.3 Multiview group box

7.7.3.1 Definition

Box Type: 'mvcg'
 Container: Multiview Information box ('mvci')
 Mandatory: No
 Quantity: Zero or more

This box specifies a multiview group for the views of the MVC or MVD stream that are output. Target output views can be indicated on the basis of `track_id`, `tier_id`, or `view_id`. When the views included in a track match an operating point, it is recommended to use `track_id` (i.e., `entry_type` equal to 0) within the Multiview Group box. When multiview sample grouping is in use, and tiers cover more than one view or some tiers contain a temporal subset of the bitstream, it is recommended to use `tier_id` (i.e., `entry_type` equal to 1) within the Multiview Group box. Otherwise, it is recommended to use one of the `view_id` based indications (i.e., `entry_type` equal to 2 or 3).

When `entry_type` is equal to 0 or 1, the following applies. Each view in a track or tier that is included in this box is a target output view, and if a track or tier included in this box contains multiple views, all the contained views are target output views.

Decoding of the output views may require decoding of other views that are not target output views. The views that are required for decoding but are not target output views can be concluded from reference view identifiers included in the View Identifier box, the 'scal' track references, or from the Tier Dependency box.

If the box contains a `track_id` or `tier_id` that is not present or refers to a `view_id` of a view that is not present, the respective view should be considered removed and the multiview group should be ignored.

7.7.3.2 Syntax

```
aligned(8) class MultiviewGroupBox extends FullBox('mvcg', version = 0, flags) {
    unsigned int(32) multiview_group_id;
    unsigned int(16) num_entries;
    bit(8) reserved = 0;
    for(i=0; i<num_entries; i++) {
        unsigned int(8) entry_type;
        if (entry_type == 0)
            unsigned int(32) track_id;
        else if (entry_type == 1) {
            unsigned int(32) track_id;
            unsigned int(16) tier_id;
        }
        else if (entry_type == 2) {
            bit(6) reserved1 = 0;
            unsigned int(10) output_view_id;
        }
        else if (entry_type == 3) {
            bit(6) reserved2 = 0;
            unsigned int(10) start_view_id;
            unsigned int(16) view_count;
        }
    }
    TierInfoBox subset_stream_info; // optional
    MultiviewRelationAttributeBox relation_attributes; // optional
    TierBitRateBox subset_stream_bit_rate; // optional
    BufferingBox subset_stream_buffering; // optional
    MultiviewSceneInfoBox multiview_scene_info; // optional
}
```

7.7.3.3 Semantics

`multiview_group_id` provides a unique identifier for the multiview group within the file.

`num_entries` is the number of tracks (entry type 0), tiers (entry type 1), target output views (entry type 2), or continuous sets of target output views (entry type 3) included in this multiview group.

`entry_type` specifies how the target output views are indicated. The following values of `entry_type` are specified:

0 – all the views included in an indicated track are target output views

- 1 – the view(s) of an indicated tier within an indicated track are target output views
- 2 – the view with view_id equal to output_view_id is a target output view
- 3 – the views having view_id within the range of start_view_id to (start_view_id + view_count – 1), inclusive, are target output views

track_id indicates a track containing target output views.

tier_id indicates a tier within a track where all views within the tier are target output views.

output_view_id indicates a view_id of a target output view.

start_view_id indicates the first view_id in a range of contiguous values of view_id all being target output views.

view_count indicates the number of contiguous values of view_id all being target output views.

track_id indicates a track.

tier_id indicates a tier within a track.

subset_stream_info indicates the characteristics of the bitstream subset containing the indicated output views and the views they depend on.

relation_attributes indicate the relations between output views. If 'ecam' is used as a common attribute, all the output views are associated with extrinsic camera parameters indicating that the cameras have identical rotation and constant spacing. If 'ecam' is used as a differentiating attribute, at least one output view is associated with extrinsic camera parameters with different rotation from the others or the output views are associated with extrinsic camera parameters not having a constant spacing.

subset_stream_bit_rate indicates the bit rate statistics of the bitstream subset containing the indicated output views and the views they depend on. The values of tierBaseBitRate, tierMaxBitRate, and tierAvgBitRate within the TierBitRateBox are unspecified.

subset_stream_buffering indicates the HRD parameters that apply to the bitstream subset containing the indicated output views and the views they depend on and operating with the indicated target output views.

multiview_scene_info contains the maximum disparity in units of integer pixel resolution between any spatially adjacent output views in any access unit.

7.7.4 Multiview group relation box

7.7.4.1 Definition

Box Type: 'swtc'
 Container: Multiview Information box ('menci')
 Mandatory: No
 Quantity: Zero or more

This box specifies a set of multiview groups from which one multiview group is decoded and played at any time. The given relation attributes specify which features are common in all associated multiview groups and which factors make the multiview groups differ from each other. The relation attributes can be used to select a suitable set of multiview groups for playback, e.g. based on the number of output views. The differentiating attributes can be used to select which multiview group within the set is suitable for the player, e.g. based on the required level for decoding.

7.7.4.2 Syntax

```
aligned(8) class MultiviewGroupRelationBox() extends FullBox('swtc', version = 0,
flags) {
    unsigned int(32) num_entries;
    for (i=0; i<num_entries; i++)
        unsigned int(32) multiview_group_id;
    MultiviewRelationAttributeBox relation_attributes;
}
```

7.7.4.3 Semantics

`num_entries` indicates the number of associated multiview groups.
`multiview_group_id` is the identifier of an associated multiview group.
`relation_attributes` indicate the relations between the associated multiview groups.

7.7.5 Multiview relation attribute box

7.7.5.1 Definition

Box Type:	<code>`mvra'</code>
Container:	MultiviewGroupBox or MultiviewGroupRelationBox
Mandatory:	No in MultiviewGroupBox, Yes in MultiviewGroupRelationBox
Quantity:	Zero or One in MultiviewGroupBox One in MultiviewGroupRelationBox

When the Multiview Relation Attribute box is contained in a Multiview Group box, it indicates the relation of the output views of the respective multiview group with each other. When the Multiview Relation Attribute box is contained in a Multiview Group Relation box, it indicates the relation of the multiview groups with each other.

The Multiview Relation Attribute box contains common and differentiating attributes. When the Multiview Relation Attribute box is included in a Multiview Group box, a common attribute indicates a characteristic that is common for each one of the target output views of the multiview group and a differentiating attribute indicates a characteristic that differs in at least one of one of the target output views of the multiview group. When Multiview Relation Attribute box is included in a Multiview Group Relation box, a common attribute indicates a characteristic that is common for the indicated multiview groups or for the respective target output views in each one of the indicated multiview groups, whereas a differentiating attribute indicates a characteristic that differs in at least one of the indicated multiview groups or at least one of the respective target output views in the indicated multiview groups.

A common attribute is associated with an additional parameter, which carries the value of the common attribute. The syntax and semantics of the additional parameter depend on the attribute in question.

For example, a file writer can create a Multiview Group for each stereo pair suitable for display from a multiview bitstream. Furthermore, a file writer can create a Multiview Group Relation box listing all the multiview groups for stereo pair output and including a Multiview Relation Attribute box with common attributes number of views (equal to 2) and in-line camera arrangement. A file reader can study the Multiview Group Relation box to find the options for stereo pair output and choose one multiview group for processing. Note that the presence of views in a group does not necessarily imply they are all

suggested as output views at any given time – the terminal may choose which views to output, and it is not limited by the group information.

7.7.5.2 Syntax

```
aligned(8) class MultiviewRelationAttributeBox
  extends FullBox('mvra', version = 0, flags) {
    bit(16) reserved1 = 0;
    unsigned int(16) num_common_attributes;
    for (i=0; i<num_common_attributes; i++) {
        unsigned int(32) common_attribute;
        unsigned int(32) common_value;
    }
    bit(16) reserved2 = 0;
    unsigned int(16) num_differentiating_attributes;
    for (i=0; i<num_differentiating_attributes; i++)
        unsigned int(32) differentiating_attribute;
}
```

7.7.5.3 Semantics

`common_attribute` and `differentiating_attribute` are selected from the list below.

Attributes that can be used as a differentiating attribute are associated with a distinguishing pointer to the field or information.

`common_value` specifies the value for the common attribute. Its syntax and semantics depend on the common attribute and are specified in the table below.

Name	Attribute	Pointer and semantics	<i>common_value</i> syntax and semantics
Profile	'prfl'	This attribute shall not be included in the Multiview Group box. When included in the Multiview Group Relation box, the attribute refers to the profile required for decoding the bitstream subset corresponding to the multiview group. The attribute points to the <code>profileIndication</code> field of the <code>subset_stream_info</code> element of the Multiview Group box.	<code>bit(24) reserved = 0;</code> <code>unsigned int(8)</code> <code>profileIndication;</code> <code>profileIndication</code> is the profile sufficient for decoding the bitstream subset corresponding to all indicated multiview groups.
Level	'levl'	This attribute shall not be included in the Multiview Group box. When included in the Multiview Group Relation box, the attribute refers to the level required for decoding the bitstream subset corresponding to the multiview group. The attribute points to the <code>levelIndication</code> field of the <code>subset_stream_info</code> element of the Multiview Group box.	<code>bit(24) reserved = 0;</code> <code>unsigned int(8)</code> <code>levelIndication;</code> <code>profileIndication</code> is the level sufficient for decoding the bitstream subset corresponding to all indicated multiview groups, or 0 if the level is unspecified.

Bitrate	'bitr'	This attribute shall not be included in the Multiview Group box. When included in the Multiview Group Relation box, the attribute refers to the total size of bitstream subset required for decoding of the multiview group divided by the duration in the track header box. The attribute points to the avgBitRate field of the subset_stream_bit_rate element of the Multiview Group box, if present, or a value that would be contained in the avgBitRate field of the subset_stream_bit_rate element of the Multiview Group box, if it were present.	unsigned int(32) bitrate; bitrate indicates the average bit rate in bits per second of the bitstream subset required for decoding the multiview group. The bitrate may be rounded up.
Frame rate	'frar'	This attribute shall not be included in the Multiview Group box. When included in the Multiview Group Relation box, the attribute refers the number of samples in the track divided by duration in the track header box.	unsigned int(16) integer_part; bit(16) reserved = 0; integer_part shall be equal to the output rate of decoded access units in second rounded to the closest integer using the Round function specified in ISO/IEC 14496-10.
Number of output views	'nvws'	Number of target output views indicated in the Multiview Group Box ('mvcg') If this attribute is included in the Multiview Group box, it shall be a common attribute and merely documents the number of output views in the respective multiview group.	unsigned int(32) num_views; num_views indicates the number of views in the multiview group.
Intrinsic camera parameters	'icam'	The intrinsic camera parameters are stored in 'avc1', 'avc2', 'avc3', 'avc4', 'mvc1', 'mvc2', 'mvc3', 'mvc4', 'mvd1', 'mvd2', 'mvd3', 'mvd4', 'a3d1', 'a3d2', 'a3d3', or 'a3d4' Sample Entry (in Sample entry box of media track). If this attribute is included in the Multiview Group box and used as a common attribute, the intrinsic camera parameters of the target output views are identical. If this attribute is included in	Unspecified.

the Multiview Group box and used as a differentiating attribute, the intrinsic camera parameters of the target output views differ at least partly.

If this attribute is included in the Multiview Group Relation box and used as a common attribute, the number of target output views in all indicated multiview groups shall be the same and the intrinsic camera parameters of the respective target output views in all indicated multiview groups are identical. If this attribute is included in the Multiview Group Relation box and used as a differentiating attribute, the intrinsic camera parameters of the respective target output views differ at least partly.

Extrinsic
camera
parameters

'ecam'

The extrinsic camera parameters are Unspecified. stored in 'avc1', 'avc2', 'avc3', 'avc4', 'mvc1', 'mvc2', 'mvc3', 'mvc4', 'mvd1', 'mvd2', 'mvd3', 'mvd4', 'a3d1', 'a3d2', 'a3d3', or 'a3d4' Sample Entry (in Sample entry box of media track).

If this attribute is included in the Multiview Group box and used as a common attribute, the rotation of the cameras for all the target output views is the same and, if the cameras are arranged in linear, elliptical, or rectangular arrangement, the distance of adjacent cameras is the same. If this attribute is included in the Multiview Group box and used as a differentiating attribute, the rotation or the distance of adjacent cameras in linear, elliptical, or rectangular arrangement differs.

If the attribute is included in the Multiview Group Relation box and used as a common attribute, the relative extrinsic camera parameters target output views in all indicated multiview groups are identical. That is, the distance of cameras relative to each other and

their rotation matches in the indicated multiview groups. If the attribute is included in the Multiview Group Relation box and used as a differentiating attribute, the relative extrinsic camera parameters of respective target output views differ at least partly.

Inline view array 'ilvi'

If used as a common attribute, the associated cameras are located on a straight line.

When included in a Multiview Group box, the attribute shall be a common attribute.

```
bit(28) reserved = 0;
unsigned int(2)
horizontal_order;
unsigned int(2)
vertical_order;
```

horizontal_order
indicates the
horizontal order of the
views:

- 0: the views are in the
same horizontal
location
- 1: the views are ordered
left-to-right
- 2: the views are ordered
right-to-left
- 3: the order of the views
is undefined, or left and
right are not well-
defined.

vertical_order
indicates the vertical
order of the views:

- 0: the views are in the
same vertical location
- 1: the views are ordered
bottom-to-top
- 2: the views are ordered
top-to-bottom
- 3: the order of the views
is undefined, or top and
bottom are not well-
defined.

Rectangular view array 'rtvi'

If used as a common attribute, the associated cameras form a rectangular shape and are regularly spaced along the orthogonal coordinate axes.

When included in a Multiview Group box, the attribute shall be a common attribute.

```
unsigned int(16)
row_view_count;
unsigned int(16)
col_view_count;
```

row_view_count
specifies the number of

rows in the rectangular array.

`col_view_count` specifies the number of columns in the rectangular array.

The views are indicated in raster scan order in the Multiview Group box.

Unspecified.

Planar view array 'plvi'

If used as a common attribute, the associated cameras are located on a plane, but may be irregularly spaced.

When included in a Multiview Group box, the attribute shall be a common attribute.

Elliptical view array 'elvi'

If used as a common attribute, the associated cameras are located on the arc of an ellipse.

When included in a Multiview Group box, the attribute shall be a common attribute.

```
bit(28) reserved = 0;
unsigned int(2)
horizontal_order;
unsigned int(2)
vertical_order;
```

The semantics are identical to those for the Inline view array.

Spherical view array 'spvi'

If used as a common attribute, the associated cameras are located on the surface of a sphere.

When included in a Multiview Group box, the attribute shall be a common attribute.

Unspecified.

Stereo view array 'stvi'

If used as a common attribute, the associated cameras are a pair of views suitable for stereo viewing.

When included in a Multiview Group box, the attribute shall be a common attribute.

```
bit(6) reserved1 = 0;
unsigned int(10)
left_view_id;
bit(6) reserved2 = 0;
unsigned int(10)
right_view_id;
```

Geometry 'geom'

If used as a differentiating attribute, indicates that the views or groups of views belong to different view arrangements (e.g. inline, planar, etc.)

Unspecified.

7.7.6 Multiview scene info box

7.7.6.1 Definition

Box Type:	'vwdi'
Container:	Multiview Group box ('mvcg')
Mandatory:	No
Quantity:	Zero or one

An optional Multiview Scene Info Box includes the maximum disparity between the adjacent views of the respective multiview group. This information can be used for processing the multiview video prior to rendering on a 3D display.

NOTE A Multiview Scene Information SEI message, as specified in MVC H.12.1.5, can indicate the maximum disparity between any adjacent views in the bitstream. Thus, the Multiview Scene Info Box represents similar information as carried in the Multiview Scene Information SEI message but is limited to a certain set of views rather than concerns all the views in the bitstream.

The Multiview Scene Info Box shall not be present for multiview groups associated with cameras that do not form a one-dimensional arrangement, such as a line or an arc of an ellipse.

7.7.6.2 Syntax

```
class MultiviewSceneInfoBox extends Box ('vwci')
{
    unsigned int(8)    max_disparity;
}
```

7.7.6.3 Semantics

`max_disparity` specifies the maximum disparity in units of integer luma samples between the spatially adjacent view components (within an access unit) in this multiview group. This information can be used for processing the multiview video prior to rendering on a 3D display.

7.7.7 MVC view priority assignment box

7.7.7.1 Definition

Box Type:	'mvp'
Container:	Sample Entry ('avc1', 'avc2', 'avc3', 'avc4', 'mvc1', 'mvc2', 'mvc3', 'mvc4', 'mvd1', 'mvd2', 'mvd3', 'mvd4')
Mandatory:	No
Quantity:	Zero or one

A priority assignment URI provides the name (in the URI space) of a method used to assign content priority values in the View Priority sample grouping. The URI is treated here as a name only; it should be de-referenceable, though this is not required. File readers may be able to recognize some methods and thereby know what stream extraction or selection of output views based on particular content priority values would do.

7.7.7.2 Syntax

```
class MVCViewPriorityAssignmentBox extends Box('mvcP')
{
    unsigned int(8)    method_count;
    string PriorityAssignmentURI[method_count];
}
```

7.7.7.3 Semantics

`method_count` provides a count of the number of following URIs.

`PriorityAssignmentURI` provides a unique name of the method used to assign `content_priority_id` values in View Priority sample groupings. In the case of absence of this box, the priority assignment method is unknown.

8 HEVC elementary streams and sample definitions

8.1 Overview

The High Efficiency Video Coding (HEVC) standard, jointly developed by the ITU-T and ISO/IEC, offers not only increased coding efficiency and enhanced robustness, but also many features for the systems that use it. To enable the best visibility of, and access to, those features, and to enhance the opportunities for the interchange and interoperability of media.

This clause specifies the storage format for single-layer HEVC (ISO/IEC 23008-2) video streams.

The storage of HEVC content uses the existing capabilities of the ISO base media file format but also defines extensions to support the following features of the HEVC codec.

- a) The video, sequence and picture parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded block data. Each slice containing the coded block data references the picture parameter set containing its decoding parameters. In turn, the picture parameter set references a sequence parameter set that contains sequence level decoding parameter information, and the sequence parameter set references a video parameter set that contains global decoding parameter information (across layers).

This document includes the following tools for supporting of HEVC contents:

- a) Temporal scalability sample grouping indicates the association of access units with different hierarchy levels of temporal scalability.
- b) Temporal sub-layer access sample grouping indicates the identification of access units as temporal sub-layer access (TSA) samples.
- c) Step-wise temporal sub-layer access sample grouping indicates the identification of access units as step-wise temporal sub-layer access (STSA) samples.

8.2 Elementary stream structure

In Clause 8, a video stream is represented by one video track in a file.

NOTE Some HEVC streams as defined in ISO/IEC 23008-2 include data that can be stored alternatively according to Clause 9 or 10.

Two types of elementary streams are defined for storing HEVC content:

- **Video Elementary Stream** that does not contain any parameter sets; all parameter sets are stored in a sample entry or sample entries;
- **Video and Parameter set elementary stream** that may contain parameter sets, and may also have parameter sets stored in their sample entry or sample entries.

8.3 Sample and configuration definition

8.3.1 Canonical order and restrictions

The canonical stream format is an HEVC elementary stream that satisfies the following conditions in addition to the general conditions in 4.2.2:

- All VCL NAL units for a single picture shall be contained within the sample whose decoding time and composition time are those of the picture.
- **Access unit delimiter NAL units:** The constraints obeyed by access unit delimiter NAL units are defined in ISO/IEC 23008-2.
- **Parameter sets:** For a video stream that a particular sample entry applies to, the video parameter set, sequence parameter sets, and picture parameter sets, shall be stored only in the sample entry when the sample entry name is 'hvc1', and may be stored in the sample entry and the samples when the sample entry name is 'hev1'.

NOTE 1 A parameter set to be used in a picture needs to be sent prior to the sample containing that picture or in the sample for that picture.

NOTE 2 Storing parameter sets in the sample entries of a video stream provides a simple and static way to supply parameter sets. Storing parameters in samples on the other hand is more complex but allows for more dynamism in the case of parameter set updates (a particular parameter set's content is changed but using the same ID) and in the case of adding additional parameter sets. A decoder initializes with the parameter sets in the sample entry, and then updates using the parameter sets as they occur in the stream, starting from any sample marked as a sync sample. Such updating can replace parameter sets with a new definition using the same identifier. Each time the sample entry changes, the decoder re-initializes with the parameter sets included in the sample entry.

- **SEI messages:** SEI messages of declarative nature may be stored in the sample entry; there is no prescription about removing such SEI messages from the samples.
- **Filler data.** Video data is naturally represented as variable bit rate in the file format and should be filled for transmission if needed. Filler Data NAL units and Filler Data SEI messages shall not be present in the file format stored stream when the sample entry does not also permit in-stream parameter sets.

NOTE 3 The removal or addition of Filler Data NAL units, start codes, SEI messages or Filler Data SEI messages can change the bitstream characteristics with respect to conformance with the HRD when operating the HRD in CBR mode as specified in ISO/IEC 23008-2:2020, Annex C.

8.3.2 Decoder configuration information

8.3.2.1 HEVC decoder configuration record

8.3.2.1.1 Definition

This subclause specifies the decoder configuration information for ISO/IEC 23008-2 video content.

This record contains the size of the length field used in each sample to indicate the length of its contained NAL units as well as the parameter sets, if stored in the sample entry. This record is externally framed (its size is supplied by the structure that contains it).

This record contains a version field. This version of the specification defines version 1 of this record. Incompatible changes to the record will be indicated by a change of version number. Readers shall not attempt to decode this record or the streams to which it applies if the version number is unrecognized.

Compatible extensions to this record will extend it and will not change the configuration version code. Readers should be prepared to ignore unrecognized data beyond the definition of the data they understand.

The values for `general_profile_space`, `general_tier_flag`, `general_profile_idc`, `general_profile_compatibility_flags`, `general_constraint_indicator_flags`, `general_level_idc`, `min_spatial_segmentation_idc`, `chroma_format_idc`, `bit_depth_luma_minus8` and `bit_depth_chroma_minus8` shall be valid for all parameter sets that are activated when the stream described by this record is decoded (referred to as "all the parameter sets" in the following sentences in this paragraph). Specifically, the following restrictions apply:

- The value of `general_profile_space` in all the parameter sets shall be identical.
- The tier indication `general_tier_flag` shall indicate a tier equal to or greater than the highest tier indicated in all the parameter sets.
- The profile indication `general_profile_idc` shall indicate a profile to which the stream associated with this configuration record conforms.

NOTE 1 If the sequence parameter sets are marked with different profiles, then the stream could need examination to determine which profile, if any, the entire stream conforms to. If the entire stream is not examined, or the examination reveals that there is no profile to which the entire stream conforms, then the entire stream needs to be split into two or more sub-streams with separate configuration records in which these rules can be met.

- Each bit in `general_profile_compatibility_flags` may only be set if all the parameter sets set that bit.
- Each bit in `general_constraint_indicator_flags` may only be set if all the parameter sets set that bit.
- 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.

- The `min_spatial_segmentation_idc` indication shall indicate a level of spatial segmentation equal to or less than the lowest level of spatial segmentation indicated in all the parameter sets.
- The value of `chroma_format_idc` in all the parameter sets shall be identical.
- The value of `bit_depth_luma_minus8` in all the parameter sets shall be identical.
- The value of `bit_depth_chroma_minus8` in all the parameter sets shall be identical.

Explicit indication is provided in the HEVC Decoder Configuration Record about the chroma format and bit depth as well as other important format information used by the HEVC video elementary stream. Each type of such information shall be identical in all parameter sets, if present, in a single HEVC configuration record. If two sequences differ in any type of such information, two different HEVC sample entries shall be used. If the two sequences differ in color space indications in their VUI information, then two different HEVC sample entries are also required.

There is a set of arrays to carry initialization NAL units. The NAL unit types are restricted to indicate SPS, PPS, VPS, prefix SEI, and suffix SEI NAL units only. NAL unit types that are reserved in ISO/IEC 23008-2 and in this document may acquire a definition in future, and readers should ignore arrays with reserved or unpermitted values of NAL unit type.

NOTE 2 This 'tolerant' behaviour is designed so that errors are not raised, allowing the possibility of backwards-compatible extensions to these arrays in future specifications.

It is recommended that the arrays be in the order VPS, SPS, PPS, prefix SEI, suffix SEI.

When `general_non_packed_constraint_flag` (bit 3 of the 6-byte `general_constraint_indicator_flags`) is equal to 0 and some of the samples referring to this sample entry represent frame-packed content and any of the default display windows specified by the active SPSs for the samples referring to this sample entry covers more than one constituent frame of the frame-packed content, the techniques described in ISO/IEC 14496-12:2020, subclause 8.15 ('Post-decoder requirements on media') using the scheme type "stvi" shall be used. In this case, the `stereo_scheme` in the Stereo Video Box should be set to 1, to indicate that the frame packing scheme used in HEVC is the same as in AVC.

8.3.2.1.2 Syntax

```

aligned(8) class HEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    bit(4) reserved = '1111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) reserved = '111111'b;
    unsigned int(2) parallelismType;
    bit(6) reserved = '111111'b;
    unsigned int(2) chroma_format_idc;
    bit(5) reserved = '11111'b;
    unsigned int(3) bit_depth_luma_minus8;
    bit(5) reserved = '11111'b;
    unsigned int(3) bit_depth_chroma_minus8;
    unsigned int(16) avgFrameRate;
    unsigned int(2) constantFrameRate;
    unsigned int(3) numTemporalLayers;
    unsigned int(1) temporalIdNested;
    unsigned int(2) lengthSizeMinusOne;
    unsigned int(8) numOfArrays;
    for (j=0; j < numOfArrays; j++) {
        unsigned int(1) array_completeness;
        bit(1) reserved = 0;
        unsigned int(6) NAL_unit_type;
        unsigned int(16) numNalus;
        for (i=0; i< numNalus; i++) {
            unsigned int(16) nalUnitLength;
            bit(8*nalUnitLength) nalUnit;
        }
    }
}

```

8.3.2.1.3 Semantics

general_profile_space, general_tier_flag, general_profile_idc, general_profile_compatibility_flags, general_constraint_indicator_flags, general_level_idc, min_spatial_segmentation_idc, chroma_format_idc, bit_depth_luma_minus8 and bit_depth_chroma_minus8 contain the matching values for the fields general_profile_space, general_tier_flag, general_profile_idc, general_profile_compatibility_flag[i] for i from 0 to 31, inclusive, the 6 bytes starting with the byte containing the general_progressive_source_flag, general_level_idc, min_spatial_segmentation_idc, chroma_format_idc, bit_depth_luma_minus8 and bit_depth_chroma_minus8 as defined in ISO/IEC 23008-2, for the stream to which this configuration record applies.

parallelismType indicates the type of parallelism that is used to meet the restrictions imposed by min_spatial_segmentation_idc when the value of min_spatial_segmentation_idc is greater than 0. Value 1 indicates that the stream to which this configuration record applies supports slice based parallel decoding. Value 2 indicates that the stream to which this configuration record applies supports tile based parallel decoding. Value 3 indicates that the stream to which this configuration record applies supports entropy

coding synchronization based parallel decoding. Value 0 indicates that the stream supports mixed types of parallel decoding or that the parallelism type is unknown. The values above can be inferred by the fields `tiles_enabled_flag` and `entropy_coding_sync_enabled_flag` as defined in ISO/IEC 23008-2. Specifically: if the fields `tiles_enabled_flag` and `entropy_coding_sync_enabled_flag` are both equal to 0 in all PPSs that are activated when the stream to which this configuration record applies is decoded, then `parallelismType` can be set to 1. If the field `tiles_enabled_flag` is equal to 1 in all PPSs that are activated when the stream to which this configuration record applies is decoded, then `parallelismType` can be set to 2. If the field `entropy_coding_sync_enabled_flag` is equal to 1 in all PPSs that are activated when the stream to which this configuration record applies is decoded, then `parallelismType` can be set to 3. If none of the above is true (or if it is unknown which of them is true) then `parallelismType` should be set to 0.

`avgFrameRate` gives the average frame rate in units of frames/(256 seconds), for the stream to which this configuration record applies. Value 0 indicates an unspecified average frame rate.

`constantFrameRate` equal to 1 indicates that the stream to which this configuration record applies is of constant frame rate. Value 2 indicates that the representation of each temporal layer in the stream is of constant frame rate. Value 0 indicates that the stream may or may not be of constant frame rate.

`numTemporalLayers` greater than 1 indicates that the stream to which this configuration record applies is temporally scalable and the contained number of temporal layers (also referred to as temporal sub-layer or sub-layer in ISO/IEC 23008-2) is equal to `numTemporalLayers`. Value 1 indicates that the stream is not temporally scalable. Value 0 indicates that it is unknown whether the stream is temporally scalable.

`temporalIdNested` equal to 1 indicates that all SPSs that are activated when the stream to which this configuration record applies is decoded have `sps_temporal_id_nesting_flag` as defined in ISO/IEC 23008-2 equal to 1 and temporal sub-layer up-switching to any higher temporal layer can be performed at any sample. Value 0 indicates that the conditions above are not or may not be met.

`lengthSizeMinusOne` plus 1 indicates the length in bytes of the `NALUnitLength` field in an HEVC video sample in the stream to which this configuration record applies. For example, a size of one byte is indicated with a value of 0. The value of this field shall be one of 0, 1, or 3 corresponding to a length encoded with 1, 2, or 4 bytes, respectively.

`numArrays` indicates the number of arrays of NAL units of the indicated type(s).

`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.

`NAL_unit_type` indicates the type of the NAL units in the following array (which shall be all of that type); it takes a value as defined in ISO/IEC 23008-2; it is restricted to take one of the values indicating a VPS, SPS, PPS, prefix SEI, or suffix SEI NAL unit.

`numNalus` indicates the number of NAL units of the indicated type included in the configuration record for the stream to which this configuration record applies. The SEI array shall only contain SEI messages of a 'declarative' nature, that is, those that provide information about the stream as a whole. An example of such an SEI could be a user-data SEI.

`nalUnitLength` indicates the length in bytes of the NAL unit.

`nalUnit` contains an SPS, PPS, VPS or declarative SEI NAL unit, as specified in ISO/IEC 23008-2. 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`.

8.4 Derivation from ISO base media file format

8.4.1 HEVC video stream definition

8.4.1.1 Sample entry name and format

8.4.1.1.1 Definition

Sample Entry and Box Types: 'hvc1', 'hev1', 'hvcC'

Container: Sample Table Box ('stbl')

Mandatory: An 'hvc1' or 'hev1' sample entry is mandatory

Quantity: One or more sample entries may be present

An HEVC visual sample entry shall contain an HEVC Configuration Box, as defined below. This includes an HEVCDecoderConfigurationRecord, as defined in 8.3.2.1.

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

When the sample entry name is 'hvc1' or 'hev1', the stream to which this sample entry applies shall be a compliant and HEVC stream as viewed by an HEVC decoder operating under the configuration (including profile, tier, and level) given in the HEVCConfigurationBox.

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.

8.4.1.1.2 Syntax

```
class HEVCConfigurationBox extends Box('hvcC') {
    HEVCDecoderConfigurationRecord() HEVCConfig;
}

class HEVCSampleEntry() extends VisualSampleEntry ('hvc1' or 'hev1'){
    HEVCConfigurationBox config;
    MPEG4ExtensionDescriptorsBox (); // optional
}
```

8.4.1.1.3 Semantics

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\013HEVC Coding" being recommended (\013 is 11, the length of the string in bytes).

HEVCDecoderConfigurationRecord is defined in 8.3.2.

8.4.2 Parameter sets in sample entry

This subclause applies to a particular type of parameter sets (VPSs, SPSs, or PPSs) when the particular type of parameter sets is included in the sample entry.

Each HEVC sample entry, which contains the HEVC video stream decoder specific information, includes a group of the particular type of parameter sets. This group of parameter sets functions much like a codebook. Each parameter set has an identifier, and each slice references the parameter set it was coded against using the parameter set's identifier.

In the file format each configuration of parameter sets is represented separately. When the value of the applicable `array_completeness` is 1, a parameter set cannot be updated without causing a different sample entry to be used.

Systems wishing to send parameter set updates will need to compare the two configurations to find the differences in order to send the appropriate parameter set updates.

It is recommended that when several parameter sets are used and parameter set updating is desired, the parameter sets are included in the samples of the stream.

NOTE Decoders conforming to this document need to support both parameter sets stored in the samples as well as parameter sets stored in the sample entries, unless restricted by another specification using this one.

8.4.3 Sync sample

A sync sample in 'hvc1' and 'hev1' tracks shall contain VCL NAL units indicating that the coded picture with `nuh_layer_id` equal to 0 in the sample is an Instantaneous Decoding Refresh (IDR) picture, a Clean Random Access (CRA) picture, or a Broken Link Access (BLA) picture. When the coded picture with `nuh_layer_id` equal to 0 in a sync sample is a BLA or CRA picture, there shall be no RASL pictures associated with that BLA or CRA picture.

Former versions of this specification allowed the presence of RASL pictures associated with sync samples, although this is not in conformance with ISO/IEC 14496-12. When RASL pictures are associated with a sync sample, tools that inspect only file-format structures might reach incorrect conclusions about the dependencies in the bitstream. RASL pictures can be signalled in the file format as leading pictures in the `SampleDependencyTypeBox`, and BLA and CRA pictures with RASL pictures can be signalled as a SAP of type 3, using the SAP sample group or using the RAP sample group. For older files conforming to previous versions of the specification, when performing random access into this track, or after a discontinuous edit, the flag *HandleCraAsBlaFlag* can be set for the HEVC decoder. For environments where HEVC file processors need information that sync samples are not associated with RASL pictures, the brand 'nras', as defined in Clause D.5, may be used to indicate compliance to this document and later versions.

NOTE In the context of L-HEVC file format specified in Clause 9, the signalling of sync samples that concerns a particular base layer picture can be different from the signalling of sync samples here for the same picture.

Table 9 indicates the mapping between HEVC VCL NAL unit types, ISOBMFF sync sample status and SAP types as documented in ISOBMFF.

Table 9 — Mapping of sync sample status and SAP types to NAL unit type

NAL Unit Type	ISOBMFF sync sample status	DASH SAP type
IDR_N_LP	true	1
IDR_W_RADL	true	2 (if the IRAP has associated RADL pictures) 1 (if the IRAP has no associated RADL pictures)
BLA_N_LP	true	1
BLA_W_RADL	true	2 (if the IRAP has associated RADL pictures) 1 (if the IRAP has no associated RADL pictures)
BLA_W_LP	false (*1)	3 (if the IRAP has associated RASL pictures)
	true	2 (if the IRAP has no associated RASL pictures but has associated RADL pictures)
	true	1 (if the IRAP has no associated leading pictures)
CRA	false (*1)	3 (if the IRAP has associated RASL pictures)
	true	2 (if the IRAP has no associated RASL pictures but has associated RADL pictures)
	true	1 (if the IRAP has no associated leading pictures)
(*1) see above discussion on sync samples and leading HEVC pictures		

When the sample entry name is 'hev1', the following applies:

- If the sample is a sync sample, all parameter sets 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 parameter sets 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.

For signalling of various types of random access points, the following guidelines are recommended:

- The sync sample table (and the equivalent flag in movie fragments) shall be used in an HEVC track unless all samples are sync samples. Note that track fragment random access box refers to the presence of signalled sync samples in a movie fragment.
- The 'roll' sample group is recommended to be used only for gradual decoding refresh (GDR) based random access points, i.e. those that contain non-intra coded slices.

- The use of the 'rap' or 'sync' sample group is optional, depending on the need of either the information on leading samples associated with the random access points or the picture types (e.g. IDR, CRA, or BLA) of the random access points.
- The use of the Alternative Startup Sequences (ISO/IEC 14496-12:2020, subclause 10.3) sample group is recommended to be used only with random access points consisting of CRA and BLA pictures.

In the context of this clause, the leading samples, defined as part of the definition of the 'rap' sample group in ISO/IEC 14496-12, contain Random Access Skipped Leading (RASL) access units as defined in ISO/IEC 23008-2.

8.4.4 Sync sample sample grouping

8.4.4.1 Overview

Sync samples in HEVC may be of various types. These sample groups may be used to identify the sync samples of a specific type. If a sample group is given for a specific type of sync sample, then all samples (if any) containing that type of sync sample are marked by the group. If the group is absent (there is no sample to group mapping for that type), it is unknown which samples contain a sync sample of that type.

8.4.4.2 Sync sample sample group entry

8.4.4.2.1 Definition

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

A sync sample sample group entry identifies samples containing a sync sample of a specific type. The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type 'sync'.

8.4.4.2.2 Syntax

```
class SyncSampleEntry() extends VisualSampleGroupEntry ('sync')
{
    bit(2) reserved = 0;
    unsigned int(6) NAL_unit_type;
}
```

8.4.4.2.3 Semantics

`NAL_unit_type` shall be a type that identifies a valid sync sample (e.g. IDR).

8.4.5 Temporal scalability sample grouping

8.4.5.1 Overview

An HEVC video track may contain zero or one instance of a `SampleToGroupBox` with a `grouping_type` 'tscl'. This `SampleToGroupBox` instance represents the assignment of samples

in the track to temporal layers (referred to as temporal sub-layers or sub-layers in ISO/IEC 23008-2). An accompanying instance of the SampleGroupDescriptionBox with the same grouping type shall be present, and contain TemporalLayerEntry sample group entries describing the temporal layers.

There may also be a set of Temporal Level sample group definitions (ISO/IEC 14496-12:2020, subclause 10.5). A sample mapped to a sample group description entry with index A of a Temporal Level sample grouping shall also be considered mapped to the sample group description entry in the temporal scalability sample group (defined here) having temporalLayerId equal to A (i.e. a sample to group mapping for the temporal scalability sample group may also be present but is not required).

8.4.5.2 Temporal layer sample group entry

8.4.5.2.1 Definition

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

A temporal layer sample group entry defines the temporal layer information for all samples in a temporal layer. Temporal layers are numbered with non-negative integers, and each temporal layer is associated with a particular value of TemporalId as defined in ISO/IEC 23008-2. A temporal layer associated with a TemporalId value greater than 0 depends on all temporal layers associated with lower TemporalId values. A temporal layer representation, also referred to as the representation of a temporal layer, associated with a particular TemporalId value consists of all temporal layers associated with TemporalId values less than or equal to the given TemporalId value.

The grouping_type_parameter is not defined for the SampleToGroupBox with grouping type 'tscl'.

8.4.5.2.2 Syntax

```
class TemporalLayerEntry() extends VisualSampleGroupEntry ('tscl')
{
    unsigned int(8)    temporalLayerId;
    unsigned int(2)    tlprofile_space;
    unsigned int(1)    tltier_flag;
    unsigned int(5)    tlprofile_idc;
    unsigned int(32)   tlprofile_compatibility_flags;
    unsigned int(48)   tlconstraint_indicator_flags;
    unsigned int(8)    tllevel_idc;
    unsigned int(16)   tlMaxBitRate;
    unsigned int(16)   tlAvgBitRate;
    unsigned int(8)    tlConstantFrameRate;
    unsigned int(16)   tlAvgFrameRate;
}
```

8.4.5.2.3 Semantics

temporalLayerId gives the ID of this temporal layer. For all samples that are members of this sample group, the VCL NAL units shall have TemporalId, as defined in ISO/IEC 23008-2, equal to temporalLayerId.

`tlprofile_space`, `tltier_flag`, `tlprofile_idc`,
`tlprofile_compatibility_flags`, `tlconstraint_indicator_flags`, and
`tllevel_idc` contain the values of `general_profile_space`, `general_tier_flag`,
`general_profile_idc`, `general_profile_compatibility_flag[i]` for `i` from 0 to 31, inclusive, the 6
bytes starting with the byte containing the `general_progressive_source_flag`, and
`general_level_idc`, respectively, for the representation of the temporal layer identified by
`temporalLayerId`.
`tlMaxBitrate` gives the maximum rate in 1000 bits per second over any window of one second,
for the representation of the temporal layer identified by `temporalLayerId`.
`tlAvgBitRate` gives the average bit rate in units of 1000 bits per second, for the representation of
the temporal layer identified by `temporalLayerId`.
`tlConstantFrameRate` equal to 1 indicates that the representation of the temporal layer
identified by `temporalLayerId` is of constant frame rate. Value zero indicates that the
representation of the temporal layer identified by `temporalLayerId` may or may not be of
constant frame rate.
`tlAvgFrameRate` gives the average frame rate in units of frames/(256 seconds), for the
representation of the temporal layer identified by `temporalLayerId`. Value 0 indicates an
unspecified average frame rate.

8.4.6 Temporal sub-layer access sample grouping

8.4.6.1 Overview

An HEVC video track may contain zero or one instance of a `SampleToGroupBox` with a `grouping_type` 'tsas'. This `SampleToGroupBox` instance represents the marking of samples as temporal sub-layer access points. An accompanying instance of the `SampleGroupDescriptionBox` with the same grouping type shall be present.

8.4.6.2 Temporal sub-layer access sample group entry

8.4.6.2.1 Overview

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

This sample group is used to mark temporal sub-layer access (TSA) samples. The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type 'tsas'.

8.4.6.2.2 Syntax

```

class TemporalSubLayerEntry() extends VisualSampleGroupEntry ('tsas')
{
}

```

8.4.7 Step-wise temporal layer access sample grouping

8.4.7.1 Overview

An HEVC video track may contain zero or one instance of a `SampleToGroupBox` with a `grouping_type` 'stsa'. This `SampleToGroupBox` instance represents the marking of samples as step-wise temporal sub-layer access points. An accompanying instance of the `SampleGroupDescriptionBox` with the same grouping type shall be present.

When `temporalIdNested` in the applicable sample entry is equal to 1, the quantity of step-wise temporal sub-layer access sample group entry shall be zero.

8.4.7.2 Step-wise temporal layer sample group entry

8.4.7.2.1 Definition

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

This sample group is used to mark step-wise temporal sub-layer access (STSA) samples. The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type 'stsa'.

8.4.7.2.2 Syntax

```
class StepwiseTemporalLayerEntry() extends VisualSampleGroupEntry ('stsa')
{
}
```

8.4.8 Definition of a sub-sample for HEVC

For the use of the `SubSampleInformationBox` (ISO/IEC 14496-12:2020, subclause 8.7.7) in an HEVC stream, a sub-sample is defined on the basis of the value of the `flags` field of the sub-sample information box as specified below. The presence of this box is optional; however, if present in a track containing HEVC data, the 'codecspecific_parameters' field in the box shall have the semantics defined here.

`flags` specifies the type of sub-sample information given in this box as follows:

- 0: NAL-unit-based sub-samples. A sub-sample contains one or more contiguous NAL units.
- 1: Decoding-unit-based sub-samples. A sub-sample contains exactly one decoding unit.
- 2: Tile-based sub-samples. A sub-sample either contains one tile and the associated non-VCL NAL units, if any, of the VCL NAL unit(s) containing the tile, or contains one or more non-VCL NAL units.

NOTE 1 Tile-based sub-samples can be used to address individual tiles within a slice, whereas tile regions (as defined in Clause 10) can only be used for a set of tiles that are enclosed in one or more complete slice segments.

- 3: CTU-row-based sub-samples. A sub-sample either contains one CTU row within a slice and the associated non-VCL NAL units, if any, of the VCL NAL unit(s) containing the CTU row or contains one or more non-VCL NAL units. This type of sub-sample information shall not be used when `entropy_coding_sync_enabled_flag` is equal to 0.
- 4: Slice-based sub-samples. A sub-sample either contains one slice (where each slice may contain one or more slice segments, each of which is a NAL unit) and the associated non-VCL NAL units, if any, or contains one or more non-VCL NAL units.
- 5: Picture-based sub-samples. A sub-sample contains one coded picture and the associated non-VCL NAL units.

Other values of `flags` are reserved.

The `subsample_priority` field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12.

The `discardable` field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded (e.g. the sub-sample consists of an SEI NAL unit).

When the first byte of a NAL unit is included in a sub-sample, the preceding length field shall also be included in the same sub-sample.

The `codec_specific_parameters` field of the `SubSampleInformationBox` is defined for HEVC as follows:

```

if (flags == 0) {
    unsigned int(1) SubLayerRefNalUnitFlag;
    unsigned int(1) RapNalUnitFlag;
    unsigned int(1) VclNalUnitFlag;
    bit(29) reserved = 0;
} else if (flags == 1) {
    bit(32) reserved = 0;
} else if (flags == 2) {
    unsigned int(2) vcl_idc;
    bit(2) reserved = 0;
    unsigned int(4) log2_min_luma_ctb;
    unsigned int(12) ctb_x;
    unsigned int(12) ctb_y;
} else if (flags == 3 || flags == 4) {
    unsigned int(2) vcl_idc;
    bit(30) reserved = 0;
} else if (flags == 5) {
    unsigned int(1) DiscardableFlag;
    unsigned int(6) VclNalUnitType;
    unsigned int(6) LayerId;
    unsigned int(3) TempId;
    unsigned int(1) NoInterLayerPredFlag;
    unsigned int(1) SubLayerRefNalUnitFlag;
    bit(14) reserved = 0;
}

```

SubLayerRefNalUnitFlag equal to 0 indicates that all NAL units in the sub-sample are VCL NAL units of a sub-layer non-reference picture as specified in ISO/IEC 23008-2. Value 1 indicates that all NAL units in the sub-sample are VCL NAL units of a sub-layer reference picture as specified in ISO/IEC 23008-2.

RapNalUnitFlag equal to 0 indicates that none of the NAL units in the sub-sample has **nal_unit_type** equal to **IDR_W_RADL**, **IDR_N_LP**, **CRA_NUT**, **BLA_W_LP**, **BLA_W_RADL**, **BLA_N_LP**, **RSV_IRAP_VCL22**, or **RSV_IRAP_VCL23** as specified in ISO/IEC 23008-2. Value 1 indicates that all NAL units in the sub-sample have **nal_unit_type** equal to **IDR_W_RADL**, **IDR_N_LP**, **CRA_NUT**, **BLA_W_LP**, **BLA_W_RADL**, **BLA_N_LP**, **RSV_IRAP_VCL22**, or **RSV_IRAP_VCL23** as specified in ISO/IEC 23008-2.

VclNalUnitFlag equal to 0 indicates that all NAL units in the sub-sample are non-VCL NAL units. Value 1 indicates that all NAL units in the sub-sample are VCL NAL units.

vcl_idc indicates whether the sub-sample contains Video Coding Layer (VCL) data, non-VCL data, or both, as follows:

- 0: the sub-sample contains VCL data and does not contain non-VCL data
- 1: the sub-sample contains no VCL data and contains non-VCL data
- 2: the sub-sample may contain both VCL and non-VCL data, which shall be associated with each other. For example, a sub-sample may contain a decoding unit information SEI message followed by the set of NAL units associated with the SEI message.
- 3: reserved

log2_min_luma_ctb indicates the unit of **ctb_x** and **ctb_y**, specified as follows:

- 0: 8 luma samples
- 1: 16 luma samples
- 2: 32 luma samples
- 3: 64 luma samples

ctb_x specifies the 0-based coordinate of the right-most luma samples of the tile associated with the sub-sample when **flags** is equal to 2 and **vcl_idc** is equal to 1 or 2, in units derived from **log2_min_luma_ctb** as specified above.

ctb_y specifies the 0-based coordinate the bottom-most luma samples of the tile associated with the sub-sample when **flags** is equal to 2 and **vcl_idc** is equal to 1 or 2, in units derived from **log2_min_luma_ctb** as specified above.

DiscardableFlag indicates the **discardable_flag** value of the VCL NAL units in the sub-sample. All the VCL NAL units in the sub-sample shall have the same **discardable_flag** value.

NOTE 2 This is not the same definition as the **discardable** field in the **SubSampleInformationBox**.

VclNalUnitType indicates the **nuh_unit_type** value of the VCL NAL units in the sub-sample. All the VCL NAL units in the sub-sample shall have the same **nuh_unit_type** value.

LayerId indicates the **nuh_layer_id** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same **nuh_layer_id** value.

TempId indicates the **TemporalId** value of the NAL units in the sub-sample. All the NAL units in the sub-sample shall have the same **TemporalId** value.

NoInterLayerPredFlag indicates the value of the **inter_layer_pred_enabled_flag** of the VCL NAL units in the sub-sample. All the VCL NAL units in the sub-sample shall have the same value of **inter_layer_pred_enabled_flag**.

8.4.9 Handling non-output samples

HEVC allows for file format samples that are used only for reference and not output (e.g. a non-displayed reference picture in video). When any such non-output sample is present in a track, the file shall be constrained as follows:

- a) A non-output sample shall be given a composition time that is outside the time-range of the samples that are output.
- b) An edit list shall be used to exclude the composition times of the non-output samples.
- c) When the track includes a CompositionOffsetBox ('ctts'),
 - 1) version 1 of the CompositionOffsetBox shall be used,
 - 2) the value of `sample_offset` shall be set equal to -2^{31} for each non-output sample,
 - 3) the CompositionToDecodeBox ('cslg') should be contained in the SampleTableBox ('stbl') of the track, and
 - 4) when the CompositionToDecodeBox is present for the track, the value of `leastDecodeToDisplayDelta` field in the box shall be equal to the smallest composition offset in the CompositionOffsetBox excluding the `sample_offset` values for non-output samples.

NOTE Thus, `leastDecodeToDisplayDelta` is greater than -2^{31} .

9 Layered HEVC elementary stream and sample definitions

9.1 Overview

This clause specifies the storage format of coded video data of layered HEVC (L-HEVC), which includes all the layered HEVC extensions that use the same layered design specified in Annex F of ISO/IEC 23008-2:2020. These layered HEVC extensions include SHVC, MV-HEVC, and 3D-HEVC. This clause also specifies the storage of temporal sub-layers of a layer in more than one track. This clause extends the definitions of the storage format of HEVC in Clause 8.

The file format as defined in this clause and Annex A uses the existing capabilities of the ISO base media file format and the plain HEVC file format (i.e. the file format specified in Clause 8). Clause D.4 specifies brands for layered HEVC. The following structures or extensions, among others, to support L-HEVC-specific features are used:

- a) Aggregators enable efficient scalable grouping of NAL units by changing irregular patterns of NAL units into regular patterns of aggregated data units.

NOTE 1 When compared to Aggregators for AVC/SVC/MVC, the syntax and semantics of the NAL unit header syntax element in the L-HEVC aggregator have been modified and the scope of the aggregator has been constrained.

- b) Extractors enable efficient extraction of NAL units from other tracks than the one containing the media data.

NOTE 2 When compared to Extractors for AVC/SVC/MVC, the syntax and semantics of the L-HEVC Extractors have been modified.

- c) `LHEVCConfigurationBox` in a track having a sample entry specified in Clause 8 enables the storage of an L-HEVC bitstream in a manner that the HEVC compatible base layer can be used by any plain HEVC file format compliant reader. L-HEVC data can be stored with the base layer or in one or more dedicated tracks.
- d) An L-HEVC bitstream using an externally provided base layer compatible with an external codec (e.g. AVC) stored in a way that the base layer can be used by any file format reader compliant to that external codec (e.g. plain AVC file format compliant reader).

9.2 Overview of L-HEVC storage

The support for L-HEVC includes a number of tools, and there are various 'models' of how they might be used. In particular, an L-HEVC stream can be placed in tracks in a number of ways, among which are the following:

- a) all the layers in one track;
- b) each layer or sub-layer in its own track;
- c) the expected operating points each in a track (e.g. the HEVC base, a stereo pair, a multiview scene).

The L-HEVC file format allows storage of one or more layers into a track. Storage of multiple layers per track can be used. For example, when a content provider wants to provide a multi-layer bitstream that is not intended for subsetting, or when the bitstream has been created for a few pre-defined sets of output layers where each layer corresponds to a view (such as 1, 2, 5, or 9 views), tracks can be created accordingly.

When an L-HEVC bitstream is represented by multiple tracks and a player uses an operating point for which the layers are stored in multiple tracks, the player shall reconstruct L-HEVC access units before passing them to the L-HEVC decoder. An L-HEVC operating point may be explicitly represented by a track, i.e., each sample in the track contains an access unit, where some or all NAL units of the access unit may be contained in or referred to by extractors and aggregators. If the number of operating points is large, it may be space-consuming and impractical to create a track for each operating point. In such a case, L-HEVC access units are reconstructed as specified in 9.5.2.2.

The storage of L-HEVC bit streams is supported by structures such as the sample entry, Operating Points Information ('`oinf`') sample group, and Layer Information ('`linf`') sample group. The structures within a sample entry provide information for the decoding or use of the samples, in this case coded video information, that are associated with that sample entry. The Operating Points Information sample group records information about operating points such as the layers and sub-layers that constitute the operating point, dependencies (if any) between them, the profile, level, and tier parameter of the operating point, and other such operating point relevant information. The layer information sample group lists all the layers and sub-layers carried in the samples of the track. The information in these sample groups, combined with using track references to find tracks, is sufficient for a reader to choose an operating point in accordance with its capabilities, identify the tracks that contain the relevant layers and sub-layers needed to decode the chosen operating point, and efficiently extract them.

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.

The L-HEVC file format supports a hybrid codec configuration where the base layer is coded using an external video codec (e.g. AVC). The base layer, if coded using an external codec specification (e.g. ISO/IEC 14496-10), shall be stored according to the external codec file format (e.g. the AVC file format as specified in Clause 5).

9.3 L-HEVC elementary stream structure

L-HEVC streams are stored in accordance with subclause 8.2, with the exception that an L-HEVC stream may be represented by more than one video track in a file, and with the following definition of an L-HEVC video elementary stream:

- **An L-HEVC video elementary stream** contains all video coding related NAL units (i.e. those NAL units containing video data or signalling video structure, possibly after resolution of extractors and or aggregators) and may contain non-video coding related NAL units such as SEI messages and access unit delimiter NAL units. Aggregators and Extractors, when present, shall not be directly output by file parsers.

9.4 Sample and configuration definition

9.4.1 Overview

A sample in an 'hvc1' or 'hev1' track with the L-HEVC configuration is an L-HEVC sample, while a sample in an 'hvc1' or 'hev1' track without the L-HEVC configuration is not an L-HEVC sample. Throughout Clause 9, unless mentioned otherwise, an access unit refers to an L-HEVC access unit and is as defined in ISO/IEC 23008-2:2020, Annex F.

9.4.2 Canonical order and restrictions

The following restrictions apply to L-HEVC data in addition to the requirements in subclause 8.3.1.

- If all VCL NAL units for a single picture are stored in the same track, they shall be contained within the sample whose decoding time and composition time are those of the picture. Otherwise (i.e., when pieces of a coded picture are stored in samples of different tracks), each of these samples shall have the decoding time and composition time that are those of the coded picture.
- **VCL NAL units:** All VCL NAL units within one access unit and belonging to the layers represented by a track shall be contained in the sample and the sample shall not contain NAL units of any other access unit. After resolution of extractors and aggregators, when present, an L-HEVC sample shall

contain an integer number of picture units. The picture units in an L-HEVC sample shall be stored in their decoding order.

- **Aggregators and extractors:** Each aggregator is allowed to aggregate NAL units belonging to one picture unit only. The order of all NAL units included in an aggregator or in a resolved extractor is exactly the decoding order as if these NAL units were present in a sample not containing aggregators or extractors. After processing the aggregator or the extractor, all NAL units shall be in valid decoding order as specified in ISO/IEC 23008-2.
- **Carriage of external-codec coded base layer track:** An external-codec (e.g. AVC) coded base layer shall always be carried in a track of its own, and shall consist of samples as specified in the external codec file format (e.g. the AVC file format as specified in Clause 5).

9.4.3 Decoder configuration record

When the decoder configuration record defined in subclause 8.3.2.1 is used for a stream that can be interpreted as either an L-HEVC or HEVC stream, the HEVC decoder configuration record shall apply to the HEVC compatible base layer, and should contain only parameter sets needed for decoding the HEVC base layer.

The syntax of `LHEVCDecoderConfigurationRecord` is as follows:

```
aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    bit(4) reserved = '1111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) reserved = '111111'b;
    unsigned int(2) parallelismType;
    bit(2) reserved = '11'b;
    unsigned int(3) numTemporalLayers;
    unsigned int(1) temporalIdNested;
    unsigned int(2) lengthSizeMinusOne;
    unsigned int(8) numOfArrays;
    for (j=0; j < numOfArrays; j++) {
        unsigned int(1) array_completeness;
        bit(1) reserved = 0;
        unsigned int(6) nal_unit_type;
        unsigned int(16) numNalus;
        for (i=0; i < numNalus; i++) {
            unsigned int(16) nalUnitLength;
            bit(8*nalUnitLength) nalUnit;
        }
    }
}
```

The semantics of the fields that are common to `LHEVCDecoderConfigurationRecord` and `HEVCDecoderConfigurationRecord` remain unchanged. If the track containing `LHEVCDecoderConfigurationRecord` has a 'scal' track reference, the semantics of the fields apply to the output bitstream resulting from the explicit reconstruction process. Otherwise, if there is at least one operating point for which the highest output layer is natively present in the track, the semantics of the fields apply to all (implicitly reconstructed) output bitstreams of the operating points for which the

highest sub-layer of the highest output layer is natively present in the track. Otherwise, the semantics of the fields apply to all sub-layers natively present in the track.

For each auxiliary picture layer included in the track, it is recommended to include, within `nalUnit`, an SEI NAL unit containing a declarative SEI message, such as the depth representation information SEI message for depth auxiliary picture layers, specifying characteristics of the auxiliary picture layer.

9.5 Derivation from the ISO base media file format and the HEVC file format (Clause 8)

9.5.1 L-HEVC track structure

An L-HEVC stream is represented by one or more video tracks in a file. Each track represents one or more layers or sub-layers of the coded bitstream.

There is a minimal set of one or more tracks that, when taken together, contain the complete set of encoded information. This group of tracks that forms the complete encoded information is called the “complete subset”. The complete encoded information can be retained when the tracks included in the “complete subset” are retained; all other tracks shall represent subsets or copies (through extractors) of the complete subset.

All tracks of the complete subset shall be indicated with a track group of type '`cstg`' (complete subset track grouping).

For each valid pair of `nuh_layer_id` and `TemporalId` among all the coded pictures of an L-HEVC bitstream, the complete subset shall include one and only one track in which the '`linf`' sample group indicates that the sub-layer identified by `nuh_layer_id` and `TemporalId` is natively present.

NOTE Consequently, a file reader can extract a sub-bitstream of any operating point without the need to parse Aggregators and Extractors.

For an L-HEVC bitstream with an external base layer (i.e., when an active VPS of the bitstream has `vps_base_layer_internal_flag` equal to 0 and `vps_base_layer_available_flag` equal to 1), the base layer is always assigned one track of its own. A track containing an external base layer for an L-HEVC bitstream shall not be a part of the '`cstg`' track group of the tracks of that L-HEVC bitstream.

If an L-HEVC bitstream has an external base layer, the track containing the external base layer is nominated as the base track. Otherwise, the track with the lowest temporal sub-layer (the VCL NAL units of which have `TemporalId` equal to 0) of the base layer natively present is nominated as the base track.

All the other tracks that are part of the same stream shall be linked to the base track by means of a track reference of type '`sbas`'. All the tracks sharing the same base track shall share the same timescale as the scalable base track.

Tracks that contain extractors shall be linked to the tracks from which they extract data using the '`scal`' track reference type.

If a track containing parts of an L-HEVC bitstream is removed from a file, tracks that contain '`scal`' and '`sbas`' track references to the removed track should also be removed.

When the base layer is coded using AVC, the base layer track shall be constructed according to Clause 5 without using separate parameter set tracks.

9.5.2 Data sharing and reconstruction of an L-HEVC bitstream

9.5.2.1 General

NOTE 1 When only the entire base layer carried in the base track is played or when only certain sub-layers, carried in the base track, of the base layer are played, and the base track natively contains all the required sub-layers of the base layer as identified by the `HEVCDecoderConfigurationRecord` of the base track, the reconstruction processes of this clause are unnecessary.

Different tracks may logically share data through the use of extractors (defined in Clause A.7).

In order to reconstruct an access unit from samples of multiple tracks carrying an L-HEVC bitstream, the target output layers and the operating point they belong to may need to be determined first.

NOTE 2 Players can conclude the layers that are required for decoding the determined target output layers from operating point list included in the Operating Points Information sample group. Tracks that carry the relevant layers for an operating point can be obtained by following the 'oref' track references and information in the layer information sample group.

If several tracks contain data for the access unit, the alignment of respective samples in tracks is performed based on the sample decoding times, i.e. using the time-to-sample table only without considering edit lists.

When an L-HEVC stream is represented by multiple tracks, the decoding times of the samples shall be such that if the tracks were combined into a single stream ordered by increasing decoding time, the access unit order would be correct as specified in ISO/IEC 23008-2.

If the track containing the highest sub-layer of the highest layer of a desired operating point has 'scal' track references, all access units of the track for that operating point are reconstructed by solving the extractors and aggregators in the samples of that track, and the result is the bitstream of the operating point; this reconstruction process is referred to as explicit reconstruction.

Otherwise, a sequence of access units is reconstructed from the respective samples in the required tracks according to the implicit reconstruction process as described in 9.5.2.2.

9.5.2.2 Implicit reconstruction

In the implicit reconstruction, the required tracks are selected based on the layers they carry and their reference layers as indicated by the Operating Points Information and Layer Information sample groups. When reconstructing a bitstream containing a sub-layer for which the VCL NAL units have `TemporalId` greater than 0, all lower sub-layers (i.e., those for which the VCL NAL units have smaller `TemporalId`) within the same layer are also included in the resulting bitstream and the required tracks are selected accordingly.

Each access unit is reconstructed as described below, the reconstructed access units are placed into the L-HEVC bitstream in increasing order of decoding time, and the duplicates of end of bitstream (EOB) and end of sequence (EOS) NAL units are removed from the L-HEVC bitstream, as described further below.

Aggregators, when present, are resolved, while extractors, when present (natively present or resolved from aggregators), are discarded without being resolved, and the resultant NAL units are arranged in an order conforming to ISO/IEC 23008-2.

When reconstructing an access unit, picture units (as specified in ISO/IEC 23008-2) from samples having the same decoding time are placed into the access unit in increasing order of the `nuh_layer_id` value. The reconstructed access units are placed into the output bitstream in increasing order of the decoding time.

For a particular access unit for which the NAL units are stored in multiple tracks, there may be more than one of the tracks containing an EOB NAL unit in the respective samples. In this case, only one of the EOB NAL units shall be kept in the final reconstructed access unit, placed at the end of the access unit; other EOB NAL units are discarded.

For access units that are within the same coded video sequence of an HEVC or L-HEVC bitstream and that belong to different sub-layers stored in multiple tracks, there may be more than one of the tracks containing an EOS NAL unit in the respective samples. In this case, only one of the EOS NAL units shall be kept in the last of these access units (the one with the greatest decoding time) in the final reconstructed bitstream, placed after all NAL units, except the EOB NAL unit (when present), of the last of these access units, and other EOS NAL units are discarded. Similarly, there may be more than one of such tracks containing an EOB NAL unit in the respective samples. In this case, only one of the EOB NAL units shall be kept in the final reconstructed bitstream, placed at the end of the last of these access units, and other EOB NAL units are discarded.

Since a particular layer or sub-layer may be represented by more than one track, when figuring out the required tracks for an operating point, a selection may need to be made among the set of tracks that all carry the particular layer or sub-layer.

The final required tracks, after selection among the tracks carrying a same layer or sub-layer, may still collectively carry some layers or sub-layers that do not belong to the target operating point. The reconstructed bitstream for the target operating point should not contain the layers or sub-layers that are carried in the final required tracks but do not belong to the target operating point.

NOTE Some L-HEVC decoder implementations take as input a bitstream as well as the target output layer set index and the highest `TemporalId` value of the target operating point, which correspond to the `TargetOlsIdx` and `HighestTid` variables in ISO/IEC 23008-2:2020, Clause F.8, respectively. In this case the bitstream reconstructed by the file parser can contain layers or sub-layers that do not belong to the target operating point, because these L-HEVC decoders are capable of removing the layers not included in the target output layer set and the sub-layers beyond the highest `TemporalId` value. Some other L-HEVC decoder implementations input a bitstream that is required not to contain any other layers and sub-layers than those included in the target operating point. In this case the file parser needs to ensure that the reconstructed bitstream does not contain any other layers and sub-layers than those included in the target operating point.

9.5.3 L-HEVC video stream definition

9.5.3.1 Sample entry name and format

9.5.3.1.1 Definition

Sample Entry and Box Types: 'hvc2', 'hev2', 'hvc3', 'hev3', 'lhv1', 'lhe1', 'lhvc'
 Container: Sample Description Box ('stsd')
 Mandatory: An 'hvc1', 'hev1', 'hvc2', 'hev2', 'hvc3', 'hev3', 'lhv1', or 'lhe1' sample entry is mandatory
 Quantity: One or more sample entries may be present

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.

When the sample entry name is 'hev2' or 'hev3' and the sample entry contains the HEVC configuration only, the same constraints as specified in subclause 8.4.3 for the sample entry name 'hev1' apply.

When the sample entry name is 'lhe1', or when the sample entry name is 'hev1', 'hev2' or 'hev3' and the sample entry contains both HEVC and L-HEVC configurations, the following applies:

NOTE 1 The constraints below impose restrictions on placement of out-of-band parameter sets (in sample entries) and in-band parameter sets (in samples), in order to enable convenient random access from access units containing IRAP pictures at least in some layers. With these constraints, a file reader that initializes with the sample entries and rolls forward from an access unit wherein all pictures are IRAP pictures will have all the parameter sets it needs.

- a) For any particular sample in a particular track, the temporally collocated sample in another track is defined as the one with the same decoding time as that of this particular sample.
- b) For an IRAP picture of a given sample, track and layer, each parameter set needed for decoding the IRAP picture shall be included in one of the following:
 - 1) the sample entry that applies to the given sample in the given track
 - 2) when implicit reconstruction is used, the sample entry of the initial sample of a track carrying a reference layer of the given layer, where the initial sample is either the given sample's temporally collocated sample, when the temporally collocated sample contains an IRAP picture of the reference layer, or the previous sample that contains an IRAP picture of the reference layer
 - 3) the given sample itself, possibly by using extractors
 - 4) when present, any temporally collocated sample of the tracks carrying reference layers of the given layer, possibly by using extractors
- c) For a non-IRAP picture of a given sample, track and layer, each parameter set needed for decoding that picture shall be included in one of the following:

- 1) the sample entry that applies to the given sample in the given track
- 2) when implicit reconstruction is used, the sample entry of the initial sample of a track carrying a reference layer of the given layer, where the initial sample is either the given sample's temporally collocated sample, when the temporally collocated sample contains an IRAP picture of the reference layer, or the previous sample that contains an IRAP picture of the reference layer
- 3) any of the samples in the given track since the previous sample containing an IRAP picture in the given layer to the given sample itself, inclusive, possibly by using extractors
- 4) when present, any of the samples in a track carrying a reference layer of the given layer since the temporally collocated sample of the previous sample containing an IRAP picture in the given layer to the temporally collocated sample of the given sample, inclusive, possibly by using extractors

The parameter sets needed for decoding a picture in a sample of a track using extractors shall only be located in media samples (of that track or of dependent tracks through extractors), or in the sample description entry associated with that sample, and parameter sets carried in sample entries of other tracks are ignored in this case. This implies that when extractors are used, some parameter sets may be duplicated in the sample description of tracks with extractors and in the other tracks depended on this track. When implicit reconstruction is used, the sample entries of a track shall only contain the parameter sets used in that track and shall not contain the parameter sets used in dependent or depended on tracks. These parameter sets may be stored in the sample or in the sample description associated to the sample, according to the sample entry type used.

When present, parameter sets in the sample entry of an HEVC/L-HEVC track and, in case implicit reconstruction is used, in the sample entries of the tracks depended on, shall be output in the reconstructed bitstream in increasing layer order, and before any parameter sets present in the samples.

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.

NOTE 2 It can happen, in files mixing both implicit reconstructions and extractors, that gathering parameter sets from sample entries result in duplicates; additionally, presence of such duplicates are conformant with ISO/IEC 23008-2, a file reader can discard them.

NOTE 3 When an extractor track is used in implicit reconstruction, file readers do not need to look at the sample entries of tracks the extractor track depends on for their parameter sets, since these parameter sets are duplicated in the sample entry of the extractor track.

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'.

NOTE 4 The above constraints mandate convenient random access (see NOTE 1 in this subclause 9.5.3.1.1) to be enabled and indicated for either all tracks carrying an HEVC or L-HEVC bitstream or none of the tracks.

When the samples of a track contain a temporal subset of an HEVC base layer, the value of numTemporalLayers field in the HEVCConfigurationBox shall match the number of temporal sub-layers present in the track, natively or through extraction. Here, the entry shall contain an HEVCConfigurationBox, possibly followed by an LHEVCConfigurationBox as defined below. The HEVCConfigurationBox documents the Profile, Tier, Level, and possibly also parameter sets of the HEVC compatible base layer as defined by the HEVCDecoderConfigurationRecord. The LHEVCConfigurationBox possibly documents parameter sets of the L-HEVC compatible enhancement layers as defined by the LHEVCDecoderConfigurationRecord, stored in the LHEVCConfigurationBox.

If the samples of a track do not contain, neither natively nor through extraction, an HEVC base layer, then the sample entry type 'lhv1' or 'lhe1' shall be used and the sample entry shall contain an LHEVCConfigurationBox, as defined below. This includes an LHEVCDecoderConfigurationRecord, as defined in this document.

The lengthSizeMinusOne field in the L-HEVC and HEVC configurations in any given sample entry of L-HEVC and HEVC tracks sharing the same base track shall have the same value.

NOTE 5 When the base layer is coded by AVC, the lengthSizeMinusOne field for the L-HEVC tracks sharing the same base track have the same value, but that value does not have to be equal to the lengthSizeMinusOne field for the AVC track.

NOTE 6 When HEVC compatibility is indicated, it can be necessary to indicate a high enough level for the HEVC base layer to accommodate the bit rate of the entire bitstream, because all the NAL units are considered as included in the HEVC base layer and hence can be fed to the decoder, which is expected to discard those NAL unit it does not recognize. This case happens when the 'hvc1', 'hev1', 'hvc2', 'hev2', 'hvc3' or 'hev3' sample entry is used and both HEVC and L-HEVC configurations are present. However, when only an HEVC and L-HEVC configuration is present, on the contrary, the optimal level for the HEVC track is the one that only accommodates the bit rate of the base layer bitstream. In this case, caution is to be taken that the profile, tier and level (PTL) in the VPS base (i.e. the first PTL structure in a VPS) might be "greater" than the PTL of the base layer that is to be included in the HEVCDecoderConfigurationRecord. Rather, the PTL information in the first PTL structure in the VPS extension can be included in HEVCDecoderConfigurationRecord.

An LHEVCConfigurationBox may be present in an 'hvc1', 'hev1', 'hvc2', 'hev2', 'hvc3' or 'hev3' sample entry. In this case the HEVCLHVCSampleEntry definition below applies.

Table 10 shows for a video track all the possible uses of sample entries, configurations and the L-HEVC tools:

Table 10 — 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.

When an 'hvc2', 'hev2', 'lhv1', or 'lhe1' track does not contain extractors and does not contain VCL NAL units with TemporalId equal to 0, the track shall contain VCL NAL units with exactly one nuh_layer_id value.

9.5.3.1.2 Syntax

```
class LHEVCCConfigurationBox extends Box('lhvC') {
    LHEVCDecoderConfigurationRecord() LHEVCConfig;
}

class HEVCLHVCSampleEntry() extends HEVCSampleEntry() {
    LHEVCCConfigurationBox      lhvcconfig;
}

// Use this if track is not HEVC compatible
class LHEVCSampleEntry() extends VisualSampleEntry ('lhv1', or 'lhe1') {
    LHEVCCConfigurationBox      lhvcconfig;
    MPEG4ExtensionDescriptorsBox (); // optional
}
```

9.5.3.1.3 Semantics

When the sample entry is 'lhv1' or 'lhe1', Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\014LHEVC Coding" being recommended (\014 is 12, the length of the string "LHEVC Coding" in bytes). When the sample entry is 'hvc2', 'hev2', 'hvc3' or 'hev3', Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\013HEVC Coding" being recommended (\013 is 11, the length of the string "HEVC Coding" in bytes).

9.5.4 L-HEVC visual width and height

The width and height documented in the VisualSampleEntry shall be set as follows:

- If the sample entry type is 'hvc1' or 'hev1', the width and height documented in the VisualSampleEntry shall be set according to subclause 4.6 using only the base layer information.
- Otherwise, if the sample entry type is 'hvc2', 'hev2', 'hvc3' or 'hev3' and the track does not contain an L-HEVC configuration record, the width and height documented in the VisualSampleEntry shall be set according to the width and height of the base layer of the bitstream resulting either by resolving the extractors, when the track contains extractors, or by implicit reconstruction, when the track does not contain extractors.
- Otherwise (the sample entry type is 'hvc2', 'hev2', 'hvc3' or 'hev3' and the track contains an L-HEVC configuration record or the sample entry type is 'lhe1' or 'lhv1'), width and height shall be the maximum cropped frame dimensions of the decoded pictures, within the scope of the sample entry, of any layer in the track that is marked as an output layer of any output layer set.

9.5.5 Sync sample

Sync samples in a track of a particular sample entry type shall conform to the following constraints:

- If the sample entry type is 'hvc1' or 'hev1' and the track does not contain an L-HEVC configuration record, the specifications of 8.4.3 apply.
- Otherwise, if the sample entry type is 'hvc1' or 'hev1' and the track contains an L-HEVC configuration record, each sync sample of the track shall contain an IRAP picture in all layers represented by the track.
- Otherwise, if the sample entry type is 'hvc2', 'hev2', 'hvc3' or 'hev3' and the track contains extractors, each sync sample of the track shall contain an IRAP picture in all layers represented by the track either natively or through extraction.
- Otherwise, if the sample entry type is 'hvc2', 'hev2', 'hvc3' or 'hev3', and the track does not contain extractors, contains a sub-layer with TemporalId equal to 0 for one or more layers, and does not contain an L-HEVC configuration record, each sync sample of the track shall contain an IRAP picture with nuh_layer_id equal to 0.
- Otherwise, if the sample entry type is 'hvc2', 'hev2', 'lhv1', or 'lhe1' and the track does not contain extractors, contains a sub-layer with TemporalId equal to 0 for one or more layers, and contains an L-HEVC configuration record, each sync sample of the track shall contain an IRAP picture in all layers natively present in the track.
- Otherwise (the sample entry type is 'hvc2', 'hev2', 'lhv1', or 'lhe1', and the track does not contain extractors and does not contain any sub-layer with TemporalId equal to 0), each sync sample of the track shall contain a TSA picture with TemporalId equal to the lowest TemporalId present in the track.

NOTE 1 As can be observed from the list above, in some cases sync samples need the presence of an IRAP picture in more than one layer. If layer-specific information is needed, the 'sap' sample grouping can be used.

Sync samples are documented in the `SyncSampleBox` and, for movie fragments, with `sample_is_non_sync_sample` flag equal to 0, as specified in ISO/IEC 14496-12. Sync samples may be additionally documented by the stream access point 'sap' sample group.

When a sync sample contains natively or through extraction a BLA or CRA picture in a layer that does not have an alternative output layer, as defined by HEVC, in an output layer set represented by the track, there shall be no RASL pictures associated with the BLA or CRA picture.

Former versions of this specification allowed the presence of RASL pictures associated with BLA and CRA pictures contained in sync samples, although this is not in conformance with ISO/IEC 14496-12. For environments where HEVC file processors need information that BLA or CRA pictures contained in sync samples are not associated with RASL pictures, the brand 'nras', as defined in annex D.5, may be used to indicate compliance to this and later versions of this specification.

NOTE 2 The presence of RASL pictures that would be needed for output is not strictly in conformance with the sync sample definition of ISO/IEC 14496-12 and hence they cannot be present.

9.5.6 Independent and disposable samples box

If the `SampleDependencyTypeBox` is used in a track that is both HEVC and L-HEVC compatible, then care should be taken that the information provided by this box is true no matter what valid subset of the L-HEVC data (possibly only the HEVC data) is used. The 'unknown' values (value 0 of the fields `sample_depends_on`, `sample_is_depended_on`, and `sample_has_redundancy`) may be needed if the information varies.

9.5.7 Stream access point sample group

To provide information of all SAPs, the SAP sample group '`sap`' specified in ISO/IEC 14496-12 is used.

The semantics of `layer_id_method_idc` equal to 0 is specified in ISO/IEC 14496-12.

When `layer_id_method_idc` is equal to 0, a SAP is interpreted as follows:

- If the sample entry type is '`hvc2`', '`hev2`', '`lhv1`', or '`lhe1`', and the track does not contain extractors and does not contain any sub-layer with `TemporalId` equal to 0, a SAP specifies access to all the sub-layers present in the track.
- Otherwise, a SAP specifies access to all layers present natively or through extraction in the track.

NOTE If the sample entry type is '`hvc2`', '`hev2`', '`lhv1`', or '`lhe1`', and the track does not contain extractors and does not contain any sub-layer with `TemporalId` equal to 0, a TSA picture with `TemporalId` equal to the lowest `TemporalId` present in the track serves as a SAP.

When `layer_id_method_idc` is equal to 1, each bit in the field `target_layers` represents a layer carried in the track. Since this field is only 28 bits in length, the indication of SAPs in a track is constrained to a maximum of 28 layers. Each bit of this field starting from the least significant bit (LSB) shall be mapped to the list of `layer_id` values signalled in the Layer Information sample group ('`linfo`') associated with that sample, in ascending order of `layer_id` values.

For example, if a track carries layer with `layer_id` values 4, 10, and 29, then the `layer_id` 4 is mapped to the least significant bit, the `layer_id` 10 is mapped to the second least significant bit, and the `layer_id` 29 maps to the third least significant bit. A value of one in the bit signals that in the sample the mapped layer has a picture that is a SAP. In the previous example Table 11 gives an example of layer specific SAP information.

Table 11 — Bit pattern for the target_layer field of the 'sap' sample grouping for an example use case described above. (informative)

Bit pattern of target layers (LSB right most bit)	layer_id values (among 4, 10, 29) of SAPs
000000000000000000000000000010	{10}
0000000000000000000000000000100	{29}
0000000000000000000000000000101	{29, 4}

When SAP sample groups are used, they shall be used on all tracks carrying an L-HEVC bitstream. When extractors are used in a track, the SAPs of those layers referred to by extractors are also documented. The SAP documentation shall be consistent for any given layer.

9.5.8 The 'roll', 'rap', 'sync', 'tsas' and 'tsa' sample groups

When included in the base track, the information provided by the 'roll', 'rap', and 'sync' sample groups applies to the base layer only, not considering any other layers that are present in the samples.

When included in an 'hvc2', 'hev2', 'hvc3', or 'hev3' track that is not the base track and contains extractors, the information provided by the 'roll', 'rap', and 'sync' sample groups in SampleToGroupBox(es) with version equal to 0 applies to the track, after resolving extractors and aggregators contained in the track.

When included in an 'hvc2', 'hev2', 'lhv1', or 'lhe1' track that is not the base track, does not contain extractors, and natively contains only one layer (including its lowest sub-layer), the information provided by the 'roll', 'rap', and 'sync' sample groups in SampleToGroupBox(es) with version equal to 0 applies to that layer.

NOTE When a 'roll', 'rap', or 'sync' sample group concerns a predicted layer, it indicates characteristics that apply when all the reference layers of the predicted layer are available and decoded. The sample group can be used to initiate decoding of the predicted layer.

The presence of the 'sync', 'roll', and 'rap' sample groups in 'hvc2', 'hev2', 'hvc3', 'hev3', 'lhv1', or 'lhe1' tracks natively containing more than one layer or natively containing one layer but not its lowest sub-layer is disallowed in this version of ISO/IEC 14496-15, and parsers according to this version of ISO/IEC 14496-15 shall ignore SampleGroupDescriptionBoxes and SampleToGroupBoxes with grouping_type equal to 'sync', 'roll', or 'rap' within such tracks.

For the 'tsas' and 'tsa' sample groups, the specifications in 9.5.7 apply with 'sap' being replaced by 'tsas' and 'tsa', respectively, and "SAP" being replaced by "TSA point" and "STSA point", respectively.

9.5.9 Definition of a sub-sample for L-HEVC

The specifications of 8.4.8 apply.

9.5.10 Handling non-output samples

What is specified in 8.4.9 is not always applicable when multiple layers are involved. If what is specified in 8.4.9 cannot be followed, then the composition time of a sample shall be set as if the sample were an output sample (i.e., at least one picture of the sample is output). If the decoding of a sample results into no output pictures, the presentation of the sample is omitted and the duration of the previous sample is extended, so that the following samples have correct composition timing.

9.6 L-HEVC specific structures

9.6.1 External base layer sample group

9.6.1.1 Definition

When a multi-layer HEVC bitstream uses an external base layer (i.e., when an active VPS of the bitstream has `vps_base_layer_internal_flag` equal to 0 and `vps_base_layer_available_flag` equal to 1), an `LhvcExternalBaseLayerInfo` sample grouping may be used to signal parameters that are required for inter-layer prediction from the external base layer to a track containing L-HEVC samples. When a sample is linked to the zero index of this sample grouping, it means that no decoded base layer picture is used for the decoding of that sample.

9.6.1.2 Syntax

```
aligned(8) class LhvcExternalBaseLayerInfo() extends
VisualSampleGroupEntry('lbli')
{
    bit(1) reserved = '1'b;
    unsigned int(1) bl_irap_pic_flag;
    unsigned int(6) bl_irap_nal_unit_type;
    signed int(8) sample_offset;
}
```

9.6.1.3 Semantics

In this subclause, the term current sample refers to the sample that is associated with an entry in the `LhvcExternalBaseLayerInfo` sample group in an L-HEVC track.

`bl_irap_pic_flag` specifies the value of the `BlIrapPicFlag` variable for the associated decoded picture, when that decoded picture is provided as a decoded base layer picture for the decoding of the current sample.

`bl_irap_nal_unit_type` specifies, when `bl_irap_pic_flag` is equal to 1, the value of the `nal_unit_type` syntax element for the associated decoded picture, when that decoded picture is provided as a decoded base layer picture for the decoding of the current sample.

`sample_offset` gives the relative index of the associated sample in the track containing the base layer (i.e., the track referred to by the 'sbas' track reference). The decoded picture resulting from the decoding of the associated sample in the track containing the base layer is the associated

decoded picture that should be provided for the decoding of the current sample. `sample_offset` equal to 0 specifies that the associated sample has the same, or the closest preceding, decoding time compared to the decoding time of the current sample; `sample_offset` equal to 1 specifies that the associated sample is the next sample relative to the associated sample derived for `sample_offset` equal to 0; `sample_offset` equal to -1 specifies that the associated sample is the previous sample relative to the associated sample derived for `sample_offset` equal to 0, and so on.

9.6.2 The operating points information sample group

9.6.2.1 Definition

Applications are informed about the different operating points provided by a given L-HEVC bitstream and their constitution by using the Operating Points Information sample group ('oinf'). Each operating point is related to an output layer set, a max TemporalId value, and a profile, level and tier signaling. All this information is captured by the 'oinf' sample group. Apart from these information, this sample group also provides the dependency information between layers, the types of scalabilities coded in the L-HEVC bit stream, and the dimension identifiers that relate to any particular layer for a given scalability type.

For all tracks of an HEVC bitstream containing only NAL units with `nuh_layer_id` 0 and for which temporal sublayers are split across several tracks, there is no requirement for any track to carry 'oinf' sample group.

For all tracks of an L-HEVC bitstream, there shall be exactly one track among this set that carries an 'oinf' sample group. Except the tracks that carry the 'oinf' sample group, all other tracks of that L-HEVC bitstream shall have a track reference of type 'oref' to the track that carries the 'oinf' sample group they relate to.

NOTE An L-HEVC bitstream usually requires using only one track with 'oinf' sample group information. However, extractors enable creating new bitstreams with different profiles and levels not signaled in the 'oinf' of the original bitstream for which data is extracted. Using extractors can require additional 'oinf' to store the operation information of these new bitstreams.

For any particular sample in a particular track, the temporally collocated sample in another track is defined as the one with the same decoding time as that of this particular sample. For each sample `SN` in a track `TN` that has an 'oref' track reference to the track `Tk` that carries the 'oinf' sample group, the follow applies:

- If there is a temporally collocated sample `Sk` in the track `Tk`, then the sample `SN` is associated with the same 'oinf' sample group entry as the sample `Sk`.
- Otherwise, the sample `SN` is associated with the same 'oinf' sample group entry as the last of the samples in the track `Tk` that precede the sample `SN` in decoding time.

When several VPSs are referenced by an L-HEVC bitstream, it may be needed to include several entries in the sample group description box with grouping_type 'oinf'. For more common cases where a single VPS is present, it is recommended to use the default sample group description index and include the Operating Points Information sample group in the sample table box, rather than including it in each track fragment.

The grouping_type_parameter is not defined for the SampleToGroupBox with grouping type 'oinf'.

An operating point shall contain at least one other layer in addition to an external base layer.

9.6.2.2 Syntax

```
class OperatingPointsRecord {
    unsigned int(16) scalability_mask;
    bit(2) reserved = 0;
    unsigned int(6) num_profile_tier_level;
    for (i=1; i<=num_profile_tier_level; i++) {
        unsigned int(2) general_profile_space;
        unsigned int(1) general_tier_flag;
        unsigned int(5) general_profile_idc;
        unsigned int(32) general_profile_compatibility_flags;
        unsigned int(48) general_constraint_indicator_flags;
        unsigned int(8) general_level_idc;
    }
    unsigned int(16) num_operating_points;
    for (i=0; i<num_operating_points) {
        unsigned int(16) output_layer_set_idx;
        unsigned int(8) max_temporal_id;
        unsigned int(8) layer_count;
        for (j=0; j<layer_count; j++) {
            unsigned int(8) ptl_idx;
            unsigned int(6) layer_id;
            unsigned int(1) is_outputlayer;
            unsigned int(1) is_alterate_outputlayer;
        }
        unsigned int(16) minPicWidth;
        unsigned int(16) minPicHeight;
        unsigned int(16) maxPicWidth;
        unsigned int(16) maxPicHeight;
        unsigned int(2) maxChromaFormat;
        unsigned int(3) maxBitDepthMinus8;
        bit(1) reserved = 0;
        unsigned int(1) frame_rate_info_flag
        unsigned int(1) bit_rate_info_flag
        if (frame_rate_info_flag) {
            unsigned int(16) avgFrameRate;
```

```

        bit(6) reserved = 0;
        unsigned int(2) constantFrameRate;
    }
    if (bit_rate_info_flag) {
        unsigned int(32) maxBitRate;
        unsigned int(32) avgBitRate;
    }
}
unsigned int(8) max_layer_count;
for (i=0; i<max_layer_count; i++) {
    unsigned int(8) layerID;
    unsigned int(8) num_direct_ref_layers;
    for (j=0; j<num_direct_ref_layers; j++) {
        unsigned int(8) direct_ref_layerID;
    }
    for (j=0; j<16; j++) {
        if (scalability_mask & (1 << j))
            unsigned int(8) dimension_identifier;
    }
}
}

class OperatingPointsInformation extends VisualSampleGroupEntry ('oinf')
{
    OperatingPointsRecord oinf;
}

```

9.6.2.3 Semantics

scalability_mask: This field indicates the scalability types that are represented by the access units resolved from the samples that are associated with this 'oinf' sample group entry. Each bit in the scalability_mask field denotes a scalability dimension as coded in the scalability_mask_flag of the VPS extension syntax as defined in ISO/IEC 23008-2. The most significant bit in the scalability_mask field corresponds to scalability_mask_flag[0] of the VPS extension syntax. A one in a bit position indicates that the scalability dimension is present.

num_profile_tier_level: Gives the number of following profile, tier, and level combinations as well as the associated fields.

general_profile_space, general_tier_flag, general_profile_idc, general_profile_compatibility_flags, general_constraint_indicator_flags, and general_level_idc are defined in ISO/IEC 23008-2.

num_operating_points: Gives the number of operating points for which the information follows.

output_layer_set_idx is the index of the output layer set that defines the operating point. The mapping between output_layer_set_idx and the layer_id values shall be the same as specified by the VPS for an output layer set with index output_layer_set_idx.

max_temporal_id: Gives the maximum TemporalId of NAL units of this operating point.

layer_count: This field indicates the number of necessary layers, as defined in ISO/IEC 23008-2, of this operating point.

ptl_idx: Indicates the index, where value 1 is the first entry in the list of profile, tier, level in the OperatingPointsRecord, of the listed profile, level, and tier flags for a layer with identifier equal to **layer_id**. When the value of **ptl_idx** equals zero for a layer, that layer shall be assumed to have no profile, level, and tier signalled. **ptl_idx** shall be equal to 0 for an external base layer. When **ptl_idx** is equal to 0 for a layer that is not an external base layer, that layer shall not be an output layer or a layer that is a direct or indirect reference layer of any output layer of the operating point.

layer_id: provides the **nuh_layer_id** values for the layers of the operating point.

is_outputlayer: A flag that indicates if the layer is an output layer or not. A one indicates an output layer.

is_alterate_outputlayer: This flag when set indicates that this layer can be considered as an alternate output layer for this operating point. This flag is set to one if and only if one layer in the operating point has its **is_outputlayer** flag set.

minPicWidth specifies the least value of the luma width indicators as defined by the **pic_width_in_luma_samples** parameter in ISO/IEC 23008-2 for the stream of the operating point.

minPicHeight specifies the least value of the luma height indicators as defined by the **pic_height_in_luma_samples** parameter in ISO/IEC 23008-2 for the stream of the operating point.

maxPicWidth specifies the greatest value of the luma width indicators as defined by the **pic_width_in_luma_samples** parameter in ISO/IEC 23008-2 for the stream of the operating point.

maxPicHeight specifies the greatest value of the luma height indicators as defined by the **pic_height_in_luma_samples** parameter in ISO/IEC 23008-2 for the stream of the operating point.

maxChromaFormat specifies the greatest value of the **chroma_format_idc** indicator as defined by the **chroma_format_idc** parameter in ISO/IEC 23008-2 for the stream of the operating point.

maxBitDepthMinus8 specifies the greatest value of the luma and chroma bit depth indicators as defined by the **bit_depth_vps_luma_minus8** and **bit_depth_vps_chroma_minus8** parameters, respectively, in ISO/IEC 23008-2 for the stream of the operating point.

frame_rate_info_flag equal to 0 indicates that no frame rate information is present for the operating point. The value 1 indicates that frame rate information is present for the operating point.

bit_rate_info_flag equal to 0 indicates that no bitrate information is present for the operating point. The value 1 indicates that bitrate information is present for the operating point.

avgFrameRate gives the average frame rate in units of frames/(256 seconds) for the operating point. Value 0 indicates an unspecified average frame rate.

constantFrameRate equal to 1 indicates that the stream of the operating point is of constant frame rate. Value 2 indicates that the representation of each temporal layer in the stream of the operating point is of constant frame rate. Value 0 indicates that the stream of the operating point may or may not be of constant frame rate.

maxBitRate gives the maximum bit rate in bits/second of the stream of the operating point, over any window of one second.

avgBitRate gives the average bit rate in bits/second of the stream of the operating point.

max_layer_count: The count of all unique non-base layers in all of the operating points.

layerID: **nuh_layer_id** of a layer for which the all the direct reference layers are given in the following loop of **direct_ref_layerID**.

num_direct_ref_layers: The number of direct reference layers for the layer with **nuh_layer_id** equal to **layerID**.

direct_ref_layerID: **nuh_layer_id** of the direct reference layer.

`dimension_identifier`: Set to the value of the `dimension_id` field as specified in the VPS extension syntax as defined in ISO/IEC 23008-2.

9.6.3 The layer information sample group

The list of layers and sub layers a track carries is signalled in the Layer Information sample group as specified in subclause 4.15. The values of `irap_gdr_pics_in_layer_only_flag` and `completeness_flag` shall both be equal to 0.

Every L-HEVC track, including the base track (when coded with HEVC), shall carry a 'linfo' sample group.

When several VPSs are referenced by an L-HEVC bitstream, it may be needed to include several entries in the sample group description box with grouping_type 'linfo'. For more common cases where a single VPS is present, it is recommended to use the default sample group description index and include the Layer Information sample group in the sample table box, rather than including it in each track fragment.

9.6.4 The decoding time hint sample group

9.6.4.1 Definition

Each decoding time hint sample group description entry ('opth') records a delta time in terms of the clock ticks (given by `timescale` of the `MediaHeaderBox`). The corrected decoding time is defined as the sum of the delta time associated with a sample through the `SampleToGroupBox` of type 'opth' and the decoding time of the sample. The corrected decoding times conform to the hypothetical reference decoder of ISO/IEC 23008-2 operating according to a partitioning scheme where each layer is in its own bitstream partition, as defined in ISO/IEC 23008-2.

All `SampleToGroupBoxes` for the decoding time hint sample group shall include `grouping_type_parameter`. The `grouping_type_parameter` field is specified for the decoding time hint sample group as follows:

```
unsigned int(16) reserved = 0;
unsigned int(16) operating_point_index;
```

`operating_point_index` specifies the index of the operating point, as given in the associated `OperatingPointsInformation` sample group description, for which this sample group provides the corrected decoding times. A value of 0 indicates the first operating point in that sample group description.

9.6.4.2 Syntax

```
class OperatingPointDecodeTimeHint()
extends VisualSampleGroupEntry ('opth')
{
    signed int(32) delta_time;
}
```

9.6.4.3 Semantics

`delta_time` plus the decoding time (derived from the `TimeToSampleBox` and `TrackRunBoxes`, if any) provides the corrected decoding time of the associated sample. Time-scale units, as given by `timescale` of the `MediaHeaderBox` of this track, are used for the calculation of `delta_time`.

10 Storage of tiled HEVC and L-HEVC video streams

10.1 Overview

ISO/IEC 23008-2 video provides support for coding of rectangular regions called tiles. HEVC tiles do not have coding dependencies with other HEVC tiles in the same coded picture but may have coding dependencies with other HEVC tiles from previous coded pictures or may be independently decoded. This clause defines tools to describe and manipulate tiles, including:

- description of HEVC tiles and their temporal coding dependencies with other HEVC tiles,
- track(s) extracting one or several tiles from other track(s), and
- track(s) containing only data from one or several HEVC tiles for fast access to a rectangular region over a network.

A rectangular region for HEVC shall consist of one or more complete HEVC tiles in one or more complete slice segments that are within the same coded picture and contain no other HEVC tiles. The slice segments of a rectangular region may but need not be contiguous in decoding order.

Rectangular regions can be described through rectangular region visual sample group description entries (i.e., instances of `RectangularRegionGroupEntry`) with `rect_region_flag` equal to 1.

NOTE The `RectangularRegionGroupEntry` and `rect_region_flag` used to be called `TileRegionGroupEntry` and `tile_region_flag`.

If each sample of a track consists of NAL units of only one rectangular region, `SampleToGroupBox` of type 'trif' can be used to associate samples to the rectangular region, but this `SampleToGroupBox` of type 'trif' can be omitted if the default sample group description index is used (i.e., when the version of the `SampleGroupDescriptionBox` of type 'trif' is equal to or greater than 2). Otherwise, samples, NAL units, and rectangular regions are associated with each other through `SampleToGroupBoxes` of type 'nalm' and `grouping_type_parameter` equal to 'trif' and `SampleGroupDescriptionBox` of type 'nalm'. A `RectangularRegionGroupEntry` describes:

- a rectangular region composed of HEVC tiles, and
- coding dependencies between HEVC tiles in this rectangular region and HEVC tiles in other rectangular regions.

Each `RectangularRegionGroupEntry` is assigned a unique identifier, called `groupID`. This identifier can be used to associate NAL units in a sample to a particular `RectangularRegionGroupEntry`.

Positioning and size of rectangular regions are identified using luma sample coordinates.

When used with movie fragments, `RectangularRegionGroupEntry` can be defined for the duration of the movie fragment, by defining a new `SampleGroupDescriptionBox` in the track fragment box as defined in ISO/IEC 14496-12:2020, subclause 8.9.4. However, there shall not be any

RectangularRegionGroupEntry in a track fragment that has the same groupID as a RectangularRegionGroupEntry already defined.

For HEVC and L-HEVC tile tracks as defined in this document, the base region used in the RectangularRegionGroupEntry is the picture to which the tiles in a rectangular region associated with this rectangular region group entry belongs

Sub-tracks can be defined by grouping the different rectangular regions for this sub-track, using a TileSubTrackGroupBox in the sub-track definition box.

NAL units mapped to a rectangular region may either be carried in the video track, as usual, or in a separate track called tile track. Tile tracks are defined for both HEVC and L-HEVC.

10.2 NAL unit map entry

The specification of NAL unit map entry can be found in subclause 4.13.

10.3 Tile region group entry

The specification of tile region group entry can be found in subclause 4.14.

10.4 Tile sub track definition

10.4.1 Overview

A tile sub track describes one or more tile regions.

An additional descriptive attribute 'tile' may be used to indicate the sub track is a spatial part of the track.

Name	Attribute	Description
Tiling	'tile'	The sub-track is spatial part or tile of the track.

A tile sub track is defined as a group of tile regions using the TileSubTrackGroupBox.

10.4.2 TileSubTrackGroupBox

10.4.2.1 Definition

Box Type: 'tstb'
 Container: Sub Track Definition box ('strd')
 Mandatory: No
 Quantity: Zero or one

10.4.2.2 Syntax

```
aligned(8) class TileSubTrackGroupBox extends FullBox('tstb', 0, 0) {
    unsigned int(16) item_count;
    for(i = 0; i < item_count; i++) {
        unsigned int(16) tileGroupID;
    }
}
```

10.4.2.3 Semantics

The union of `tileGroupIDs` in this box describes the sub track defined by this box.

`item_count` counts the number of tile groups listed in this box.

`tileGroupID` is the identifier of the tile region group (`groupID`) contained in this sub track.
`groupID` is defined in `RectangularRegionGroupEntry`.

10.5 HEVC and L-HEVC tile track

10.5.1 Overview

There are cases where storing NAL units of HEVC or L-HEVC tile regions in different tracks are useful for easy access to one or a few particular tile regions. For such cases, tile tracks may be created using the `HEVCTileSampleEntry`, `HEVCTileSSHInfoSampleEntry` or `LHEVCTileSampleEntry` sample description format.

An HEVC or L-HEVC tile track is a video track in which there is a 'tbas' track reference to the HEVC or L-HEVC track, respectively. This HEVC or L-HEVC track is referred to as the HEVC or L-HEVC tile base track of the HEVC or L-HEVC tile track, respectively. The sample entry type for an HEVC tile track is 'hvt1' or 'hvt3'. The sample entry type for an L-HEVC tile track is 'lht1'. The use of sample entry types for tile base tracks is defined in 10.5.5.

A tile track or tile base track shall not include extractors. A tile base track shall not include VCL NAL units. A tile track shall not carry VCL NAL units belonging to more than one layer.

For any picture unit carried by samples in a tile base track and a number of tile tracks, all the NAL units that apply to the entire coded picture shall be carried in the tile base track. These NAL units include (but are not limited to) parameter sets (VPS, SPS, and PPS) as well as EOB and EOS NAL units, when present. The `SampleDescriptionBox` of a tile track shall not carry NAL units that apply to any entire coded picture, either. The NAL units that do not apply to a tile shall not be carried in the tile track containing that tile. The NAL units that apply to a tile shall be carried in the tile track containing that tile.

NOTE 1 When a NAL unit applies to multiple tiles, these tiles can only be stored in the same tile track.

A tile track shall only depend on the tile base track and shall be independent from any other tile track that includes VCL NAL units of the same layer as this tile track.

An HEVC or L-HEVC sample of a tile track consists of one or more complete tiles in one or more complete slice segments.

An HEVC or L-HEVC sample in the tile base track is considered as a sync sample if and only if the sample resulting from the merging of the corresponding samples of all tile tracks and the tile base track, as defined in 10.5.4, is a sync sample as defined for a non tiled HEVC or L-HEVC track, respectively, that has the sample entry type equal to that in the tile base track and the same configuration records as those in the tile base track.

A sample stored in a tile track is considered as a sync sample when the respective sample in the tile base track is a sync sample.

The `SubSampleInformationBox` and sample groupings defined for regular HEVC or L-HEVC tracks also apply to HEVC or L-HEVC tile tracks, respectively.

NOTE 2 Many aspects of the tile track can be discovered by inspecting the base track if they are not explicitly declared in the tile track; for example, a 'roll' sample group.

NOTE 3 An implementation can decide to decode only a subset of the tiles of an HEVC or L-HEVC stream. In this case, it can use the dependency information in the `RectangularRegionGroupEntry` sample group descriptions to discard unneeded NAL units.

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.

10.5.2 Sample entry name and format for HEVC tile tracks

10.5.2.1 Definition

Sample Entry Type: 'hvt1'
 Container: Sample Description Box ('stsd')
 Mandatory: No
 Quantity: Zero or more sample entries may be present

This sample entry describes media samples of an HEVC tile track. The width and height of the `VisualSampleEntry` for an HEVC tile track (sample entry type 'hvt1') 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 'hvt1' sample description.

10.5.2.2 Syntax

```
class HEVCTileSampleEntry() extends VisualSampleEntry ('hvt1'){
    HEVCTileConfigurationBox  config(); // optional
}
```

```

class HEVCTileConfigurationBox extends Box('hvtC') {
    HEVCTileTierLevelConfigurationRecord() HEVCTileTierLevelConfig;
}

aligned(8) class HEVCTileTierLevelConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    bit(7) reserved = 0;
    unsigned int(1) mcts_tier_flag;
    unsigned int(8) mcts_level_idc;
    bit(8) reserved = 0;
}

```

10.5.2.3 Semantics

The HEVCTileSampleEntry shall not contain any HEVCCConfigurationBox, LHEVCCConfigurationBox or MPEG4ExtensionDescriptorsBox; these boxes are found in the tile base track's sample description. Other optional boxes may be included. There are usually as many entries in the SampleDescriptionBox of an HEVC tile track as there are entries in the SampleDescriptionBox of the tile base track.

Optionally, the HEVCTileSampleEntry may contain one HEVCTileTierLevelConfigurationRecord, used to indicate the tier and level information in the case the tile region in this tile track is a motion-constrained tile or tile set.

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\020HEVC Tile Coding" being recommended; the first byte is a count of the remaining bytes, here represented by \020, which (being octal 20) is 16 (decimal), the number of bytes in the rest of the string.

mcts_tier_flag, mcts_level_idc are set to the values of the fields mcts_tier_flag and mcts_level_idc in the temporal motion-constrained tile sets SEI message if the tile region in the tile track is a temporal motion-constrained tile set as defined in ISO/IEC 23008-2. If the tile region in the tile track is not the same as any of the temporal motion-constrained tile set in the stream or no temporal motion-constrained tile sets SEI message is present in the stream, an appropriate value according to the temporal motion-constrained tile sets SEI as defined in ISO/IEC 23008-2 shall be set.

10.5.3 Sample entry name and format for L-HEVC tile tracks

10.5.3.1 Definition

Sample Entry Type: 'lht1'
 Container: Sample Description Box ('stsd')
 Mandatory: No
 Quantity: Zero or more sample entries may be present

This sample entry describes media samples of an L-HEVC tile track. The width and height of the VisualSampleEntry for an L-HEVC tile track (sample entry type 'lht1') shall be set to the width and height of the minimum bounding box enclosing all tile regions of all layers contained in the track. The layout information in the track header (i.e., layer, matrix, width and height) of an L-HEVC tile track shall be ignored by file parsers. CleanApertureBox and PixelAspectRatioBox shall not be present in an 'lht1' sample description.

10.5.3.2 Syntax

```
class LHEVCTileSampleEntry() extends VisualSampleEntry ('lht1'){
}
```

10.5.3.3 Semantics

The `LHEVCTileSampleEntry` shall not contain any `LHEVCConfigurationBox`, `HEVCConfigurationBox` or `MPEG4ExtensionDescriptorsBox`; these boxes are found in the tile base track's sample description. Other optional boxes may be included. There are usually as many entries in the `SampleDescriptionBox` of an L-HEVC tile track as there are entries in the `SampleDescriptionBox` of the tile base track.

`Compressorname` in the base class `VisualSampleEntry` indicates the name of the compressor used with the value `"\022L-HEVC Tile Coding"` being recommended; the first byte is a count of the remaining bytes, here represented by `\022`, which (being octal 22) is 18 (decimal), the number of bytes in the rest of the string.

10.5.4 Bitstream reconstruction from tile base and tile tracks

Since extractors cannot be used in a tile base track, a tile base track shall indicate the tile ordering using a `'sabt'` track reference to the tile tracks. 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. The `'sabt'` track reference may only be used to refer to tile tracks from a tile base track. When the `'sabt'` track reference is present in a track, the samples of this track shall not use aggregators.

The bitstream is reconstructed as follows:

- a) For a tile base track and a number of tile tracks carrying VCL NAL units of one layer, a picture unit is firstly reconstructed to consist of the following NAL units in the order listed:
 - 1) If the picture unit corresponds to a sample that is the first sample of a set of samples associated with a sample entry, the parameter sets and SEI NAL units contained in the sample entry
 - 2) NAL units in the sample of the tile base track
 - 3) NAL units in the samples of the tile tracks in the order of the `'sabt'` track references

Then the following steps apply, in the order listed, for the above-reconstructed picture unit:

- 4) If there is one or more than one EOS NAL unit present, an EOS NAL unit is placed at the end of the picture unit and any other EOS NAL unit is removed.
 - 5) If there is one or more than one EOB NAL unit present, one EOB NAL unit is placed at the end of the picture unit; any other EOB NAL unit is removed.
- b) If only one layer is involved, the reconstructed picture unit is the access unit. Otherwise, the access unit is reconstructed from all the picture units of the involved layers as specified in 9.5.2.2.
- c) Finally, the bitstream is reconstructed from the reconstructed access units as specified in 9.5.2.2.

NOTE Picture units, access units, and bitstreams reconstructed as above do not always conform to ISO/IEC 23008-2. For example, for a picture unit reconstructed as above, some VCL NAL units covering some tiles of the picture can be missing due to the regions represented by those tiles are not needed by the application.

10.5.5 Sample entry names for tile base tracks

When no NAL unit is included in a sample in a tile base track (i.e. all the NAL units are in the tile tracks), the size of the sample may be 0, and the sample documents the timing and other sample properties of the sample.

The sample entry names of a tile base track and the corresponding tile tracks shall be one of the rows in Table 12.

Table 12 — Sample entry names for a tile base track and the corresponding tile tracks

Sample entry name for a tile base track	Sample entry name for a corresponding tile track
hvc2	hvt1
hev2	hvt1
lhv1	lht1
lhe1	lht1

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 4.2.3 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 `rbp_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 ('std')

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 `rbps_slice_segment_trailing_bits()` of a typically independent slice segment. No other data shall be present in samples of 'hvt2' tracks.

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 reference the 'hvt2' track need to also be removed. If an extractor track is removed from a file, all 'hvt2' tracks that the extractor track references need to be removed provided that there is no other extractor track referencing them.

10.6.2.2 Syntax

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

10.6.2.3 Semantics

The `HEVCSliceSegmentDataSampleEntry` shall not contain any `HEVCCConfigurationBox`, `LHEVCCConfigurationBox` 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.

11 VVC elementary streams and sample definitions

11.1 Overview

11.1.1 General

The Versatile Video Coding (VVC) standard (ISO/IEC 23090-3) offers increased coding efficiency and many features for the systems that use it. To enable the best visibility of, and access to, those features, and to enhance the opportunities for the interchange and interoperability of media, this clause specifies the storage format for both single-layer VVC bitstreams and multilayer VVC bitstreams with more than one layer.

This clause specifies the storage of layers and temporal sublayers of a VVC bitstream in more than one track. Multi-track storage could be used for accessing and playing the layers or temporal sublayers selectively. Subclause 11.1.4 contains an overview of storage options for VVC bitstreams with multiple layers or sublayers.

This clause specifies the storage of rectangular regions of one or more VVC subpictures or slices as individual VVC subpicture tracks. Such storage could be used for accessing VVC subpicture tracks selectively in immersive video applications where only a subset of the video content is viewed at any moment. Subclause 11.1.5 contains an overview for storage using VVC subpicture tracks.

This clause enables storing non-VCL NAL units in VVC non-VCL tracks. Such storage could be used for sharing the same non-VCL NAL units among multiple VVC tracks or VVC subpicture tracks and accessing non-VCL NAL units only when needed.

Subclause 11.5.4 specifies a process for reconstructing of a VVC bitstream that readers shall use whenever multiple tracks carry NAL units of the bitstream.

11.1.2 Background: features of VVC

The storage of VVC content uses the existing capabilities of the ISO base media file format but also defines extensions to support the following features of the VVC codec:

a) Parameter sets and DCI and OPI NAL units:

The VPS, SPS, and PPS mechanism decouples the transmission of infrequently changing information from the transmission of coded block data. Each coded picture references the PPS containing its decoding parameters. In turn, the PPS references an SPS that contains sequence level decoding parameter information, and the SPS references a VPS that contains global decoding parameter information across layers. When the referenced VPS ID has a value zero, the bitstream contains only one layer and there is no VPS actually referenced.

The APS mechanism allows efficient storage and transmission of information that typically has a large number of variations within a bitstream and is often updated across pictures, such as parameters for the adaptive loop filter (ALF), parameters for the adaptive in-loop resampler, also known as luma mapping with chroma scaling (LMCS), and parameters for a scaling list, which associates each frequency index with a scale factor for a scaling process specified in ISO/IEC 23090-3. APSs carrying parameters for ALF, LMCS, and scaling list are also referred to as ALF APS, LMCS APS, and scaling list APS, respectively. Each slice containing the coded block data may reference one or more APSs containing ALF parameters for the slice. Moreover, each picture header may reference APSs of any type.

A VVC bitstream may contain DCI NAL units containing parameters describing the maximum capability needed for decoding the entire bitstream.

A VVC bitstream may also contain OPI NAL units containing the indication of the operating point, which is determined by the target output layer set and target highest TemporalId.

b) Picture header:

A picture header (PH) structure includes parameters that have the same value for all slices of a picture, is present for each picture, and is present either in its own PH NAL unit or directly in the slice header (SH). A PH structure can only be included in the SH of a slice if it is the only slice in the picture. The presence of a PH structure in a SH is indicated by the first bit in the SH. Within a CLVS, PH

structures can only be either all in PH NAL units or all in SHs. When a PH NAL unit is present, some parameters may be present either in the PH NAL unit or in the SH of slices associated with the PH NAL unit.

c) Subpictures:

A VVC subpicture is a rectangular region of one or more slices within a picture. A VVC subpicture contains one or more complete tiles or a part of only one tile. An encoder may treat the subpicture boundaries like picture boundaries and may turn off loop filtering across the subpicture boundaries. Thus, it is possible to encode subpictures so that selected subpictures can be extracted from VVC bitstream(s) or merged to a destination VVC bitstream. Furthermore, such VVC bitstream extraction or merging operations can be performed without modifications of the VCL NAL units. The subpicture identifiers (IDs) for the subpictures that are present in the bitstream may be indicated in the SPS(s) or PPS(s).

11.1.3 Types of tracks for carriage of VVC elementary streams

This document specifies the following types of tracks for carriage of VVC elementary streams:

- a) VVC track
- b) VVC non-VCL track

NOTE 1 A VVC non-VCL track could contain APSs, which carry ALF, LMCS, or scaling list parameters, with or without other non-VCL NAL units, stored in and transmitted through a track that is separate from the track containing the VCL NAL units. A VVC non-VCL track could also contain picture header NAL units, with or without APS NAL units, and with or without other non-VCL NAL units, stored in and transmitted through a track that is separate from the track containing the VCL NAL units.

c) VVC subpicture track.

NOTE 2 VVC non-VCL tracks and VVC subpicture tracks enable an optimal delivery of VVC video in streaming applications as follows. These tracks can each be carried in DASH representations of their own, and for decoding and rendering of a subset of the tracks, the DASH representations containing the subset of the VVC subpicture tracks as well as the DASH representation containing the non-VCL tracks can be requested by the client, segment by segment. This way, redundant transmission of APSs and other non-VCL NAL units can be avoided, and transmission of unnecessary subpictures can also be avoided.

NOTE 3 When a VVC subpicture track contains one or more complete slices but not all slices of a subpicture, all slices in the subpicture track belong to the same subpicture, and any VVC merge base track that references the subpicture track, through the 'subp' track reference, also references the subpicture track(s) that contain(s) the rest of the slices from the same subpicture.

11.1.4 Overview of VVC storage with multiple layers or sublayers

A VVC elementary stream with multiple layers can be placed in tracks in several ways, among which are the following:

- a) all the layers in one track, where all the layers correspond to an operating point;
- b) all the layers in one track, where there is no operating point that contains all the layers;

- c) one or more layers or sublayers in individual tracks, where a bitstream containing all samples of indicated one or more tracks corresponds to an operating point;
- d) one or more layers or sublayers in individual tracks, where there is no operating point that contains all NAL units of a set of one or more tracks.

The VVC file format allows storage of one or more layers into a track. Storage of multiple layers per track can be used. For example, when a content provider wants to provide a multi-layer bitstream that is not intended for subsetting, or when the bitstream has been created for a few pre-defined sets of output layers where each layer corresponds to a view (for example stereo pair), tracks can be created accordingly.

When a VVC bitstream is represented by multiple tracks and a player uses an operating point for which the layers are stored in multiple tracks, the player shall reconstruct VVC access units before passing them to the VVC decoder as specified in subclause 11.5.4.

A VVC operating point may be explicitly represented by a track, i.e., each sample in the track contains an access unit natively or through resolving 'subp' or 'recl' track references (when present), and through resolving 'vvcN' track references (when present). The access units contain NAL units from all the layers and sublayers that are part of the operating point.

The storage of VVC bitstreams is supported by structures such as the

- a) sample entry,
- b) VVC bitstream entity group ('vvcB'),
- c) Operating Points Information ('vopi') sample group,
- d) Operating Points Entity Group ('opeG').

When the track does not natively contain a VVC bitstream and does not represent a VVC bitstream after resolving 'subp', 'recl' and 'vvcN' track references (when present), one of the following shall apply:

- The track has an 'vreg' track reference to an 'opeG' entity group.
- The track has a 'vref' track reference to a track that contains a 'vopi' sample group.
- The track contains a 'vopi' sample group and all layers and sublayers specified in the 'vopi' sample group.

NOTE For example, when a single-layer VVC bitstream contains two temporal sublayers stored in different tracks, the track that contains the sublayer with TemporalId equal to 1 contains an 'vref' track reference to the track that contains the sublayer with TemporalId equal to 0.

The structures within a sample entry provide information for the decoding or use of the samples, in this case coded video and non-VCL data information, that are associated with that sample entry.

The Operating Points Information sample group records information about operating points such as the layers and sublayers that constitute the operating point, dependencies (if any) between them, the profile, tier, and level parameter of the operating point, and other such operating point relevant information.

The Operating Points Entity Group records information about operating points such as the layers and sublayers that constitute the operating point, dependencies (if any) between them, the profile, tier, and level parameter of the operating point, and other such operating point relevant information, as well as, identification of the tracks carrying each operating point.

The information in these sample groups, combined with using track references to find tracks, or in the operating points entity group is sufficient for a reader to choose an operating point in accordance with its capabilities, identify the tracks that contain the relevant layers and sublayers needed to decode the chosen operating point, and efficiently extract them.

11.1.5 Overview of VVC storage with VVC subpictures

11.1.5.1 Options to store VVC subpictures

VVC subpictures can be stored in different ways:

- a) VVC subpictures can be stored into several different 'vvc1'/'vvi1' or 'vvs1' tracks, i.e., VVC subpicture tracks. VVC subpictures stored in different VVC subpicture tracks can be merged into a single VVC bitstream based on a VVC merge base track that references the VVC subpicture tracks or based on a subpicture entity group.
- b) VVC subpictures of a VVC bitstream can be stored into a single 'vvc1'/'vvi1' track, from which a subset of VVC subpictures can be extracted into another VVC bitstream based on a VVC extraction base track.

11.1.5.2 VVC subpicture tracks, VVC merge base tracks, and VVC subpicture entity groups

VVC subpicture tracks enable the storage of VVC subpictures as separate tracks so that any combination of subpictures can be streamed or decoded. VVC subpicture tracks enable representing rectangular regions of the same video content at different bitrates or resolutions. Consequently, bitrate or resolution emphasis on regions can be adapted dynamically by selecting the VVC subpicture tracks that are streamed or decoded.

It is possible to create VVC subpicture tracks in two ways, presented in Figure 4:

- a) Encoding the video content with multiple subpictures into a VVC bitstream, extracting the coded subpicture sequences from the VVC bitstream, and storing each extracted coded subpicture sequence as a 'vvs1' VVC subpicture track.
- b) Splitting the uncompressed video before encoding into multiple uncompressed subpicture sequences, encoding each uncompressed sequence as a VVC bitstream, and storing each VVC bitstream as a 'vvc1'/'vvi1' track.

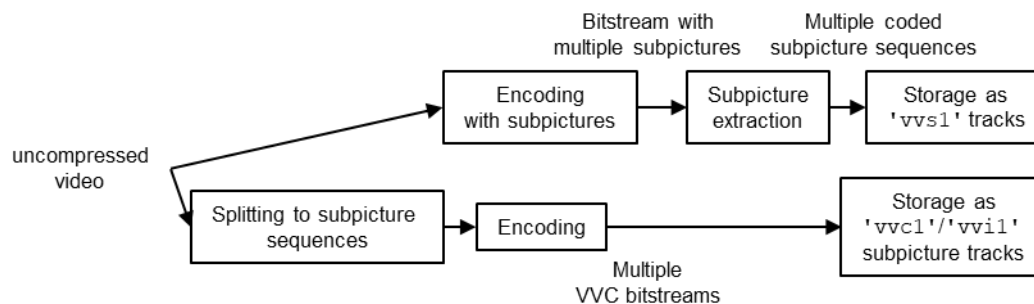


Figure 4 — Possibilities for creating VVC subpicture tracks

It is possible to mix the above-described ways of creating the VVC subpicture tracks. For example, after extracting coded subpictures, they could be stored as 'vvc1'/'vvi1' tracks, which would require creating some high-level syntax structures, such as parameter sets.

One or more VVC merge base tracks are created for a set of VVC subpicture tracks or 'alte' track groups of VVC subpicture tracks. A VVC merge base track references to VVC subpicture tracks or 'alte' track groups of VVC subpicture tracks through a 'subp' track reference. In addition, a VVC merge base track contains a subpicture order ('spor') sample group, which specifies, either implicitly or explicitly, an order of 'subp' track references. A VVC merge base track represents a VVC bitstream that is formed by merging the referenced VVC subpicture tracks in the order specified by the subpicture order sample group. When an 'alte' track group is referenced, readers may select any VVC subpicture track from the 'alte' track group for merging. When a track reference index of the same 'alte' track group is included multiple times in a 'spor' sample group description entry, a reader is expected to resolve it to different VVC subpicture tracks. A reader shall reconstruct a VVC bitstream by following the bitstream reconstruction process for a VVC merge base track, as specified in subclause 11.5.4.

NOTE A VVC subpicture track that does not contain content intended for displaying (referred to as a substitute subpicture track in this NOTE) can be indicated with a box header flag of `VvcConfigurationBox` or `VvcNALUConfigBox` in the sample entry and can be used for either or both of the following purposes: 1) A substitute subpicture track can be referenced by a VVC merge base track when the other referenced VVC subpicture tracks would not fill in the entire picture area. 2) A player can select a substitute subpicture track from an 'alte' track group, when a VVC merge base track references the 'alte' track group and no other VVC subpicture track of the 'alte' track group is available, e.g. due to a delayed transmission.

Subpicture entity groups may be created for representing combinations of VVC subpicture tracks that could be merged as VVC bitstreams. A reader needs to create some high-level structures of the bitstream, such as parameter sets, when merging VVC subpicture tracks as suggested by subpicture entity groups.

Annex G contains examples of the usage of VVC merge base tracks and VVC subpicture tracks.

11.1.5.3 VVC extraction base track extracting subpictures from a VVC track

VVC subpictures of a VVC bitstream can be stored into a single 'vvc1'/'vvi1' track. In this case, the individual VVC subpictures can be reconstructed from a 'vvc1'/'vvi1' VVC extraction base track with a track reference 'recr' to the track containing all VVC subpictures of the VVC bitstream, as presented in Figure 5.

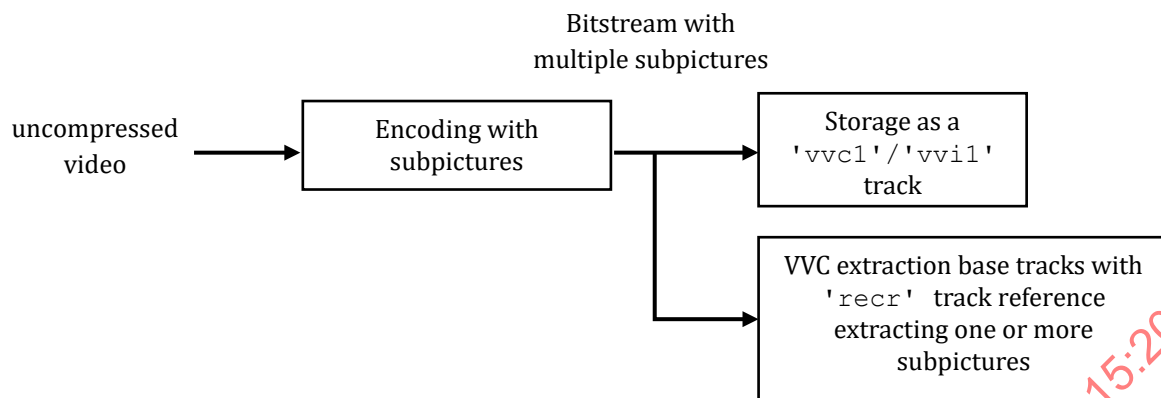


Figure 5 — VVC extraction base tracks for subpicture extraction

One or more VVC extraction base tracks are created for extracting one or more VVC subpictures from the 'vvc1'/'vvi1' track with all VVC subpictures. Such VVC extraction base tracks reference to the 'vvc1'/'vvi1' track with all VVC subpictures through a 'recr' track reference. In addition, a VVC extraction base track contains a rectangular region order ('rro') sample group, which specifies an order of rectangular regions to be extracted from the track referenced through the 'recr' track reference. A VVC extraction base track represents a VVC bitstream that is formed by extracting the referenced VVC subpictures corresponding to the indicated rectangular regions in the order specified by the rectangular region order sample group. A reader shall reconstruct a VVC bitstream by following the bitstream reconstruction process for a VVC extraction base track, as specified in subclause 11.5.4.

11.1.6 Overview of rectangular regions carried in a VVC bitstream

This document provides support for describing rectangular regions composed of either:

- a sequence of one or more VVC subpictures that forms a rectangular region, or
- a sequence of one or more complete slices that forms a rectangular region.

Rectangular regions can be described through rectangular region visual sample group description entries (i.e., instances of `RectangularRegionGroupEntry`) with `rect_region_flag` equal to 1.

If each sample of a track consists of NAL units of only one rectangular region, `SampleToGroupBox` of type 'trif' can be used to associate samples to the rectangular region, but this `SampleToGroupBox` of type 'trif' can be omitted if the default sample group description index is used (i.e., when the version of the `SampleGroupDescriptionBox` of type 'trif' is equal to or greater than 2). Otherwise, samples, NAL units, and rectangular regions are associated with each other through `SampleToGroupBoxes` of type 'nalm' and `grouping_type_parameter` equal to 'trif' and `SampleGroupDescriptionBox` of type 'nalm'. A `RectangularRegionGroupEntry` describes:

- a rectangular region,
- coding dependencies between this rectangular region and other rectangular regions.

Each `RectangularRegionGroupEntry` is assigned a unique identifier, called `groupID`. This identifier can be used to associate NAL units in a sample to a particular

`RectangularRegionGroupEntry`. When multiple tracks carry a VVC elementary stream, `groupID` values should be unique among all the tracks carrying the VVC elementary stream and an identical `groupID` value in any of the tracks carrying the VVC elementary stream should indicate the same rectangular region.

Positioning and size of rectangular regions are identified using luma sample coordinates.

When used with movie fragments, `RectangularRegionGroupEntry` can be defined for the duration of the movie fragment, by defining a new `SampleGroupDescriptionBox` in the track fragment box as defined in ISO/IEC 14496-12:2020, subclause 8.9.4. However, there shall not be any `RectangularRegionGroupEntry` in a track fragment that has the same `groupID` as a `RectangularRegionGroupEntry` already defined.

The base region used in the `RectangularRegionGroupEntry` is the picture to which the NAL units in a rectangular region associated with this rectangular region group entry belongs.

If there is any change in the base region size in consecutive samples, samples should be associated with different `RectangularRegionGroupEntry` entries reflecting their respective base region size.

NAL units mapped to a rectangular region may either be carried in a VVC track or in a separate track called VVC subpicture track.

11.2 Sample and configuration definition

11.2.1 Sample format of VVC tracks and VVC subpicture tracks

The sample format of VVC tracks and VVC subpicture tracks conforms to that in subclause 4.2.3.

A VVC merge base track shall not contain VCL NAL units and shall not be referred to by a VVC track through a '`vvcN`' track reference.

A VVC extraction base track shall not contain VCL NAL units and shall not be referred to by a VVC track through a '`vvcN`' track reference.

A sample of a VVC subpicture track contains either of the following:

- One or more complete subpictures as specified in ISO/IEC 23090-3 that form a rectangular region.
- One or more complete slices as specified in ISO/IEC 23090-3 that form a rectangular region.

11.2.2 Sample format of VVC non-VCL tracks

A sample in a VVC non-VCL track contains only one or more non-VCL NAL units. The sample format conforms to that specified in subclause 4.2.3 with `DecoderConfigurationRecord` being the `config` field of the sample entry.

The following constraints apply separately for each sample of a VVC non-VCL track, hereafter referred to as the non-VCL sample. Let the VVC sample be the sample for which both of the following are true:

- The VVC sample resides in a VVC track or a VVC subpicture track that contains a 'vvcN' track reference to the VVC non-VCL track.
- The VVC sample is aligned in decoding time with the non-VCL sample.

Let the referenced sample entry be the sample entry that the VVC sample references.

When the referenced sample entry has type 'vvc1' or 'vvi1' and the value of `array_completeness` in the referenced sample entry is equal to 1 for DCI, VPS, SPS, or PPS NAL units, the non-VCL sample shall not contain any DCI, VPS, SPS, or PPS NAL units, respectively.

When the referenced sample entry has type 'vvs1', the non-VCL sample shall not contain any DCI, OPI, VPS, SPS, PPS, AUD, PH, EOS, and EOB NAL units.

11.2.3 Canonical order and restrictions

The canonical stream format for VVC is an elementary stream that consists of NAL units specified in ISO/IEC 23090-3 and satisfies the following conditions in addition to the general conditions in subclause 4.2.2:

- If all VCL NAL units for a single picture are stored in the same track, they shall be contained within the sample whose decoding time and composition time are those of the picture. Otherwise (i.e., when pieces of a coded picture are stored in samples of different tracks), each of these samples shall have the decoding time and composition time that are those of the coded picture.

NOTE 1 For example, when VVC subpictures of a coded picture are stored in different VVC subpicture tracks, the samples of the VVC subpicture tracks that belong to the same coded picture have the same decoding and composition times that are equal to those of the coded picture.

- **Access unit delimiter NAL units:** The constraints obeyed by access unit delimiter NAL units are defined in ISO/IEC 23090-3.
- **DCI NAL units, VPSs, SPSS, and PPSs:** A VPS, SPS, or PPS to be used for decoding a picture shall be present prior to the sample containing that picture or in the sample for that picture.
- **APSS:** A prefix APS NAL unit is constrained to be present before the first VCL NAL unit of a picture unit (PU). Prefix APS NAL units with a particular value of `aps_adaptation_parameter_set_id` and a particular value of `aps_params_type` within a PU are allowed to be present but required to have the same content. A suffix APS NAL unit is constrained to be present following the last VCL NAL unit of the PU. Suffix APS NAL units with a particular value of `aps_adaptation_parameter_set_id` and a particular value of `aps_params_type` within a PU are allowed to be present but required to have the same content.
- **Filler data:** Video data is naturally represented as variable bit rate in the file format and should be filled for transmission if needed. Filler data NAL units and Filler payload SEI messages shall not be present in samples when the sample entry does not also permit in-stream parameter sets.

NOTE 2 The removal or addition of Filler Data NAL units, start codes, SEI messages or filler payload SEI messages can change the bitstream characteristics with respect to conformance with the HRD when operating the HRD in the CBR mode as specified in ISO/IEC 23090-3:2021, Annex C.

The following constraints apply for storing a canonical stream in samples and sample entries of one or more tracks:

- The DCI NAL units (if any), VPSs (if any), SPSs, and PPSs shall be stored only in the sample entry when the sample entry name is 'vvc1', and may be stored in the sample entry and the samples when the sample entry name is 'vvi1'.

NOTE 3 Storing DCI NAL units, VPSs, SPSs, and PPSs in the sample entries provides a simple and static way to supply decoding capability information and parameter sets. Storing these NAL units in samples on the other hand is more complex but allows for more dynamism in the case of parameter set updates (a particular parameter set's content is changed but using the same ID) and in the case of adding additional parameter sets. A decoder initializes with these parameter sets in the sample entry, and then updates using these parameter sets as they occur in the elementary stream, starting from any sample marked as a sync sample. Such updating can replace these parameter sets with a new definition using the same identifier. Each time the sample entry changes, the decoder re-initializes with these parameter sets included in the sample entry.

- LMCS APS NAL units and scaling list APS NAL units may be stored either in the samples and/or sample entries of VVC tracks or in the samples of VVC non-VCL tracks but not both at the same time.
- ALF APS NAL units may be stored in the samples and/or sample entries of VVC tracks, in the samples of VVC non-VCL tracks, or both at the same time.
- Picture header NAL units may be stored either in the samples of VVC tracks or in the samples of VVC non-VCL tracks, but not both at the same time.
- SEI messages of declarative nature may be stored in the sample entry; there is no prescription about removing such SEI messages from the samples.

11.2.4 Decoder configuration information

11.2.4.1 VVC profile-tier-level record

11.2.4.1.1 Definition

This subclause specifies the profile-tier-level syntax structure for ISO/IEC 23090-3 video content stored in a file.

11.2.4.1.2 Syntax

```

aligned(8) class VvcPTLRecord(num_sublayers) {
    bit(2) reserved = 0;
    unsigned int(6) num_bytes_constraint_info;
    unsigned int(7) general_profile_idc;
    unsigned int(1) general_tier_flag;
    unsigned int(8) general_level_idc;
    unsigned int(1) ptl_frame_only_constraint_flag;
    unsigned int(1) ptl_multi_layer_enabled_flag;
    unsigned int(8*num_bytes_constraint_info - 2) general_constraint_info;
    for (i=num_sublayers - 2; i >= 0; i--)
        unsigned int(1) ptl_sublayer_level_present_flag[i];
    for (j=num_sublayers; j<=8 && num_sublayers > 1; j++)
        bit(1) ptl_reserved_zero_bit = 0;
    for (i=num_sublayers-2; i >= 0; i--)
        if (ptl_sublayer_level_present_flag[i])
            unsigned int(8) sublayer_level_idc[i];
    unsigned int(8) ptl_num_sub_profiles;
    for (j=0; j < ptl_num_sub_profiles; j++)
        unsigned int(32) general_sub_profile_idc[j];
}

```

11.2.4.1.3 Semantics

num_bytes_constraint_info is used to specify the length of the general_constraint_info field. The length of the general_constraint_info field is num_bytes_constraint_info * 8 - 2 bits. The value shall be greater than 0. The value equal to 1 indicates that the gci_present_flag in the general_constraint_info() syntax structure represented by the general_constraint_info field is equal to 0.

general_profile_idc, general_tier_flag, general_level_idc, ptl_frame_only_constraint_flag, ptl_multi_layer_enabled_flag, general_constraint_info, ptl_sublayer_level_present_flag[i], sublayer_level_idc[i], ptl_num_sub_profiles and general_sub_profile_idc[j] contain the matching values for the fields or syntax structures general_profile_idc, general_tier_flag, general_level_idc, ptl_frame_only_constraint_flag, ptl_multi_layer_enabled_flag, general_constraint_info(), ptl_sublayer_level_present_flag[i], sublayer_level_idc[i], ptl_num_sub_profiles, and general_sub_profile_idc[j] as defined in ISO/IEC 23090-3 for the CVSs to which this VvcPTLRecord applies.

When sublayer_level_idc[i] is not present, its value is inferred as follows:

- The value of sublayer_level_idc[num_sublayers - 1] is inferred to be equal to general_level_idc.
- For i from num_sublayers - 2 to 0 (in decreasing order of values of i), inclusive, sublayer_level_idc[i] is inferred to be equal to sublayer_level_idc[i + 1].

NOTE The VvcPTLRecord in the decoder configuration record could correspond to an operating point that has less sublayers than indicated in the VPS or SPS. That is, num_sublayers minus 1 could be different from vps_max_sublayers_minus1 or sps_max_sublayers_minus1. When this is the case, general_level_idc in the VvcPTLRecord has matching values for sublayer_level_idc[num_sublayers - 1] in the VPS or SPS.

11.2.4.2 VVC decoder configuration record

11.2.4.2.1 Definition

This subclause specifies the decoder configuration information for ISO/IEC 23090-3 video content.

NOTE 1 The definition and semantics of the decoder configuration record are specified in the context of including the record in a sample entry of a VVC track. It is possible to use the decoder configuration record in other contexts too, such as for image items, by re-specifying parts of the definition and semantics.

The term referenced CVS is defined within the context of subclause 11.2.4 to be any CVS for which either of the following is true:

- The CVS is natively present in the samples that reference the sample entry containing this `VvcDecoderConfigurationRecord`.
- The CVS is reconstructed as specified in subclause 11.5.4 with the samples that reference the sample entry containing this `VvcDecoderConfigurationRecord` as input.

NOTE 2 When `ptl_present_flag` is equal to 1, the referenced CVS is reconstructed from both of the following:

- The samples referencing the sample entry containing this `VvcDecoderConfigurationRecord`.
- Time-aligned samples of the referenced VVC non-VCL track, if any.

The term referenced CVSs is defined within the context of subclause 11.2.4 to be a set of CVSs that consists of each referenced CVS natively present in or reconstructed from the samples that reference the sample entry containing this `VvcDecoderConfigurationRecord`.

This record contains the size of the length field used in each sample to indicate the length of its contained NAL units as well as the parameter sets, DCI, OPI, and SEI NAL units, if stored in the sample entry. This record is externally framed (its size is supplied by the structure that contains it).

When the `ptl_present_flag` is equal to zero in the decoder configuration record of a track and the track does not contain a 'vopi' sample group and all layers and sublayers specified in the 'vopi' sample group, the value of `track_in_movie` flag shall be equal to 0.

There is a set of arrays to carry initialization non-VCL NAL units. The NAL unit types are restricted to indicate DCI, OPI, VPS, SPS, PPS, prefix APS, and prefix SEI NAL units only. NAL unit types that are reserved in ISO/IEC 23090-3 and in this document may acquire a definition in the future specification, and readers should ignore arrays with reserved or unpermitted values of NAL unit type.

NOTE 3 This 'tolerant' behaviour is designed so that errors are not raised, allowing the possibility of backwards-compatible extensions to these arrays in future specifications.

NOTE 4 The NAL units carried in a sample entry are included immediately following the AUD and OPI NAL units (if any) in, or otherwise at the beginning of, the access unit reconstructed from the first sample that references the sample entry.

It is recommended that the arrays be in the order DCI, OPI, VPS, SPS, PPS, prefix APS, prefix SEI.

11.2.4.2.2 Syntax

```

aligned(8) class VvcDecoderConfigurationRecord {
    bit(5) reserved = '11111'b;
    unsigned int(2) LengthSizeMinusOne;
    unsigned int(1) ptl_present_flag;
    if (ptl_present_flag) {
        unsigned int(9) ols_idx;
        unsigned int(3) num_sublayers;
        unsigned int(2) constant_frame_rate;
        unsigned int(2) chroma_format_idc;
        unsigned int(3) bit_depth_minus8;
        bit(5) reserved = '11111'b;
        VvcPTLRecord(num_sublayers) native_ptl;
        unsigned int(16) max_picture_width;
        unsigned int(16) max_picture_height;
        unsigned int(16) avg_frame_rate;
    }
    unsigned int(8) num_of_arrays;
    for (j=0; j < num_of_arrays; j++) {
        unsigned int(1) array_completeness;
        bit(2) reserved = 0;
        unsigned int(5) NAL_unit_type;
        if (NAL_unit_type != DCI_NUT && NAL_unit_type != OPI_NUT)
            unsigned int(16) num_nalus;
        for (i=0; i < num_nalus; i++) {
            unsigned int(16) nal_unit_length;
            bit(8*nal_unit_length) nal_unit;
        }
    }
}

```

11.2.4.2.3 Semantics

LengthSizeMinusOne plus 1 indicates the length in bytes of the NALUnitLength field in the samples referencing the sample entry that contains this configuration record. For example, a size of one byte is indicated with a value of 0. The value of this field shall be one of 0, 1, or 3 corresponding to a length encoded with 1, 2, or 4 bytes, respectively. lengthSizeMinusOne shall have the same value in all configuration records for the same VVC bitstream.

ptl_present_flag equal to 1 specifies that the samples that reference the sample entry containing this VvcDecoderConfigurationRecord contain one or more CVSSs natively or through resolving 'subp' or 'recr' track references, those one or more CVSSs correspond to the operating point specified by ols_idx and num_sublayers, and all NAL units of those one or more CVSSs belong to that operating point. ptl_present_flag equal to 0 indicates that such constraints may not apply.

NOTE 1 For example, the referenced CVSSs could correspond to multiple output layer sets or samples that reference the sample entry containing this VvcDecoderConfigurationRecord could contain one or more individual layers that do not form an output layer set or individual sublayers excluding the sublayer with TemporalId equal to 0.

ols_idx specifies the output layer set index of an output layer set represented by the referenced CVSSs. The value of ols_idx may be used as the value of the TargetOlsIdx variable provided by external means or by an OPI NAL unit to the VVC decoder, as specified in ISO/IEC 23090-3, for decoding the referenced CVSSs.

num_sublayers greater than 1 indicates that the referenced CVSSs are temporally scalable and the contained number of sublayers is equal to num_sublayers. Value 1 indicates that the

referenced CVSs are not temporally scalable. Value 0 indicates that it is unknown whether the referenced CVSs are temporally scalable.

`constant_frame_rate` equal to 1 indicates that the referenced CVSs are of constant frame rate.

Value 2 indicates that the representation of each sublayer of the referenced CVSs are of constant frame rate. Value 0 indicates that the referenced CVSs may or may not be of constant frame rate. When the referenced CVSs contain multiple layers and samples are reconstructed for the operating point specified by `ols_idx` and `num_sublayers`, this gives the indication of whether the referenced CVSs of the operating point has constant access unit rate. Value 3 is reserved.

`chroma_format_idc` indicates the chroma format that applies to the referenced CVSs. The following constraints apply for `chroma_format_idc`:

- If the referenced CVSs contain one and only one layer, the value of `sps_chroma_format_idc`, as defined in ISO/IEC 23090-3, shall be the same in all SPSSs referenced by the VCL NAL units of the referenced CVSs, and the value of `chroma_format_idc` shall be equal to that value of `sps_chroma_format_idc`.
- Otherwise (the referenced CVSs contain more than one layer), the value of `vps_ols_dpb_chroma_format[MultiLayerOlsIdx[ols_idx]]`, as defined in ISO/IEC 23090-3, shall be the same for all the referenced CVSs, and the value of `chroma_format_idc` shall be equal to that value of `vps_ols_dpb_chroma_format[MultiLayerOlsIdx[ols_idx]]`.

`bit_depth_minus8` indicates the bit depth that applies to the referenced CVSs. The following constraints apply for `bit_depth_minus8`:

- If the referenced CVSs contain one and only one layer, the value of `sps_bitdepth_minus8`, as defined in ISO/IEC 23090-3, shall be the same in all SPSSs referenced by the VCL NAL units of the referenced CVSs, and the value of `bit_depth_minus8` shall be equal to that value of `sps_bitdepth_minus8`.
- Otherwise (the referenced CVSs contain more than one layer), the value of `vps_ols_dpb_bitdepth_minus8[MultiLayerOlsIdx[ols_idx]]`, as defined in ISO/IEC 23090-3, shall be the same for all the referenced CVSs, and the value of `bit_depth_minus8` shall be equal to that value of `vps_ols_dpb_bitdepth_minus8[MultiLayerOlsIdx[ols_idx]]`.

NOTE 2 If two coded video sequences differ in colour space or bit depth indications, then two different VVC sample entries are also required.

`native_ptl` specifies the profile, tier, and level of the output layer set represented by the referenced CVSs. `native_ptl` shall be present in the decoder configuration record when the samples that reference the sample entry containing this `VvcDecoderConfigurationRecord` form one or more CVSs natively or through resolving 'subp' or 'recr' track references that corresponds to an operating point specified by `ols_idx` and `num_sublayers`. The values for the syntax elements within `native_ptl` shall be valid for all parameter sets that are referenced when the referenced CVSs are decoded (referred to as "all the parameter sets" in the following sentences in this paragraph). Specifically, the following restrictions apply:

- The profile indication `general_profile_idc` shall indicate a profile to which the output layer set identified by `ols_idx` in this configuration record conforms.

NOTE 3 If different profiles are marked for different CVSs of the output layer set identified by `ols_idx` in this configuration record, then the referenced CVSs could need examination to determine which profile, if any, all the referenced CVSs conform to. If all the referenced CVSs are not examined, or the examination reveals that there is no profile to which all the referenced CVSs

conform, then the referenced CVSs are expected to be re-mapped to two or more separate configuration records in which these rules can be met.

- The tier indication `general_tier_flag` shall indicate a tier equal to or greater than the highest tier indicated in all the `profile_tier_level()` syntax structures (in all the parameter sets) to which the output layer set identified by `ols_idx` in this configuration record conforms.
- A bit at a particular bit index in `general_constraint_info` shall be equal to 0 when the bit of that same particular bit index is equal to 0 in any `general_constraints_info()` syntax structure among all the `profile_tier_level()` syntax structures in all the parameter sets referred to by the referenced CVSs.
- The level indication `general_level_idc` shall indicate a level of capability equal to or greater than the highest level in all the `profile_tier_level()` syntax structures (in all the parameter sets) to which the operating point identified by `ols_idx` and `num_sublayers` in this configuration record conforms.

`max_picture_width` indicates the maximum picture width, in units of luma samples, that applies to the referenced CVSs. The following constraints apply for `max_picture_width`:

- If the referenced CVSs contain one and only one layer, the value of `sps_pic_width_max_in_luma_samples`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units of the referenced CVSs, and the value of `max_picture_width` shall be equal to that value of `sps_pic_width_max_in_luma_samples`.
- Otherwise (the referenced CVSs contain more than one layer), the value of `vps_ols_dpb_pic_width[MultiLayerOlsIdx[ols_idx]]`, as defined in ISO/IEC 23090-3, shall be the same for all the referenced CVSs, and the value of `max_picture_width` shall be equal to that value of `vps_ols_dpb_pic_width[MultiLayerOlsIdx[ols_idx]]`.

`max_picture_height` indicates the maximum picture height, in units of luma samples, that applies to the referenced CVSs. The following constraints apply for `max_picture_height`:

- If the referenced CVSs contain one and only one layer, the value of `sps_pic_height_max_in_luma_samples`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units of the referenced CVSs, and the value of `max_picture_height` shall be equal to that value of `sps_pic_height_max_in_luma_samples`.
- Otherwise (the referenced CVSs contain more than one layer), the value of `vps_ols_dpb_pic_height[MultiLayerOlsIdx[ols_idx]]`, as defined in ISO/IEC 23090-3, shall be the same for all the referenced CVSs, and the value of `max_picture_height` shall be equal to that value of `vps_ols_dpb_pic_height[MultiLayerOlsIdx[ols_idx]]`.

NOTE 4 VVC enables picture resolution change within a CLVS. The maximum picture height and width is indicated in the SPS, whereas the actual width and height for each picture is indicated in the PPS.

`avg_frame_rate` gives the average frame rate in units of frames/(256 seconds), for the referenced CVSs. Value 0 indicates an unspecified average frame rate. When the referenced CVSs contain multiple layers and samples are reconstructed for the operating point specified by `ols_idx` and

`num_sublayers`, this gives the average access unit rate of the referenced CVSs of the operating point.

`numArrays` indicates the number of arrays of NAL units of the indicated type(s).

`array_completeness` when equal to 1 indicates that all NAL units of the given type are in the following array and none are present in the samples used for reconstructing the referenced CVSs; when equal to 0 indicates that additional NAL units of the indicated type may be present in the samples used for reconstructing the referenced CVSs; the permitted values are constrained by the sample entry name.

`NAL_unit_type` indicates the type of the NAL units in the following array (which shall be all of that type); it takes a value as defined in ISO/IEC 23090-3; it is restricted to take one of the values indicating a DCI, OPI, VPS, SPS, PPS, prefix APS or prefix SEI NAL unit.

`num_nalus` indicates the number of NAL units of the indicated type included in the configuration record for the referenced CVSs. The SEI array shall only contain SEI messages of a 'declarative' nature, that is, those that provide information about the referenced CVSs as a whole. An example of such an SEI could be a user-data SEI. When not present, the value of `numNalus` is inferred to be equal to 1.

`nal_unit_length` indicates the length in bytes of the NAL unit.

`nal_unit` contains a DCI, OPI, VPS, SPS, PPS, APS or declarative SEI NAL unit, as specified in ISO/IEC 23090-3.

11.2.4.3 VVC configuration box

11.2.4.3.1 Definition

This subclause specifies `VvcConfigurationBox` that carries a VVC decoder configuration record.

This box derives from `FullBox` and hence contains a `version` field. This version of the specification defines version 0 of this box. Incompatible changes to the box will be indicated by a change of `version` number. Readers shall not attempt to decode this box or the referenced CVSs if the `version` number is unrecognized.

Compatible extensions to this box will extend it and need not change the `version` code. Readers should be prepared to ignore unrecognized data beyond the definition of the data they understand.

11.2.4.3.2 Syntax

```
class VvcConfigurationBox extends FullBox('vvcC', version=0, flags) {
    VvcDecoderConfigurationRecord() VvcConfig;
}
```

11.2.4.3.3 Semantics

`VvcDecoderConfigurationRecord` is defined in subclause 11.2.4.2.

The following semantics apply when `VvcConfigurationBox` is contained in a sample entry of type 'vvc1' or 'vv11':

(`flags & 1`) equal to 1 in a `VvcConfigurationBox` included in a VVC subpicture track indicates that the samples referencing the sample entry do not contain content intended for displaying.

`LengthSizeMinusOne` plus 1 indicates the length in bytes of the `NALUnitLength` field in a track that contains the `VvcNALUConfigBox`. For example, a size of one byte is indicated with a value of 0. The value of this field shall be one of 0, 1, or 3 corresponding to a length encoded with 1, 2, or 4 bytes, respectively. `lengthSizeMinusOne` shall have the same value in all instances of `VvcConfigurationBox` and `VvcNALUConfigBox` contained in the sample entries referenced by the samples used for reconstructing a VVC bitstream as specified in subclause 11.5.4.

11.2.4.4 VVC NAL unit configuration box

11.2.4.4.1 Definition

This subclause specifies `VvcNALUConfigBox` that carries the `LengthSizeMinusOne` field used for indicating the byte count of the `NalUnitLengthField` that delimits NAL units. `VvcNALUConfigBox` is intended to be used in container structures where an entire `VvcConfigurationBox` is unnecessary.

11.2.4.4.2 Syntax

```
class VvcNALUConfigBox extends FullBox('vvnC', version=0, flags) {
    unsigned int(6) reserved=0;
    unsigned int(2) LengthSizeMinusOne;
}
```

11.2.4.4.3 Semantics

(`flags & 1`) equal to 1 in a `VvcNALUConfigBox` included in a '`vvsl`' sample entry indicates that the samples referencing the sample entry do not contain content intended for displaying.

`LengthSizeMinusOne` plus 1 indicates the length in bytes of the `NALUnitLength` field used for indicating the byte count of the `NalUnitLengthField` that delimits NAL units. The value of this field shall be one of 0, 1, or 3 corresponding to `NalUnitLengthField` of 1, 2, or 4 bytes, respectively. `lengthSizeMinusOne` shall have the same value in all instances of `VvcConfigurationBox` and `VvcNALUConfigBox` contained in the sample entries referenced by the samples used for reconstructing a VVC bitstream as specified in subclause 11.5.4.

11.3 Derivation from ISO base media file format

11.3.1 VVC sample entries

11.3.1.1 Definition

Sample Entry Types: '`vvcl`', '`vvil`'

Container: Sample Table Box ('`stbl`')

Mandatory: An '`vvcl`' or '`vvil`' sample entry is mandatory in at least one track among the tracks carrying a VVC bitstream

Quantity: One or more sample entries may be present

A VVC sample entry is defined to be a sample entry that has sample entry type equal to '`vvcl`' or '`vvil`'. Each sample entry of a VVC track shall be a VVC sample entry. A VVC sample entry shall contain a `VvcConfigurationBox`, which includes a `VvcDecoderConfigurationRecord`.

When the `BitRateBox` is present in a VVC sample entry with `ptl_present_flag` equal to 1 in its decoder configuration record, it signals the bit rate information of the sequence of consecutive CVSs associated with the VVC sample entry.

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.

When the sample entry name is 'vvc1' or 'vvi1' and profile, tier, and level information is present in the sample entry, the contiguous CVSs that the sample entry references shall form a compliant VVC bitstream as viewed by a VVC decoder operating under the configuration (including profile, tier, and level) given in the `VvcConfigurationBox`.

When the sample entry name is 'vvc1', the following applies:

- The value of `array_completeness` shall be equal to 1 for arrays of SPS, and PPS NAL units.
- If a VVC bitstream includes DCI NAL unit(s), the value of `array_completeness` shall be equal to 1 for the array of DCI units. Otherwise, `NAL_unit_type` shall not indicate DCI NAL units.
- If a VVC bitstream includes VPS NAL unit(s), the value of `array_completeness` shall be equal to 1 for the array of VPS NAL units. Otherwise, `NAL_unit_type` shall not indicate VPS NAL units.

When the value of `array_completeness` is equal to 1 for an array of a particular `NAL_unit_type` value, NAL units of that `NAL_unit_type` value cannot be updated without causing a different sample entry to be used.

When the sample entry name is 'vvi1', the value of `array_completeness` of at least one of the following arrays shall be equal to 0:

- The array of DCI NAL units, if present
- The array of VPS NAL units, if present
- The array of SPS NAL units
- The array of PPS NAL units

NOTE 1 Systems wishing to send parameter set updates will need to compare the constraints of 'vvc1' and 'vvi1' sample entry types to find the differences in order to send parameter set updates appropriately. For VPSs, SPSs, and PPSs, it is suggested that when parameter set updating is desired, the parameter sets are included in the samples. For APSS, it is suggested that when parameter set updating is desired, the APSS are included in the VVC non-VCL track.

NOTE 2 Decoders conforming to this specification are required to support both parameter sets stored in the samples as well as parameter sets stored in the sample entries, unless restricted by another specification using this one.

File readers shall discard the NAL units or NAL-unit-like structures having `nal_unit_type` in the range of UNSPEC_28 to UNSPEC_31, inclusive, from a VVC bitstream reconstructed using any sample of 'vvc1' or 'vvil' sample entries. The canonical order and restrictions of a VVC bitstream as defined in subclause 11.2.3 shall apply after the removal of NAL units or NAL-unit-like structures specified above.

11.3.1.2 Syntax

```
class VvcSampleEntry() extends VisualSampleEntry ('vvc1' or 'vvil') {
    VvcConfigurationBox config;
    MPEG4ExtensionDescriptorsBox (); // optional
}
```

11.3.1.3 Semantics

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\012VVC Coding" being recommended (\012 is 10, the length of the string in bytes).

11.3.2 VVC subpicture sample entry 'vvs1'

11.3.2.1 Definition

Sample Entry Types: 'vvs1'

Container: Sample Table Box ('stbl')

Mandatory: No

Quantity: One or more sample entries may be present

When a VVC subpicture track does not contain a conforming VVC bitstream, a 'vvs1' sample entry is used for the VVC subpicture track, and the following constraints apply for the track:

- The `track_in_movie` flag shall be equal to 0.
- The track shall contain one and only one sample entry.
- The track shall be referenced by at least one VVC merge base track through the 'subp' track reference.
- DCI, OPI, VPS, SPS, PPS, AUD, PH, EOS, EOB and other AU- or picture-level non-VCL NAL units shall be absent both in the sample entry and in the samples of 'vvs1' tracks.
- Unless indicated otherwise, child boxes of video sample entries (such as CleanApertureBox and PixelAspectRatioBox) should not be present in the sample entry and, if present, shall be ignored.
- The composition time offset information for samples of a 'vvs1' track shall not be present.
- Subsample information for samples of a 'vvs1' track may be present; if present, the subsample information shall follow the definition of sub-samples for VVC.

The sample entry of type 'vvs1' shall contain VvcNALUConfigBox.

File readers shall discard the NAL units or NAL-unit-like structures having `nal_unit_type` in the range of UNSPEC_28 to UNSPEC_31, inclusive, from a VVC bitstream reconstructed using any sample of 'vvs1' sample entry. The canonical order and restrictions of a VVC bitstream as defined in subclause 11.2.3 shall apply after the removal of NAL units or NAL unit like structures specified above.

11.3.2.2 Syntax

```
class VvcSubpicSampleEntry() extends VisualSampleEntry ('vvs1') {
    VvcNALUConfigBox config;
}
```

11.3.2.3 Semantics

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\012VVC Coding" being recommended (\012 is 10, the length of the string in bytes).

11.3.3 VVC non-VCL sample entry

11.3.3.1 Sample entry name and format

11.3.3.1.1 Definition

Sample Entry Type: 'vvcN'
 Container: Sample Description Box ('std')
 Mandatory: Yes
 Quantity: One or more sample entries may be present

A VVC non-VCL sample entry shall contain a VvcNALUConfigBox.

When a 'vvcN' sample entry is present in a track, the track_in_movie flag shall be equal to 0 for the track.

11.3.3.1.2 Syntax

```
class VvcNonVCLSampleEntry() extends VisualSampleEntry ('vvcN') {
    VvcNALUConfigBox config;
}
```

11.3.3.1.3 Semantics

Compressorname in the base class VisualSampleEntry indicates the name of the compressor used with the value "\013VVC non-VCL" being recommended (\013 is 11, the length of the string as a byte).

11.3.3.2 Track reference

A track reference of type 'vvcN' may be included in a VVC track or a VVC subpicture track, referencing a VVC non-VCL track. When present, this track reference is used to connect from the VVC track or the VVC subpicture track to the non-VCL track.

When either a VVC track or a VVC subpicture track has a track reference of type 'vvcN' to a VVC non-VCL track (hereafter called the associated VVC non-VCL track), the following applies:

- A sample in the video track that is not a sync sample may or may not have a corresponding sample in the associated VVC non-VCL track having the same decoding time. For each sync sample in the video track, if there is a corresponding sample in the associated VVC non-VCL track having the same decoding time, then that sample in the associated VVC non-VCL track shall also be a sync sample.

- For each sample in the associated VVC non-VCL track, there shall be one and only one sample in the video track having the same decoding time.

11.3.4 Constraints related to VVC merge base tracks, VVC extraction base tracks and VVC subpicture tracks

When a VVC subpicture track contains a conforming VVC bitstream, which can be consumed without other VVC subpicture tracks, a regular VVC sample entry is used ('vvc1' or 'vvi1') for the VVC subpicture track.

A VVC track may contain a 'subp' track reference. The entries of a 'subp' track reference shall contain either a `track_ID` value of a VVC subpicture track or a `track_group_id` value of an 'alte' track group of VVC subpicture tracks.

A sample group of type 'spor', as specified in subclause 11.4.17, shall be present in each VVC merge base track.

A VVC track may contain a 'recl' track reference. When a 'recl' track reference is present, it shall contain one and only one entry. The entry of a 'recl' track reference shall contain a `track_ID` value of a VVC track that is neither a VVC merge base track nor a VVC extraction base track. A track referenced by a 'recl' track reference shall not have any associated VVC non-VCL track. A VVC extraction base track shall not have any associated VVC non-VCL track.

A sample group of type 'rrol', as specified in subclause 11.4.20, shall be present in each VVC extraction base track.

A VVC track shall not contain both 'subp' track reference and 'recl' track reference.

A VVC merge base track may contain a 'mixn' track reference, with entries containing any of the following:

- a `track_ID` value of a VVC subpicture track,
- a `track_group_id` value of an 'alte' track group of VVC subpicture tracks, or
- a `track_group_id` value of a 'snut' track group of VVC subpicture tracks.

When a VVC subpicture track is referenced by a VVC merge base track containing a 'spor' sample group description entry having `subpic_id_info_flag` equal to 1, the VVC subpicture track shall contain a subpicture ID sample group description, potentially using the default sample group description index.

When the VVC track is referenced by a VVC extraction base track containing a 'rrol' sample group description entry having `subpic_id_info_flag` equal to 1, the VVC track shall contain a subpicture ID sample group description with `rect_region_flag` equal to 1, potentially using the default sample group description index.

When the VVC track is referenced by a VVC extraction base track, the VVC track shall contain a 'nalm' sample group that should be associated with `grouping_type_parameter` equal to 'trif'.

NOTE In this version of this document, a 'trif' sample group is the only sample group that is suitable to be used as the `grouping_type_parameter` for a 'nalm' sample group. However, readers are expected to handle any value `grouping_type_parameter` for a 'nalm' sample group when resolving `groupID` values contained in 'rrof' sample group description entries.

When a VVC track is referenced by a VVC extraction base track, the DCI, OPI, VPS, SPS, PPS, AUD, PH, EOS, and EOB NAL units present in a sample of the VVC track shall be mapped to `groupID` equal to 0 in the `NALUMapEntry` applying to the sample of the VVC track.

A 'snut' track group shall contain the VVC subpicture tracks, and each time-aligned set of samples of all the VVC subpicture tracks in the 'snut' track group shall contain the subpicture(s) with the same VCL NAL unit type.

The VVC subpicture tracks in the same 'snut' track group are necessarily independent of any other VVC subpicture tracks referenced by the same VVC merge base track to avoid decoding mismatches and could therefore be constrained as follows:

- All the VVC subpicture tracks contain VVC subpictures.
- The subpicture boundaries are like picture boundaries.

When a 'subp' track reference in a VVC merge base track references an 'alte' track group that collects VVC subpicture tracks with different VCL NAL unit types in any time-aligned sample, either of the following shall be true:

- All the VVC subpicture tracks that are referenced by the 'subp' track reference are grouped into 'snut' track groups, where each 'snut' track group contains the VVC subpicture tracks with the same VCL NAL unit type in time-aligned samples. The VVC merge base track contains a track reference of type 'mixn' where the entries are the `track_group_id` values of the 'snut' track groups.
- The VVC merge base track contains a track reference of type 'mixn' where the entries are the `track_id` values of all the VVC subpicture tracks with different VCL NAL unit types in any time-aligned sample in the respective 'alte' track group.

11.3.5 Sync sample

A sync sample in 'vvc1' and 'vvil' tracks shall contain VCL NAL units indicating that the coded picture in the sample is an Instantaneous Decoding Refresh (IDR) picture, a Clean Random Access (CRA) picture, a Gradual Decoding Refresh (GDR) picture, or a Step-wise Temporal Sublayer Access (STSA) picture. When the coded picture in a sync sample is a CRA picture, there shall be no RASL pictures associated with that CRA picture. When the coded picture in a sync sample is a GDR picture, it shall have `ph_recovery_poc_cnt` equal to 0 and there shall be no pictures following it in decoding order that precede it in presentation order. When the coded picture in a sync sample is an STSA picture, its `TemporalId` shall be equal to the lowest `TemporalId` in the track and there shall be no pictures following it in

decoding order that precede it in presentation order and the first sample that follows it in decoding order in each sublayer with TemporalId greater than tIdSync shall be of type STSA.

A sync sample in 'vvc1' and 'vvil' track carrying multiple layers of a VVC bitstream shall contain VCL NAL units indicating that the coded pictures in the sample are Instantaneous Decoding Refresh (IDR) pictures, Clean Random Access (CRA) pictures, Gradual Decoding Refresh (GDR) pictures or Step-wise Temporal Sublayer Access (STSA) pictures. The following conditions apply for the coded pictures in the sample of a track carrying multiple layers of a VVC bitstream.

- When the coded pictures in a sync sample are CRA pictures, there shall be no RASL pictures associated with the corresponding CRA pictures.
- When the coded pictures in a sync sample are GDR pictures, they shall have ph_recovery_poc_cnt equal to 0 and there shall be no pictures following it in decoding order that precede it in presentation order.
- When the coded pictures in a sync sample are STSA pictures, their TemporalId tIdSync shall be equal to the lowest TemporalId in the track and there shall be no pictures following the sync sample in decoding order that precede it in presentation order and the first sample that follows it in decoding order in each sublayer with TemporalId greater than tIdSync shall be of type STSA.

NOTE 1 When there is no sync sample among the samples described by the TrackBox, the SyncSampleBox is present and has entry_count equal to 0. Storage of a VVC bitstream might not result into any sync samples being present, for example when either of the following is true:

- A VVC bitstream does not contain IDR or CRA pictures and starts with a GDR picture with ph_recovery_poc_cnt greater than 0.
- A VVC bitstream does not contain IDR pictures and contains some CRA pictures but each of them has at least one associated RASL picture.

NOTE 2 A VVC bitstream stored under multiple tracks might not have time-aligned sync samples in different tracks. In such cases sync samples at an earlier time might be needed for accessing all the layers and sublayers of an operating point. Decoders might need sync samples to be time-aligned in all the layers of an operating point to perform random accessing.

Table 13 indicates the mapping between VVC VCL NAL unit types, ISOBMFF sync sample status and SAP types as documented in ISOBMFF for a VVC track of type 'vvc1' and 'vvil'.

Table 13 — Mapping of VVC VCL NAL unit types to sync sample status and SAP types for a VVC track of type 'vvc1' and 'vvi1'

NAL Unit Type	DASH SAP type	ISOBMFF sync sample status
IDR_N_LP	1	true
IDR_W_RADL	2 (if IRAP picture(s) has/have associated RADL pictures)	true
	1 (if IRAP picture(s) has/have no associated RADL pictures)	true
CRA	3 (if IRAP picture(s) has/have associated RASL pictures)	false
	2 (if IRAP picture(s) has/have no associated RASL pictures but has/have associated RADL pictures)	true
	1 (if IRAP picture(s) has/have no associated leading pictures)	true
GDR	4 (if GDR picture(s) has/have ph_recovery_poc_cnt greater than 0)	false
	3 (if GDR picture(s) has/have ph_recovery_poc_cnt equal to 0 and GDR picture(s) has/have pictures following it in decoding order that precede it in presentation order)	false
	1 (if GDR(s) has/have ph_recovery_poc_cnt equal to 0 and GDR picture(s) does/do not have pictures following in decoding order that precedes it in presentation order)	true
STSA	3 (if the picture(s) in the first AU of each further sublayer present in the track that follows the STSA picture(s) in presentation order is/are also STSA picture(s). There might be pictures in each further sublayer present in the track that follow the STSA picture(s) in decoding order but precede it in presentation order that are not STSA pictures)	false
	2 (if the picture(s) in the first AU of each further sublayer present in the track that follows the STSA picture(s) in decoding order is/are also STSA picture(s). There might be pictures following the STSA picture(s) in decoding order that precede it in presentation order)	true, if TemporalId equal to smallest TemporalId in track and the first following sample of each further sublayer in decoding order are of type STSA
	1 (if the picture(s) in the first picture of each further sublayer present in the track that follows the STSA picture(s) is/are also STSA picture(s). There are no pictures preceding the STSA picture(s) in presentation order)	true, if TemporalId equal to smallest TemporalId in track and the first following sample of each further sublayer in decoding order are of type STSA

A sync sample in 'vvs1' VVC subpicture tracks shall contain one or more subpictures, and each of these subpictures shall be an Instantaneous Decoding Refresh (IDR) subpicture, a Clean Random Access (CRA) subpicture, a Gradual Decoding Refresh (GDR) subpicture, or a Step-wise Temporal Sublayer Access (STSA) subpicture, being constrained as follows.

- When a coded subpicture in a sync sample is a CRA subpicture, there shall be no RASL subpictures associated with that CRA subpicture.
- When a coded subpicture in a sync sample is a GDR subpicture, the time-aligned sample in each VVC merge base track that references this VVC subpicture track shall have `ph_recovery_poc_cnt` equal to 0 and there shall be no subpictures following it in decoding order that precede it in presentation order.
- When a coded subpicture in a sync sample is an STSA subpicture, its `TemporalId` `tIdSync` shall be equal to the lowest `TemporalId` in the track and there shall be no subpictures following it in decoding order that precede it in presentation order and the first sample that follows it in decoding order in each sublayer with `TemporalId` greater than `tIdSync` shall be of type STSA.

Table 14 indicates the mapping between VVC VCL NAL unit types, ISOBMFF sync sample status and SAP types as documented in ISOBMFF for a VVC subpicture track of type 'vvs1'.

Table 14 — Mapping of VVC VCL NAL unit types to sync sample status and SAP types for 'vvs1' VVC subpicture track

NAL Type	Unit	DASH SAP type	ISOBMFF sync sample status
IDR_N_LP	1		true
IDR_W_RADL	2 (if IRAP subpicture(s) has/have associated RADL subpicture(s))		true
	1 (if IRAP subpicture(s) has/have no associated RADL subpicture(s))		true
CRA	3 (if IRAP subpicture(s) has/have associated RASL subpicture(s))		false
	2 (if IRAP subpicture(s) has/have no associated RASL subpicture(s) but has/have associated RADL subpicture(s))		true
	1 (if IRAP subpicture(s) has/have no associated leading subpicture(s))		true
GDR	4 (if the sample consists of GDR subpicture(s) and the time-aligned sample in each VVC merge base track that references this VVC subpicture track has <code>ph_recovery_poc_cnt</code> greater than 0)		false
	3 (if this sample consists of GDR subpicture(s) and the time-aligned sample in each VVC merge base track that references this VVC subpicture track has <code>ph_recovery_poc_cnt</code> equal to 0)		false

	and the GDR subpicture(s) has/have subpicture(s) following it in decoding order that precede it in presentation order)	
	1 (if this sample consists of GDR subpicture(s) and the time-aligned sample in each VVC merge base track that references this VVC subpicture track has/have ph_recovery_poc_cnt equal to 0 and the GDR subpicture(s) does/do not have subpicture(s) following in decoding order that precedes it in presentation order)	true
STSA	3 (if the subpicture(s) in the first AU of each further sublayer present in the track that follows the STSA subpicture(s) in presentation order is/are also STSA subpicture(s). There might be subpicture(s) in each further sublayer present in the track that follow the STSA subpicture(s) in decoding order but precede it in presentation order that are not STSA subpicture(s))	false
	2 (if the subpicture(s) in the first AU of each further sublayer present in the track that follows the STSA subpicture(s) in decoding order is/are also STSA subpicture(s). There might be subpicture(s) following the STSA subpicture(s) in decoding order that precede it in presentation order)	true, if TemporalId equal to smallest TemporalId in track and the first following sample of each further sublayer in decoding order are of type STSA
	1 (if the subpicture(s) in the first subpicture of each further sublayer present in the track that follows the STSA subpicture(s) is/are also STSA subpicture(s). There are no subpictures preceding the STSA subpicture(s) in presentation order)	true, if TemporalId equal to smallest TemporalId in track and the first following sample of each further sublayer in decoding order are of type STSA

The following requirement applies for each bitstream reconstructed as specified in subclause 11.5.4 with a VVC non-VCL track as one of the input tracks:

- All parameter sets needed for decoding shall be present in the bitstream starting from the decoding time of a sync sample in the VVC non-VCL track.

When the sample entry name is 'vvc1' or 'vvi1', and the track has a track reference of type 'vvcN', the following requirement applies for each bitstream reconstructed as specified in subclause 11.5.4:

- All parameter sets needed for decoding shall be present in the bitstream starting from the decoding time of the sync sample in the 'vvc1' or 'vvi1' track.

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.

When the sample entry name is 'vvi1' and the track does not have a track reference of type 'vvcN', the following applies:

- If the sample is a sync sample, all parameter sets 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 parameter sets 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.

When the sample entry type is 'vvs1', the following applies:

- If the sample is a sync sample and there is a 'vvi1' VVC merge base track referencing this 'vvs1' VVC subpicture track, all parameter sets needed for decoding the sample shall be present in the access unit reconstructed using the sync sample as input as specified in subclause 11.5.4.

NOTE 3 Parameter sets can be repeated in any sample in the VVC merge base track. Consequently, the VVC merge base track can, but does not need to, have a sync sample aligned in decoding time with a sync sample of a 'vvs1' track.

- Otherwise, if the sample is a sync sample and there is a 'vvc1' VVC merge base track referencing this 'vvs1' VVC subpicture track, all APSs needed for decoding the sample shall be present in the access unit reconstructed using the sync sample as input as specified in subclause 11.5.4.

NOTE 4 When VVC subpicture tracks are in use, it is not necessary to store all parameter sets in the same track. For example, an SPS and a PPS could be stored in a VVC merge base track, while an APS could be stored in a 'vvs1' subpicture track or in a VVC non-VCL track associated with one or more 'vvs1' subpicture tracks.

The sync sample table (and the equivalent flag in movie fragments) shall be used in a VVC track unless all samples are sync samples. Note that track fragment random access box refers to the presence of signalled sync samples in a movie fragment.

NOTE 5 A file writer could need to duplicate APS NAL units from previous samples to provide APSs needed for decoding a sync sample if a particular value of `aps_adaptation_parameter_set_id` and a particular value of `aps_params_type` are not present in the original bitstream for the AU marked as sync sample. When doing so, a file writer might need to rewrite the `nal_unit_type` value from `SUFFIX_APS_NUT` to `PREFIX_APS_NUT`.

NOTE 6 A file reader can detect repeated APS NAL units by matching an entire APS NAL unit to a previously decoded APS NAL unit. A file reader can remove repeated APS units before passing the bitstream on e.g. to a decoder, but this is not required as the bitstream is conformant with them present.

11.3.6 Definition of a sub-sample for VVC

For the use of the `SubSampleInformationBox` (specified in ISO/IEC 14496-12) in a VVC track, a VVC subpicture track or a VVC non-VCL track, a sub-sample is defined on the basis of the value of the `flags` field of the sub-sample information box as specified in this subclause. The presence of this box is optional; however, if present in a track containing VVC data, the '`codec_specific_parameters`' field in the box shall have the semantics defined here.

`flags` specifies the type of sub-sample information given in this box as follows:

- 0: NAL-unit-based sub-samples. A sub-sample contains one or more contiguous NAL units.
- 1: Tile-based sub-samples. A sub-sample contains one tile and the associated non-VCL NAL units, if any, of the VCL NAL unit(s) containing the tile.
- 2: CTU-row-based sub-samples. A sub-sample either contains one CTU row within a tile and the associated non-VCL NAL units, if any, of the VCL NAL unit(s) containing the CTU row. This type of sub-sample information shall not be used when `sps_entropy_coding_sync_enabled_flag` is equal to 0.
- 3: Slice-based sub-samples. A sub-sample contains one slice (i.e., one VCL NAL unit) and the associated non-VCL NAL units, if any.
- 4: Subpicture-based sub-samples. A sub-sample contains one coded subpicture and the associated non-VCL NAL units, if any, of the VCL NAL unit(s) containing the subpicture.
- 5: Picture-based sub-samples. A sub-sample contains one coded picture and the associated non-VCL NAL units.

Other values of `flags` are reserved.

The `subsample_priority` field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12.

The `discardable` field shall be set to 1 only if this sample is still decodable if this sub-sample is discarded (e.g. the sub-sample consists of an SEI NAL unit).

When the first byte of a NAL unit is included in a sub-sample, the preceding length field shall also be included in the same sub-sample.

The `codec_specific_parameters` field of the `SubSampleInformationBox` is defined for VVC as follows:

```

if (flags == 0) {
    unsigned int(1) rap_nal_unit_flag;
    unsigned int(1) gdr_nal_unit_flag;
    unsigned int(1) vcl_nal_unit_flag;
    bit(29) reserved = 0;
} else if (flags == 1) {
    unsigned int(2) vcl_idc;
    bit(2) reserved = 0;
    unsigned int(4) log2_min_luma_ctb;
    unsigned int(12) ctb_x;
    unsigned int(12) ctb_y;
} else if (flags == 2 || flags == 3) {
    unsigned int(2) vcl_idc;
    bit(30) reserved = 0;
} else if (flags == 4) {
    unsigned int(2) vcl_idc;
    unsigned int(5) vcl_nalu_type;
    unsigned int(2) log2_min_luma_ctb;
    unsigned int(8) ctb_x;
    unsigned int(8) ctb_y;
    unsigned int(1) discardable_flag;
    unsigned int(1) no_inter_layer_pred_flag;
    unsigned int(1) treated_as_picture_flag;
    unsigned int(1) loop_filter_across_enabled_flag;
    bit(3) reserved = 0;
} else if (flags == 5) {
    unsigned int(1) discardable_flag;
    unsigned int(1) no_inter_layer_pred_flag;
    unsigned int(1) all_intra_slices_flag;
    bit(29) reserved = 0;
}

```

rap_nal_unit_flag equal to 0 indicates that none of the NAL units in the sub-sample has nal_unit_type equal to IDR_W_RADL, IDR_N_LP, or CRA_NUT, or has nal_unit_type equal to GDR_NUT with associated ph_recovery_poc_cnt equal to 0, as specified in ISO/IEC 23090-3. Value 1 indicates that all NAL units in the sub-sample have nal_unit_type equal to IDR_W_RADL, IDR_N_LP, or CRA_NUT, or have nal_unit_type equal to GDR_NUT with associated ph_recovery_poc_cnt equal to 0, as specified in ISO/IEC 23090-3.

gdr_nal_unit_flag equal to 0 indicates that none of the NAL units in the sub-sample has nal_unit_type equal to GDR_NUT as specified in ISO/IEC 23090-3. Value 1 indicates that all NAL units in the sub-sample have nal_unit_type equal to GDR_NUT as specified in ISO/IEC 23090-3.

vcl_nal_unit_flag equal to 0 indicates that all NAL units in the sub-sample are non-VCL NAL units. Value 1 indicates that all NAL units in the sub-sample are VCL NAL units.

vcl_idc indicates whether the sub-sample contains Video Coding Layer (VCL) data, non-VCL data, or both, as follows:

0: the sub-sample contains VCL data and does not contain non-VCL data

1: the sub-sample contains no VCL data and contains non-VCL data

2: the sub-sample may contain both VCL and non-VCL data, which shall be associated with each other. For example, a sub-sample may contain a decoding unit information SEI message followed by the set of NAL units associated with the SEI message.

3: reserved

vcl_nalu_type specifies the VCL nal_unit_type of the sub-sample as specified in ISO/IEC 23090-3.

log2_min_luma_ctb indicates the unit of ctb_x and ctb_y, specified as follows:

0: 32 luma samples

- 1: 64 luma samples
- 2: 128 luma samples
- 3: reserved

`ctb_x` specifies the 0-based coordinate of the right-most luma samples of the tile associated with the sub-sample when `flags` is equal to 2 and `vcl_idc` is equal to 1 or 2, in units derived from `log2_min_luma_ctb` as specified above.

`ctb_y` specifies the 0-based coordinate the bottom-most luma samples of the tile associated with the sub-sample when `flags` is equal to 2 and `vcl_idc` is equal to 1 or 2, in units derived from `log2_min_luma_ctb` as specified above.

`discardable_flag` indicates the `ph_non_ref_pic_flag` value of the VCL NAL units in the sub-sample. All the VCL NAL units in the sub-sample shall have the same `ph_non_ref_pic_flag` value.

NOTE This is not the same definition as the `discardable` field in the `SubSampleInformationBox`.

`no_inter_layer_pred_flag` equal to 1 indicates that the current subpicture or picture has no inter-layer reference pictures as active entries in reference picture lists 0 and 1. `no_inter_layer_pred_flag` shall be equal to 1 for an independent layer. `no_inter_layer_pred_flag` equal to 0 indicates that the current subpicture or picture may have inter-layer reference pictures as active entries in reference picture list 0 or 1.

`treated_as_picture_flag` equal to 1 indicates that the current subpicture is treated as a picture in the decoding process excluding in-loop filtering operations. `treated_as_picture_flag` equal to 0 indicates that the current subpicture is not treated as a picture in the decoding process excluding in-loop filtering operations. When the subpicture is a picture (i.e. there is one subpicture in a picture), `treated_as_picture_flag` shall be equal to 1.

`loop_filter_across_enabled_flag` equal to 1 indicates that in-loop filtering operations across subpicture boundaries are enabled and might be performed across the boundaries of the current subpicture. `loop_filter_across_enabled_flag` equal to 0 indicates that in-loop filtering operations across subpicture boundaries are disabled and are not performed across the boundaries of the current subpicture. When the subpicture is a picture (i.e. there is one subpicture in a picture), `loop_filter_across_enabled_flag` shall be equal to 0.

`all_intra_slices_flag` equal to 1 indicates that all coded slices of the current picture have `sh_slice_type` equal to 2. `all_intra_slices_flag` equal to 0 specifies that there might or might not be one or more coded slices in the current picture that have `sh_slice_type` equal to 0 or 1.

11.3.7 Handling non-output samples

VVC allows for file format samples that are used only for reference and not output (e.g. a non-displayed reference picture in video). When any such non-output sample is present in a track, the specifications of ISO/IEC 14496-12:2020, subclause 8.6.1.1 apply

The definition on handling non-output samples specified above is not always applicable when multiple layers are involved. If what is specified above cannot be followed, then the composition time of a sample shall be set as if the sample were an output sample (i.e., at least one picture of the sample is output). If the decoding of a sample results into no output pictures, the

presentation of the sample is omitted and the duration of the previous sample is extended, so that the following samples have correct composition timing.

11.4 Sample groups

11.4.1 Common layer_id_method_idc semantics

The syntax and semantics of `grouping_type_parameter` for the stream access point sample group is specified in ISO/IEC 14496-12:2020, subclause 10.6.1. The syntax of `grouping_type_parameter` consists of two syntax elements, namely 28-bit `target_layers` and 4-bit `layer_id_method_idc`. According to ISO/IEC 14496-12 `layer_id_method_idc` equal to 0 specifies that the target layers consist of all the layers represented by the track. Furthermore, ISO/IEC 14496-12 specifies that any `layer_id_method_idc` value not equal to 0 is specified by derived media format specifications.

This clause specifies that the `grouping_type_parameter` as specified in ISO/IEC 14496-12:2020, subclause 10.6.1 shall be used for the following sample groups, when present:

- the stream access point sample group ('sap')
- the random access recovery point sample group ('roll')
- the step-wise sublayer access sample group ('stsa')

This clause specifies the semantics of the following `layer_id_method_idc` values:

- `layer_id_method_idc` equal to 0 or 2 specifies that the target layers consist of all the layers represented by the track.
- The semantics of `layer_id_method_idc` equal to 1 and 3 are specified be the same as the semantics of `layer_id_method_idc` equal to 1 in subclause 9.5.7.

In addition, the following semantics are specified for the 'sap' sample group with `SAP_type` equal to 4 and for the 'roll' sample group:

- `layer_id_method_idc` equal to 0 specifies that the sample contains a GDR picture in all the layers represented by the track.
- `layer_id_method_idc` equal to 1 specifies that the sample contains a GDR picture in the target layers indicated through `target_layers`.
- `layer_id_method_idc` equal to 2 or 3 for a sample in a VVC track specifies that the following constraints are true for all the layers represented by the track or the target layers indicated through `target_layers`, respectively:
 - The sample references a PPS with `pps_mixed_nalu_types_in_pic_flag` equal to 1.

- For each subpicture index i in the range of 0 to $\text{sps_num_subpics_minus1}$, inclusive, all the following conditions are true:
 - $\text{sps_subpic_treated_as_pic_flag}[i]$ is equal to 1;
 - there is at least one IRAP subpicture in the current sample;
 - for any subpicture i in the current sample that is not an IRAP subpicture, there is at least one IRAP subpicture in the sample in the same CLVS that follows the current sample in decoding order.

NOTE 1 VVC enables subpictures with different VCL NAL unit types within the same coded picture. A gradual decoding refresh is obtained by updating subpictures of each subpicture index as an IRAP subpicture within a range of pictures. However, VVC does not specify a decoding process that would start from a picture having mixed VCL NAL unit types.

In addition, when the 'roll' sample group is used with $\text{layer_id_method_idc}$ equal to 2 or 3, each sample mapped to a 'roll' sample group shall be mapped to a sample group description entry with a roll_distance value that is correct for a decoding process that omits decoding of subpictures with a particular subpicture index before there is an IRAP subpicture.

Let an initiating sample be defined for this subclause as a sample marked with at least one of the following:

- the 'sap' sample group with SAP_type equal to 4 and $\text{layer_id_method_idc}$ equal to 2 or 3;
- the 'roll' sample group with $\text{layer_id_method_idc}$ equal to 2 or 3.

NOTE 2 The AU reconstructed from an initiating sample can be contained within a track natively or through resolving 'subp' or 'recl' track references (when present), and through resolving 'vvcN' track references (when present).

When a reader uses an initiating sample for starting the decoding, either of the following approaches is possible:

- a) A non-standard VVC decoding process is used, where the decoding is started from the initiating sample containing mixed VAL NAL unit types and the decoding of subpictures at a particular subpicture location is omitted until the first IRAP subpicture, in decoding order, at that subpicture location.
- b) A standard VVC decoding process is used for the bitstream reconstructed according to subclause 11.5.4 and further modified as follows so that the bitstream starting with the initiating sample is a conforming bitstream:
 - 1) When the AU reconstructed from initiating sample has an AUD NAL unit, the AUD NAL unit has $\text{aud_irap_or_gdr_flag}$ equal to 1. When the AU reconstructed from initiating sample has no AUD NAL unit and the value of the $\text{vps_max_layers_minus1}$ is greater than 0 in the VPS referred to by the initiating sample (if any), an AUD NAL unit is inserted with $\text{aud_irap_or_gdr_flag}$ equal to 1.

- 2) Any SPS referred to by the initiating sample has `sps_gdr_enabled_flag` equal to 1.
- 3) Any PPS referred to by the initiating sample has `pps_mixed_nalu_types_in_pic_flag` equal to 0.
- 4) All VCL NAL units of the AU reconstructed from the initiating sample have `nal_unit_type` equal to GDR_NUT.
- 5) Any picture header of the AU reconstructed from the initiating sample has `ph_gdr_pic_flag` equal to 1 and a value of `ph_recovery_poc_cnt` corresponding to the `roll_distance` of the 'roll' sample group description entry to which the sample is mapped.

Further sample group specific constraints on `layer_id_method_idc` are specified in the subsequent subclauses.

11.4.2 Stream access point sample group

The stream access point (SAP) sample group 'sap' specified in ISO/IEC 14496-12 is used to provide information of all SAPs.

When `layer_id_method_idc` is equal to 0, a SAP is interpreted as follows:

- If the sample entry type is 'vvc1', 'vvil' or 'vvs1', and the track does not contain any sublayer with `TemporalId` equal to 0, a SAP specifies access to all the sublayers present in the track.
- Otherwise, a SAP specifies access to all layers present in the track.

NOTE If the sample entry type is 'vvc1' or 'vvil' and the track does not contain any sublayer with `TemporalId` equal to 0, an STSA picture with `TemporalId` equal to the lowest `TemporalId` present in the track serves as a SAP.

When `SAP_type` is less than or equal to 3 or both `SAP_type` is equal to 4 and `layer_id_method_idc` is equal to 0 or 1, the samples mapped to a 'sap' sample group description entry shall conform to the constraints of Table 13 for the 'vvc1' or 'vvil' sample entry type or to the constraints of Table 14 for the 'vvs1' sample entry type.

When a SAP sample group is present in any VVC track carrying one or more layers or sublayers of a VVC bitstream, a SAP sample group shall be present in each of the VVC tracks carrying one or more layers or sublayers of that VVC bitstream.

11.4.3 Random access recovery point sample group

The random access recovery point sample group 'roll' specified in ISO/IEC 14496-12 is used to provide information on recovery points for gradual decoding refresh.

When a 'roll' sample group is present in an VVC tracks, the track shall contain samples with `TemporalId` equal to 0.

When a 'roll' sample group concerns a dependent layer but not its reference layer(s), the sample group indicates characteristics that apply when all the reference layers of the dependent layer are available and decoded. The sample group can be used to initiate decoding of the predicted layer.

11.4.4 Alternative startup sequences sample group

The use of the Alternative Startup Sequences sample group (specified in ISO/IEC 14496-12) is recommended to be used only with random access points consisting of CRA pictures.

11.4.5 Random access point sample group

The random access point sample group 'rap' specified in ISO/IEC 14496-12 should not be used for VVC tracks or VVC subpicture tracks. The use of the 'sap' sample grouping is preferred for VVC tracks and VVC subpicture tracks.

In the context of this clause, the leading samples, defined as part of the definition of the 'rap' sample group in ISO/IEC 14496-12, contain RASL pictures in all layers present in the track.

11.4.6 Temporal level sample group

The temporal level sample group 'tele' specified in ISO/IEC 14496-12 should be used to indicate the TemporalId value. When the 'tele' sample group is present in a VVC track, the pictures with TemporalId equal to tidValue shall be mapped to the sample group description index equal to tidValue + 1. When the 'tele' sample group is present in a 'vvs1' VVC subpicture track, the VVC subpictures with TemporalId equal to tidValue shall be mapped to the sample group description index equal to tidValue + 1.

11.4.7 Step-wise sublayer access sample group

The step-wise sublayer access sample group 'stsa' specified in subclause 8.4.7 is used to mark step-wise sublayer access (STSA) pictures, when the 'stsa' sample group is present in a VVC track.

When an 'stsa' sample group is used with 'vvs1' VVC subpicture tracks, the step-wise sublayer access sample group is used to mark step-wise sublayer access (STSA) subpictures and the SampleToGroupBox, when present, shall have version equal to 0.

11.4.8 Decoding time hint sample group

Each decoding time hint sample group description entry ('opth') records a delta time in terms of the clock ticks (given by timescale of the MediaHeaderBox). The corrected decoding time is defined as the sum of the delta time associated with a sample through the SampleToGroupBox of type 'opth' and the decoding time of the sample. The corrected decoding times conform to the hypothetical reference decoder of ISO/IEC 23090-3.

All SampleToGroupBoxes for the decoding time hint sample group shall include grouping_type_parameter. The grouping_type_parameter field is specified for the decoding time hint sample group as follows:

```

unsigned int(16) reserved = 0;
unsigned int(16) operating_point_index;

```

`operating_point_index` specifies the index of the operating point, as given in the associated Operating Points Information sample group description (when applicable) or in the Operating Point entity group (when applicable), for which this sample group provides the corrected decoding times. A value of 0 indicates the first operating point in that sample group description.

The syntax and semantics for the 'opth' sample group are specified in subclauses 9.6.4.2 and 9.6.4.3, respectively.

11.4.9 Layer information sample group

The list of layers and sublayers that a track carries is signalled in the layer information ('linfo') sample group as specified in subclause 4.15.

NOTE When the current track contains more than one layer, more than one sublayer, or one sublayer but the sublayer is not sublayer with `TemporalId` equal to 0, layer and sublayer information can be needed to interpret information signalled in other sample groups, such as the 'sap', 'rap', 'sync', 'tele', and 'vopi' sample groups. The value of `layer_info_present_flag` has to be set appropriately.

When a 'linfo' sample group is present in a VVC track, `sub_layer_presence_flags` shall be equal to 0.

When several VPSs are referenced by a VVC bitstream, the `SampleGroupDescriptionBox` with `grouping_type` equal to 'linfo' may need several entries. When a single VPS is present or when the allocation of layers and sublayers to a track remains unchanged, it is recommended to use the `default_group_description_index` and include the Layer Information sample group in the `SampleTableBox` of the track.

11.4.10 Operating points information sample group

11.4.10.1 Definition

Applications are informed about the different operating points provided by a given VVC elementary stream and their constitution by using the Operating Points Information sample group ('vopi'). Each operating point is related to an output layer set, a max `TemporalId` value, and a profile, tier, and level signalling. All this information is captured by the 'vopi' sample group. Apart from this information, this sample group also provides the dependency information between layers.

When more than one VVC track is present for a VVC elementary stream and no Operating Point entity group is present for the VVC elementary stream, both of the following apply:

- Among the VVC tracks for the VVC elementary stream there shall be one and only one track that carries a 'vopi' sample group.
- All the other VVC tracks of the VVC elementary stream shall have a track reference of type 'vref' to the track that carries the 'vopi' sample group.

For any specific sample in a given track, the temporally collocated sample in another track is defined as the one with the same decoding time as that of this specific sample. For each sample S_N in a track T_N that has a 'vref' track reference to the track T_k that carries the 'vopi' sample group, the following applies:

- If there is a temporally collocated sample S_k in the track T_k , then the sample S_N is associated with the same 'vopi' sample group entry as the sample S_k .
- Otherwise, the sample S_N is associated with the same 'vopi' sample group entry as the last of the samples in the track T_k that precede the sample S_N in decoding time.

When several VPSs are referenced by a VVC elementary stream, it may be needed to include several entries in the sample group description box with `grouping_type` equal to 'vopi'. When one and only one VPS is present, it is recommended to use the default sample group description index and include the operating points information sample group in the sample table box, rather than including it in each track fragment.

The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type 'vopi'.

11.4.10.2 Syntax

```
class VvcOperatingPointsRecord {
    unsigned int(8) num_profile_tier_level_minus1;
    for (i=0; i<=num_profile_tier_level_minus1; i++) {
        bit(5) reserved = 0;
        unsigned int(3) ptl_max_temporal_id[i];
        VvcPTLRecord(ptl_max_temporal_id[i]+1) ptl[i];
    }
    unsigned int(1) all_independent_layers_flag;
    if (all_independent_layers_flag){
        unsigned int(1) each_layer_is_an_ols_flag;
        bit(1) reserved = 0;
    } else
        unsigned int(2) ols_mode_idc;
    bit(2) reserved = 0;
    unsigned int(1) incomplete_operating_points_flag;
    unsigned int(1) num_ref_sublayer_for_ols_present_flag;
    unsigned int(9) num_olss;
    for (i=0; i<num_olss; i++) {
        unsigned int(8) ptl_idx[i];
        unsigned int(9) ols_idx[i];
        unsigned int(6) layer_count[i];
        bit(1) reserved = 0;
        for (j=0; j<layer_count[i]; j++) {
            unsigned int(6) layer_id[i][j];
            unsigned int(1) is_output_layer[i][j];
            bit(1) reserved = 0;
            if(!is_outputlayer[i][j] && num_ref_sublayer_for_ols_present_flag)
                unsigned int(8) num_ref_sublayers_in_layer_in_ols[i][j];
        }
    }
    bit(4) reserved = 0;
    unsigned int(12) num_operating_points;
}
```

```

for (i=0; i<num_operating_points; i++) {
    unsigned int(9) ols_loop_entry_idx;
    unsigned int(3) max_temporal_id;
    unsigned int(1) frame_rate_info_flag;
    unsigned int(1) bit_rate_info_flag;
    if (incomplete_operating_points_flag)
        unsigned int(2) op_availability_idc;
    else
        bit(2) reserved = 0;
    bit(3) reserved = 0;
    unsigned int(2) chroma_format_idc;
    unsigned int(3) bit_depth_minus8;
    unsigned int(16) max_picture_width;
    unsigned int(16) max_picture_height;
    if (frame_rate_info_flag) {
        unsigned int(16) avg_frame_rate;
        bit(6) reserved = 0;
        unsigned int(2) constant_frame_rate;
    }
    if (bit_rate_info_flag) {
        unsigned int(32) max_bit_rate;
        unsigned int(32) avg_bit_rate;
    }
}
if (!all_independent_layers_flag) {
    unsigned int(8) max_layer_count;
    for (i=0; i<max_layer_count; i++) {
        unsigned int(8) layer_id;
        unsigned int(8) num_direct_ref_layers;
        for (j=0; j<num_direct_ref_layers; j++)
            unsigned int(8) direct_ref_layer_id;
    }
}
}

class VvcOperatingPointsInformation extends VisualSampleGroupEntry ('vopi') {
    VvcOperatingPointsRecord oin;
}

```

11.4.10.3 Semantics

`num_profile_tier_level_minus1` plus 1 gives the number of the subsequent profiles, tier, and level combinations as well as the associated fields.

`ptl_max_temporal_id[i]` gives the maximum TemporalID of NAL units of the associated bitstream for the specified *i*-th profile, tier, and level structure.

NOTE The semantics of `ptl_max_temporal_id[i]` and `max_temporal_id` of an operating point, given below, are different even though they can carry the same numerical value.

`ptl[i]` specifies the *i*-th profile, tier, and level structure.

`all_independent_layers_flag`, `each_layer_is_an_ols_flag`, `ols_mode_idc` and `max_tid_il_ref_pics_plus1` are defined in ISO/IEC 23090-3.

`incomplete_operating_points_flag` equal to 1 specifies that some operating points declared in the structure may be incomplete (i.e., some output layer sets or temporal sublayers may be missing in some operating points). `incomplete_operating_points_flag` equal to 0 specifies that all operating points declared in the structure are complete (i.e., all output layer sets and temporal sublayers associated with the operating points are present in the file).

`num_ref_sublayer_for_ols_present_flag` equal to 1 specifies that the syntax element `num_ref_sublayers_in_layer_in_ols[i][j]` may be present. `num_ref_sublayer_for_ols_present_flag` equal to 0 specifies that the syntax element `num_ref_sublayers_in_layer_in_ols[i][j]` is not present.

`num_olss` specifies the number of output layer sets signalled in this syntax structure. The value of `num_olss` shall be less than or equal to the value of `TotalNumOlss` as specified in ISO/IEC 23090-3.

`ptl_idx[i]` specifies the zero-based index of the listed profile, tier, and level structure for the *i*-th output layer set signalled in this syntax structure.

`ols_idx[i]` is the output layer set index of the *i*-th output layer set signalled in this syntax structure.

`layer_count[i]` specifies the number of layers in the *i*-th output layer set signalled in this syntax structure.

`layer_id[i][j]` specifies the `nuh_layer_id` value for the *j*-th layer in the *i*-th output layer set signalled in this syntax structure.

`is_output_layer[i][j]` equal to 1 specifies that the *j*-th layer is an output layer in the *i*-th output layer set signalled in this syntax structure. `is_output_layer[i][j]` equal to 0 specifies that the *j*-th layer is not an output layer in the *i*-th output layer set signalled in this syntax structure.

`num_ref_sublayers_in_layer_in_ols[i][j]` equal to 0 specifies that the pictures of the layer with layer Id equal to `layer_id[i][j]` that are neither IRAP pictures nor GDR pictures with `ph_recovery_poc_cnt` equal to 0 are not used as inter-layer reference pictures for decoding of pictures of other layers in the output layer set. `num_ref_sublayers_in_layer_in_ols[i][j]` greater than 0 specifies that, for decoding pictures of the other layers of the output layer set, no picture from the layer with layer Id equal to `layer_id[i][j]` with `TemporalId` greater than `num_ref_sublayers_in_layer_in_ols[i][j] - 1` is used as inter-layer reference picture and no APS with `nuh_layer_id` equal to `layer_id[i][j]` and `TemporalId` greater than `num_ref_sublayers_in_layer_in_ols[i][j] - 1` is referenced. When not present, the value of `num_ref_sublayers_in_layer_in_ols[i][j]` is inferred to be equal to 7.

`num_operating_points` gives the number of operating points for which the information follows. The list of operating points shall include all operating points that are present in sample entry of tracks representing the bitstream.

`ols_loop_entry_idx` is the index to the list of output layer sets signalled in this syntax structure for the operating point. The variable `olsIdx` is set equal to `ols_idx[ols_loop_entry_idx]` for the present loop entry.

`max_temporal_id` indicates the maximum `TemporalId` of NAL units of this operating point.

`frame_rate_info_flag` equal to 0 indicates that no frame rate information is present for the operating point. The value 1 indicates that frame rate information is present for the operating point.

`bit_rate_info_flag` equal to 0 indicates that no bitrate information is present for the operating point. The value 1 indicates that bitrate information is present for the operating point.

`op_availability_idc` equal to 0 indicates that the operating point is complete. `op_availability_idc` equal to 1 indicates that at least one layer is missing in the operating point. `op_availability_idc` equal to 2 indicates that at least one temporal sublayer is missing in the operating point. `op_availability_idc` equal to 3 indicates that at least one layer is missing in the operating point and additionally at least one temporal sublayer is missing in the operating point.

NOTE The process specified in subclause 11.5.4 is specified for operating points when `incomplete_operating_points_flag` is equal to 0 or `op_availability_idc` is equal to 0 (operating point is complete). When `op_availability_idc` is equal to 1, 2 or 3, a file reader needs to inspect the file to determine if the operating point can be processed and might need to alter the bitstream reconstruction process compared to what is specified in subclause 11.5.4.

`chroma_format_idc` indicates the chroma format that applies to this operating point. The following constraints apply for `chroma_format_idc`:

- If this operating point contains only one layer, the value of `sps_chroma_format_idc`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `chroma_format_idc` shall be equal to that value of `sps_chroma_format_idc`.
- Otherwise (this operating point contains more than one layer), the value of `chroma_format_idc` shall be equal to the value of `vps_ols_dpb_chroma_format[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`bit_depth_minus8` indicates the bit depth that applies to this operating point. The following constraints apply for `bit_depth_minus8`:

- If this operating point contains only one layer, the value of `sps_bitdepth_minus8`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `bit_depth_minus8` shall be equal to that value of `sps_bitdepth_minus8`.
- Otherwise (this operating point contains more than one layer), the value of `bit_depth_minus8` shall be equal to the value of `vps_ols_dpb_bitdepth_minus8[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`max_picture_width` indicates the maximum picture width, in units of luma samples, that applies to this operating point. The following constraints apply for `max_picture_width`:

- If this operating point contains only one layer, the value of `sps_pic_width_max_in_luma_samples`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `max_picture_width` shall be equal to that value of `sps_pic_width_max_in_luma_samples`.
- Otherwise (this operating point contains more than one layer), the value of `max_picture_width` shall be equal to the value of `vps_ols_dpb_pic_width[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`max_picture_height` indicates the maximum picture height, in units of luma samples, that applies to this operating point. The following constraints apply for `max_picture_height`:

- If this operating point contains only one layer, the value of `sps_pic_height_max_in_luma_samples`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `max_picture_height` shall be equal to that value of `sps_pic_height_max_in_luma_samples`.
- Otherwise (this operating point contains more than one layer), the value of `max_picture_height` shall be equal to the value of `vps_ols_dpb_pic_height[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`avg_frame_rate` gives the average frame rate in units of frames/(256 seconds) for the operating point. Value 0 indicates an unspecified average frame rate. When the bitstream of the operating point contains multiple layers, this gives the average access unit rate.

`constant_frame_rate` equal to 1 indicates that the bitstream of the operating point is of constant frame rate. Value 2 indicates that the representation of each sublayer in the bitstream of the operating point is of constant frame rate. Value 0 indicates that the bitstream of the operating point may or may not be of constant frame rate. `constant_frame_rate` equal to 3 is reserved. When the bitstream of the operating point contains multiple layers, this gives the indication of whether the bitstream of the operating point has constant access unit rate.

`max_bit_rate` gives the maximum bit rate in bits/second of the bitstream of the operating point, over any window of one second.

`avg_bit_rate` gives the average bit rate in bits/second of the bitstream of the operating point.

`max_layer_count` specifies the count of all unique layers in all of the operating points described in this syntax structure.

`layer_id` specifies `nuh_layer_id` of a layer for which all the direct reference layers are given in the following loop of `direct_ref_layer_id`.

`num_direct_ref_layers` specifies the number of direct reference layers for the layer with `nuh_layer_id` equal to `layer_id`. When `all_independent_layers_flag` is equal to 1, `num_direct_ref_layers` is inferred to be equal to 0 for each layer.

`direct_ref_layer_id` indicates `nuh_layer_id` of the direct reference layer.

`max_tid_il_ref_pics_plus1` equal to 0 specifies that the pictures of the layer with `nuh_layer_id` equal to `direct_ref_layer_id` that are neither IRAP pictures nor GDR pictures with `ph_recovery_poc_cnt` equal to 0 are not used as inter-layer reference pictures for decoding of pictures of the layer with `nuh_layer_id` equal to `layer_id`. A value greater than 0 specifies that, for decoding pictures of the layer with `nuh_layer_id` equal to `layer_id`, no picture from the layer with `nuh_layer_id` equal to `direct_ref_layer_id` with `TemporalId` greater than `max_tid_il_ref_pics_plus1 - 1` is used as an inter-layer reference picture and no APS with `nuh_layer_id` equal to `direct_ref_layer_id` and `TemporalId` greater than `max_tid_il_ref_pics_plus1 - 1` is referenced.

11.4.11 Decoding capability information sample group

11.4.11.1 Definition

A sample group description entry of this sample group contains a DCI NAL unit.

NOTE When a decoder is initialized in a manner that it can handle coded video sequences of any `profile_tier_level()` syntax structures included in the DCI NAL unit without re-initialization and two consecutive samples in decoding order reference two sample entries that are mapped to the same decoding capability information sample group description entry, a player could switch sample entries without re-initialization of the decoder. When two consecutive samples in decoding order belong to different VVC bitstreams, switching sample entries without decoder re-initialization could require special handling of the coded picture buffer and the decoder picture buffer.

If any DCI NAL unit is present in any sample entry or inband, it shall be exactly the same as the DCI NAL unit included in the corresponding decoding capability information sample group.

For playing a single track, the decoding capability information sample group from that track applies.

When a VVC bitstream for a particular operating point is reconstructed from multiple VVC tracks, the following applies:

- When a 'vopi' sample group is present for the VVC bitstream, the decoding capability information sample group from the track carrying the 'vopi' sample group applies.
- When an 'opeg' entity group is present for the VVC bitstream, the decoding capability information sample group from the first track listed in the 'opeg' entity group as belonging to the operating point applies.

11.4.11.2 Syntax

```
class DecodingCapabilityInformation extends VisualSampleGroupEntry ('dcfi') {
    unsigned int(16) dci_nal_unit_length;
    bit(8*dci_nal_unit_length) dci_nal_unit;
}
```

11.4.11.3 Semantics

`dci_nal_unit_length` indicates the length in bytes of the DCI NAL unit.
`dci_nal_unit` contains a DCI NAL unit as specified in ISO/IEC 23090-3.

11.4.12 Parameter set sample group

11.4.12.1 Definition

A sample group description entry of this sample group contains an SPS NAL unit.

When a sample is mapped to a parameter set sample group ('pase'), it indicates that the SPS NAL unit contained within the sample group description entry needs to be inserted into the reconstructed AU if the target maximum picture format corresponds to the picture format indicated by the `grouping_type_parameter` field in the `SampleToGroupBoxes` for the parameter set sample group.

When a sample group of type 'pase' is present in a VVC track, a sample shall be marked as belonging to the 'pase' sample group when any of the following is true:

- The sample of the VVC track contains an SPS NAL unit.
- The VVC track has a track reference of type 'vvcN' to a VVC non-VCL track that and there is a corresponding sample in the associated VVC non-VCL track having the same decoding time that contains an SPS NAL unit.
- The sample references a different sample entry than the previous sample in decoding order and the sample entry contains an SPS NAL unit.

All instances of the `SampleToGroupBox` for the parameter set sample group shall include `grouping_type_parameter`. The `grouping_type_parameter` field is specified for the parameter set sample group as follows:

```
unsigned int(16) max_width;
unsigned int(16) max_height;
```

`max_width` and `max_height` specify the maximum picture format of the target reconstructed bitstream for which a SPS NAL unit is to be inserted.

11.4.12.2 Syntax

```
class ParameterSetNALUEntry() extends VisualSampleGroupEntry ('pase')
{
    unsigned int(16) ps_nalu_length;
    bit(8* ps_nalu_length) ps_nal_unit;
}
```

11.4.12.3 Semantics

`ps_nalu_length` indicates the length in bytes of the SPS NAL unit.
`ps_nal_unit` contains an SPS NAL unit as specified in ISO/IEC 23090-3.

11.4.13 Access unit delimiter sample group**11.4.13.1 Definition**

A sample group description entry of this sample group contains an AUD NAL unit.

When a sample is mapped to an access unit delimiter sample group ('aud '), it indicates that the AUD NAL unit contained within the sample group description entry needs to be inserted into the reconstructed AU if the target operating point corresponds to any of the output layer sets indicated by the `grouping_type_parameter` field in the `SampleToGroupBoxes` for the access unit delimiter sample group.

All instances of the `SampleToGroupBox` for the access unit delimiter sample group shall include `grouping_type_parameter`. The `grouping_type_parameter` field is specified for the access unit delimiter sample group as follows:

```
bit(22) reserved = 0;
unsigned int(9) ols_idx;
unsigned int(1) lower_ols_idx_inclusive;
```

`ols_idx` specifies the OLS that the sample group applies to.

`lower_ols_idx_inclusive` equal to 0 specifies that the sample group applies only to the OLS with index equal to `ols_idx`; `lower_ols_idx_inclusive` equal to 1 specifies that the sample group applies to any OLS with index from 0 to `ols_idx`, inclusive.

11.4.13.2 Syntax

```
class AUDSampleEntry() extends VisualSampleGroupEntry ('aud ')
{
    bit(24) audNalUnit;
}
```

11.4.13.3 Semantics

`audNalUnit` contains an AUD NAL unit as specified in ISO/IEC 23090-3.

11.4.14 End of sequence sample group**11.4.14.1 Definition**

A sample group description entry of this sample group contains one or more EOS NAL units.

When a sample is mapped to an end of sequence sample group ('eos '), it indicates that the EOS NAL units contained within the sample group description entry needs to be inserted in the reconstructed AU if the target operating point corresponds to any of the output layer sets and maximum TemporalId indicated by the `grouping_type_parameter` field in the instances of the `SampleToGroupBox` for the end of sequence sample group.

All instances of the `SampleToGroupBox` for the end of sequence sample group shall include `grouping_type_parameter`. The `grouping_type_parameter` field is specified for the end of sequence sample group as follows:

```
bit(19) reserved = 0;
unsigned int(9) ols_idx;
unsigned int(3) max_tid;
unsigned int(1) lower_ols_idx_inclusive;
```

`ols_idx` and `max_tid` specify the operating point that the sample group applies to.

`lower_ols_idx_inclusive` equal to 0 specifies that the sample group applies only to the operating point with OLS index equal to `ols_idx`. `lower_ols_idx_inclusive` equal to 1 specifies that the sample group applies to any OLS with index from 0 to `ols_idx`, inclusive.

NOTE The assignment of samples to an 'eos ' sample group description entry and to instances of the `SampleToGroupBox` with particular `max_tid` values is described with an example as follows: In the example, AU(N) is the last AU in a CVS and has TemporalId equal to 1, and AU(N-1) has TemporalId equal to 0. AU(N-1) is mapped to an 'eos ' sample group with `max_tid` equal to 0, which indicates that if the selected operating point has the highest TemporalId value equal to 0, the EOS NAL units in the mapped sample group description entry need to be inserted into the reconstructed AU(N-1). AU(N) is mapped to an 'eos ' sample group with `max_tid` equal to 1, which indicates that if the selected operating point has the highest TemporalId value equal to 1, the EOS NAL units in the mapped sample group description entry need to be inserted into the reconstructed AU(N).

11.4.14.2 Syntax

```
class EndOfSequenceSampleEntry() extends VisualSampleGroupEntry ('eos ')
{
    unsigned int(8) num_eos_nal_unit_minus1;
    for (i=0; i <= num_eos_nal_unit_minus1; i++)
        bit(16) eosNalUnit[i];
}
```

11.4.14.3 Semantics

`num_eos_nal_unit_minus1` plus 1 specifies the number of EOS NAL units in the sample group description entry.

`eosNalUnit[i]` contains the *i*-th EOS NAL unit in the sample group description entry. The syntax and semantics of the EOS NAL unit are specified in ISO/IEC 23090-3.

11.4.15 End of bitstream sample group

11.4.15.1 Definition

A sample group description entry of this sample group contains an EOB NAL unit.

When a sample is mapped to an end of bitstream sample group ('eob '), it indicates that the EOB NAL unit contained within the sample group description entry needs to be inserted into the reconstructed AU

if the target operating points corresponds to any of the output layer sets and maximum TemporalId indicated by the `grouping_type_parameter` field in the instances of the `SampleToGroupBox` for the end of bitstream sample group.

All instances of the `SampleToGroupBox` for the end of bitstream sample group shall include `grouping_type_parameter`. The `grouping_type_parameter` field is specified for the end of bitstream sample group as follows:

```
bit(19) reserved = 0;
unsigned int(9) ols_idx;
unsigned int(3) max_tid;
unsigned int(1) lower_ols_idx_inclusive;
```

`ols_idx` and `max_tid` specify the operating point that the sample group applies to.
`lower_ols_idx_inclusive` equal to 0 specifies that the sample group applies only to the operating point with OLS index equal to `ols_idx`. `lower_ols_idx_inclusive` equal to 1 specifies that the sample group applies to any OLS with index from 0 to `ols_idx`, inclusive.

11.4.15.2 Syntax

```
class EndOfBitstreamSampleEntry() extends VisualSampleGroupEntry ('eob ')
{
    bit(16) eobNalUnit;
}
```

11.4.15.3 Semantics

`eobNalUnit` contains an EOB NAL unit as specified in ISO/IEC 23090-3.

11.4.16 Subpicture ID sample group

11.4.16.1 Definition

This sample group may be present in such VVC tracks and VVC subpicture tracks that contain one or more VVC subpictures with `sh_subpic_id` present in the slice headers. This sample group shall not be present in tracks that do not contain VVC subpictures with `sh_subpic_id` present in the slice headers. Each sample group description entry indicates a list of one or more subpicture ID values in decoding order that are contained in samples mapped to this sample group description entry.

11.4.16.2 Syntax

```
aligned(8) class VvcSubpicIDEntry() extends VisualSampleGroupEntry('spid')
{
    unsigned int(1) rect_region_flag;
    bit(2) reserved = 0;
    unsigned int(1) continuous_id_flag;
    unsigned int(12) num_subpics_minus1;
    for (i = 0; i <= num_subpics_minus1; i++) {
        if ((continuous_id_flag && i == 0) || !continuous_id_flag)
            unsigned int(16) subpic_id[i];
        if (rect_region_flag)
            unsigned int(16) groupID[i];
    }
}
```

11.4.16.3 Semantics

`rect_region_flag` equal to 0 specifies that no association information to rectangular regions is provided for the subpictures contained in the sample mapped to this sample group description entry. `rect_region_flag` equal to 1 specifies that association information to rectangular regions is provided for the subpictures contained in the sample mapped to this sample group description entry.

`num_subpics_minus1` plus 1 specifies the number of VVC subpictures contained in the sample mapped to this sample group description entry.

`subpic_id[i]` specifies the *i*-th subpicture identifier, in decoding order, contained in the sample mapped to this sample group description entry. The value of `subpic_id[i]` shall not be equal to any other value of `subpic_id[j]`, where *i* is not equal to *j*, in the same `VvcSubpicIDEntry`. When `continuous_id_flag` is equal to 1, the value of `subpic_id[i]` for *i* in the range of 1 to `num_subpics_minus1`, inclusive, is equal to `subpic_id[i-1]+1`.

`groupID[i]` specifies unique identifier of the rectangular region associated with the *i*-th subpicture contained in the sample mapped to this sample group description entry.

11.4.17 Subpicture order sample group**11.4.17.1 Definition**

This sample group is used in VVC merge base tracks with 'subp' track referencing to VVC subpicture tracks. Each sample group description entry indicates the subpictures or slices of a coded picture in decoding order, where each index of the track reference of type 'subp' indicates one or more subpictures or slices that are contiguous in decoding order.

To ease PPS or SPS rewriting in response to subpicture selection, each sample group description entry may contain a `VVCSubpicIDRewritingInformationStruct` structure that contains:

- an indication of whether selected subpicture IDs should be changed in PPS or SPS NAL units;
- the length (in bits) of subpicture ID syntax elements;
- the bit position of subpicture ID syntax elements in the containing RBSP;
- a flag indicating whether start code emulation prevention bytes are present before or within subpicture IDs;
- the parameter set ID of the parameter set containing the subpicture IDs.

11.4.17.2 Syntax of VVCSubpicIDRewritingInfomationStruct

```
aligned(8) class VVCSubpicIDRewritingInfomationStruct()
{
    unsigned int(4) subpic_id_len_minus1;
    unsigned int(12) subpic_id_bit_pos;
    unsigned int(1) start_code_emul_flag;
    unsigned int(1) pps_sps_subpic_id_flag;
    if (pps_sps_subpic_id_flag)
        unsigned int(6) pps_id;
    else {
        unsigned int(4) sps_id;
        bit(2) reserved = 0;
    }
}
```

11.4.17.3 Semantics of VVCSubpicIDRewritingInfomationStruct

subpic_id_len_minus1 plus 1 specifies the number of bits in subpicture identifier syntax elements in PPS or SPS, whichever is referenced by this structure.

subpic_id_bit_pos specifies the bit position starting from 0 of the first bit of the first subpicture ID syntax element in the referenced PPS or SPS RBSP.

start_code_emul_flag equal to 0 specifies that start code emulation prevention bytes are not present before or within subpicture IDs in the referenced PPS or SPS NAL unit.
start_code_emul_flag equal to 1 specifies that start code emulation prevention bytes may be present before or within subpicture IDs in the referenced PPS or SPS NAL unit.

pps_sps_subpic_id_flag, when equal to 1, specifies that the PPS NAL units applying to the samples mapped to this sample group description entry contain subpicture ID syntax elements.
pps_sps_subpic_id_flag, when equal to 0, specifies that the SPS NAL units applying to the samples mapped to this sample group description entry contain subpicture ID syntax elements.

pps_id, when present, specifies the PPS ID of the PPS applying to the samples mapped to this sample group description entry.

sps_id, when present, specifies the SPS ID of the SPS applying to the samples mapped to this sample group description entry.

11.4.17.4 Syntax of VvcSubpicOrderEntry

```
aligned(8) class VvcSubpicOrderEntry() extends VisualSampleGroupEntry('spor')
{
    unsigned int(1) subpic_id_info_flag;
    unsigned int(15) num_subpic_ref_idx;
    for (i = 0; i < num_subpic_ref_idx; i++)
        unsigned int(16) subp_track_ref_idx[i];
    if (subpic_id_info_flag)
        VVCSubpicIDRewritingInfomationStruct() subpic_id_rewriting_info;
}
```

11.4.17.5 Semantics of VvcSubpicOrderEntry

subpic_id_info_flag equal to 0 specifies that the subpicture ID values explicitly signalled in the SPSs and/or PPSs or inferred without explicit signalling are correct for the indicated set of subp_track_ref_idx[i] values and thus no rewriting of SPSs or PPSs is required.
subpic_id_info_flag equal to 1 specifies that SPSs and/or PPSs may need rewriting to indicate the subpictures corresponding to the set of subp_track_ref_idx[i] values and the subpicture ID values are not inferred.

`num_subpic_ref_idx` greater than 0 specifies the number of reference indices of subpicture tracks or track groups of subpicture tracks referenced by the VVC track. `num_subpic_ref_idx` equal to 0 specifies that 1) the number of reference indices of subpicture tracks or track groups of subpicture tracks referenced by the VVC track is equal to the number of entries in the 'subp' track reference of the VVC track, and 2) the order of subpicture tracks or track groups of subpicture tracks to be resolved for inclusion into the VVC bitstream reconstructed from the VVC track is the same as the order of the entries in the 'subp' track reference of the VVC track.

`subp_track_ref_idx[i]` specifies a 'subp' track reference index of the *i*-th list of one or more subpictures or slices to be included in the VVC bitstream reconstructed from the VVC track.

`subpic_id_rewriting_info` contains syntax elements assisting in rewriting subpicture IDs in PPS or SPS NAL units.

11.4.18 Subpicture layout map entry

11.4.18.1 Definition

The `VvcSubpictureLayoutMapEntry` may be used to assign an identifier, called `groupID`, to each loop entry of the 'spor' sample group.

The `VvcSubpictureLayoutMapEntry`, when present, shall be linked to a sample group description providing the semantics of that `groupID`. This link shall be provided by setting the `groupID_info_4cc` to the four-character code of the associated sample grouping type. It is forbidden to indicate an associated `grouping_type` for which the sample group description definition does not provide semantics for `groupID`.

If `num_subpic_ref_idx` in `VvcSubpicOrderEntry` is greater than 0, the *i*-th loop entry in the `VvcSubpicLayoutMapEntry` maps to `subp_track_ref_idx[i]` in `VvcSubpicOrderEntry`. Otherwise (`num_subpic_ref_idx` equal to 0), the *i*-th loop entry in the `VvcSubpicLayoutMapEntry` maps to the *i*-th entry of the 'subp' track reference.

When `groupID_info_4cc` is equal to 'trif' in a VVC merge base track, the same track shall include a `SampleGroupDescriptionBox` of type 'trif' with entries constrained as follows:

- `rect_region_flag` shall be equal to 1.
- `full_picture` shall be equal to 0.

11.4.18.2 Syntax

```
class VvcSubpicLayoutMapEntry() extends VisualSampleGroupEntry ('sulum') {
    unsigned int(32) groupID_info_4cc;
    unsigned int(16) entry_count_minus1;
    for(i=0; i <= entry_count_minus1; i++)
        unsigned int(16) groupID;
}
```

11.4.18.3 Semantics

`groupID_info_4cc` specifies that further information of the `groupID` values specified in this entry are provided in a `SampleGroupDescriptionBox` with `grouping_type` equal to `groupID_info_4cc`.

`entry_count_minus1` plus 1 specifies the number of entries in the map. When `num_subpic_ref_idx` in `VvcSubpicOrderEntry` is equal to 0, `entry_count_minus1` plus 1 shall be equal to the number of entries in the 'subp' track reference. Otherwise, `entry_count_minus1` plus 1 shall be equal to `num_subpic_ref_idx` in `VvcSubpicOrderEntry`.

`groupID` specifies the unique identifier of the group. More information about the group is provided by the sample group description entry with this `groupID` and `grouping_type` equal to `groupID_info_4cc`.

11.4.19 Mixed NAL unit type pictures sample group

11.4.19.1 Definition

This sample group is used in VVC merge base tracks, i.e., in VVC tracks with 'subp' track referencing to VVC subpicture tracks. Each sample group description entry indicates a group of pairs of 'mixn' track reference indices, which reference VVC subpicture tracks or track groups. When a VVC bitstream is resolved from the VVC merge base track containing the 'minp' sample group by merging the pairs of VVC subpicture tracks signalled in the 'minp' sample group entry then there is mixing of different NAL unit types.

To ease PPS rewriting in response to subpicture selection, each sample group description entry may contain:

- the bit position of the `pps_mixed_nalu_types_in_pic_flag` syntax element in the containing RBSP;
- the parameter set ID of the parameter set containing the `pps_mixed_nalu_types_in_pic_flag` syntax element.

11.4.19.2 Syntax

```
aligned(8) class VvcMixedNALUnitTypePicEntry() extends
VisualSampleGroupEntry('minp')
{
    unsigned int(16) num_mix_nalu_pic_idx;
    for (i = 0; i < num_mix_nalu_pic_idx; i++){
        unsigned int(16) mix_subp_track_idx1[i];
        unsigned int(16) mix_subp_track_idx2[i];
    }
    unsigned int(10) pps_mix_nalu_types_in_pic_bit_pos;
    unsigned int(6) pps_id;
}
```

11.4.19.3 Semantics

`num_mix_nalu_pic_idx` specifies the number of pairs of 'mixn' track reference indices of subpicture tracks or track groups of subpicture tracks referenced by the VVC merge base track in this mixed NAL unit type pictures sample group description entry.

`mix_subp_track_idx1[i]` and `mix_subp_track_idx2[i]`, for each value of `i`, specifies a 'mixn' track reference index value, each referencing a `track_ID` value of a VVC subpicture track or a `track_group_id` value of an 'snut' or 'alte' track group of VVC subpicture

tracks. The value of `mix_subp_track_idx1[i]` shall not be equal to `mix_subp_track_idx2[i]`.

`pps_mix_nalu_types_in_pic_bit_pos` specifies the bit position starting from 0 of the `pps_mixed_nalu_types_in_pic_flag` syntax element in the referenced PPS RBSP.

`pps_id` specifies the PPS ID of the PPS applying to the samples mapped to this sample group description entry.

11.4.20 Rectangular region order sample group

11.4.20.1 Definition

This sample group is used in VVC extraction base tracks. Each sample group description entry indicates the rectangular regions of a picture in decoding order, where each `region_id[i]` value present in the sample group description entry identifies a rectangular region.

To ease PPS or SPS rewriting in response to rectangular region selection, each sample group description entry may contain a `VVCSubpicIDRewritingInfomationStruct` structure that is specified in subclauses 11.4.17.2 and 11.4.17.3 and contains:

- an indication of whether selected subpicture IDs should be changed in PPS or SPS NAL units;
- the length (in bits) of subpicture ID syntax elements;
- the bit position of subpicture ID syntax elements in the containing RBSP;
- a flag indicating whether start code emulation prevention bytes are present before or within subpicture IDs;
- the parameter set ID of the parameter set containing the subpicture IDs.

11.4.20.2 Syntax

```
aligned(8) class VvcRectRegionOrderEntry () extends
VisualSampleGroupEntry('rror')
{
    unsigned int(1) subpic_id_info_flag;
    bit(7) reserved = 0;
    unsigned int(16) num_alternate_region_set;
    for (i = 0; i < num_alternate_region_set; i++) {
        unsigned int(16) num_regions_in_set[i];
        unsigned int(16) alternate_region_set_id[i];
        for (j = 0; j < num_regions_in_set[i]; j++)
            unsigned int(16) groupID[i][j];
    }
    unsigned int(16) num_regions_minus1;
    for (i = 0; i < num_regions_minus1; i++)
        unsigned int(16) region_id[i];
    if (subpic_id_info_flag)
        VVCSubpicIDRewritingInfomationStruct() subpic_id_rewriting_info;
}
```

11.4.20.3 Semantics

`subpic_id_info_flag` equal to 0 specifies that the subpicture ID values explicitly signalled in the SPSs and/or PPSs or inferred without explicit signalling are correct for the indicated set of

`region_id[i]` values and thus no rewriting of SPSs or PPSs is required. `subpic_id_info_flag` equal to 1 specifies that SPSs and/or PPSs may need rewriting to indicate the subpictures corresponding to the set of `region_id[i]` values and the subpicture ID values are not inferred.

`num_alternate_region_set` specifies the number of alternate region sets signalled in this sample group description entry.

`num_regions_in_set[i]` specifies the number of rectangular regions of the *i*-th alternate region set in this sample group description entry.

`alternate_region_set_id[i]` specifies the unique identifier of the *i*-th alternate region set. The value of `alternate_region_set_id[i]` shall not be equal to 0 and shall not be equal to any `groupID` in the `NALUMapEntry` in the referenced VVC track by 'recr' track reference in this track.

`groupID[i][j]` specifies the unique identifier of *j*-th rectangular region that belongs to the *i*-th alternate region set. `groupID[i][j]` value is equal to `groupID` in the `NALUMapEntry` in the VVC track referenced by the 'recr' track reference in this track. The value of `groupID[i][j]` shall not be equal to 0.

`num_regions_minus1` plus 1 specifies the number of rectangular regions.

`region_id[i]`, for each value of *i*, specifies the value of `groupID` in the `NALUMapEntry` in the referenced VVC track of the *i*-th rectangular region to be included in the VVC bitstream reconstructed from this track or the value of `alternate_region_set_id` of an alternate region set of the *i*-th rectangular region to be included in the VVC bitstream reconstructed from this track. The value of `region_id[i]` shall not be equal to 0.

`subpic_id_rewriting_info` contains syntax elements assisting in rewriting subpicture IDs in PPS or SPS NAL units. The syntax and semantics of `subpic_id_rewriting_info` are specified in specified in subclauses 11.4.17.2 and 11.4.17.3, respectively.

11.4.21 Subpicture level information sample group

11.4.21.1 Definition

When a subpicture entity group with `level_is_static_flag` equal to 0 is present, a subpicture level information sample group ('spli') shall be present in the track with `track_ID` equal to the `level_info_entity_idx`-th `entity_id` value of the subpicture entity group.

A subpicture level information sample group provides level information for one or more groups of VVC subpicture tracks indicated by a subpicture entity group, referred to as the associated subpicture entity group. Information for identifying the associated subpicture entity group is signalled by the `grouping_type_parameter` field in the instances of the `SampleToGroupBox` for the subpicture level information sample group.

All instances of the `SampleToGroupBox` for the subpicture level information sample group shall include `grouping_type_parameter`. The `grouping_type_parameter` field is specified for the subpicture level information sample group as follows:

```
unsigned int(32) subpic_entity_group_id;
```

`subpic_entity_group_id` specifies the `group_id` in the `EntityToGroupBox` of the associated subpicture entity group.

11.4.21.2 Syntax

```
aligned(8) class SubpicLevelInfoEntry() extends VisualSampleGroupEntry('spli')
{
    unsigned int(8) level_idc;
}
```

11.4.21.3 Semantics

`level_idc` specifies the level to which the set of samples that are members of this sample group in each combination of subpicture tracks indicated by the associated subpicture entity group conforms.

11.5 Entity groups

11.5.1 Subpicture entity groups

11.5.1.1 General

Subpicture entity groups are defined to provide level information indicating conformance of a merged bitstream out of several VVC subpicture tracks.

NOTE The VVC merge base track provides another mechanism for merging VVC subpicture tracks.

The implicit reconstruction process based on the subpicture entity groups requires modification of parameter sets. The subpicture entity groups give guidance to ease parameter set generation for the reconstructed bitstreams.

When the VVC subpicture tracks within the group that are to be jointly decoded are interchangeable, i.e. the player can select any set of `num_active_tracks` subpicture tracks from a group with the same level contribution, the `SubpicCommonGroupBox` indicates the combinatorial rules.

When there are coded subpictures with different properties, e.g. different resolutions, which are selected to be jointly decoded, the `SubpicMultipleGroupBox` indicates the combinatorial rules.

Both `SubpicCommonGroupBox` and `SubpicMultipleGroupBox` may contain the `level_idc` value applicable when decoding jointly any combination of VVC subpicture tracks indicated in the entity group. When the applicable `level_idc` value is not static, both `SubpicCommonGroupBox` and `SubpicMultipleGroupBox` may indicate the track that contains the associated subpicture level information sample group that indicates the applicable `level_idc` value. An applicable `level_idc` value is provided in at least one of a subpicture entity group and the associated subpicture level information sample group.

All the `entity_id` values included in the subpicture entity groups shall identify VVC subpicture tracks. When present, `SubpicCommonGroupBox` and `SubpicMultipleGroupBox` shall be contained in the `GroupsListBox` in the file-level `MetaBox` and shall not be contained in `MetaBoxes` of other levels.

11.5.1.2 Syntax of subpicture common group box

```
aligned(8) class SubpicCommonGroupBox extends EntityToGroupBox('acgl',0,0)
{
    unsigned int(1) level_is_present_flag;
    unsigned int(1) level_is_static_flag;
    bit(6) reserved = 0;
    if( level_is_present_flag )
        unsigned int(8) level_idc;
    if( level_is_static_flag == 0 )
        unsigned int(32) level_info_entity_idx;
    unsigned int(16) num_active_tracks;
}
```

11.5.1.3 Semantics of subpicture common group box

level_is_present_flag equal to 1 specifies that level_idc is present in this SubpicCommonGroupBox. level_is_present_flag equal to 0 specifies that level_idc is not present in this SubpicCommonGroupBox.

level_is_static_flag equal to 1 specifies that the level to which any set of num_active_tracks entities among the current entity group conforms is static for all samples in the selection of tracks and is indicated by the level_idc field in the current SubpicCommonGroupBox. level_is_static_flag equal to 0 specifies that the level for such a selection, for a particular set of samples, is indicated by the level_idc field in the sample group entry of the associated subpicture level information sample group, which is in the track with track_ID equal to level_info_entity_idx-th entity_id value in this SubpicCommonGroupBox and has subpic_entity_group_id equal to the group_id in this SubpicCommonGroupBox. The particular set of samples is the set of samples that are members of the associated subpicture level information sample group. When level_is_present_flag is equal to 0, level_is_static_flag shall be equal to 0.

level_idc specifies the level to which any set of num_active_tracks entities among the current entity group conforms.

level_info_entity_idx specifies the track containing the associated subpicture level information sample group. The track with track_ID equal to level_info_entity_idx-th entity_id value in this SubpicCommonGroupBox should be a track that contains picture header NAL units.

num_active_tracks specifies the number of tracks for which the value of level_idc value is provided in this SubpicCommonGroupBox or in the associated subpicture level information sample group.

11.5.1.4 Syntax of subpicture multiple groups box

```
aligned(8) class SubpicMultipleGroupsBox extends EntityToGroupBox('amgl', 0, 0)
{
    unsigned int(1) level_is_present_flag;
    unsigned int(1) level_is_static_flag;
    bit(6) reserved = 0;
    if( level_is_present_flag )
        unsigned int(8) level_idc;
    if( level_is_static_flag == 0 )
        unsigned int(32) level_info_entity_idx;
    unsigned int(16) num_subgroup_ids;
    subgroupIdLen = (num_subgroup_ids >= (1 << 8)) ? 16 : 8;
    for (i = 0; i < num_entities_in_group; i++)
        unsigned int(subgroupIdLen) track_subgroup_id[i];
    for (i = 0; i < num_subgroup_ids; i++)
        unsigned int(16) num_active_tracks[i];
}
```

11.5.1.5 Semantics

`level_is_present_flag` equal to 1 specifies that `level_idc` is present in this `SubpicMultipleGroupsBox`. `level_is_present_flag` equal to 0 specifies that `level_idc` is not present in this `SubpicMultipleGroupsBox`.

`level_is_static_flag` equal to 1 specifies that the level to which the set of any `num_active_tracks[i]` tracks among the subgroup with ID equal to `i` for all values of `i` in the range of 0 to `num_subgroup_ids - 1`, inclusive, conforms is static for all samples in the subpicture tracks included in the current entity group, and is indicated by the `level_idc` field in the current `SubpicMultipleGroupsBox`. `level_is_static_flag` equal to 0 specifies that the level for such a combination, for a particular set of samples, is indicated by the `level_idc` field in the sample group entry of the associated subpicture level information sample group, which is in the track with `track_id` equal to `level_info_entity_idx-th entity_id` value in this `SubpicMultipleGroupsBox` and has `subpic_entity_group_id` equal to the `group_id` in the `EntityToGroupBox` of the current entity group. The particular set of samples is the set of samples that are members of the associated subpicture level information sample group. When `level_is_present_flag` is equal to 0, `level_is_static_flag` shall be equal to 0.

`level_idc` specifies the level to which the set of any `num_active_tracks[i]` tracks among the subgroup with ID equal to `i` for all values of `i` in the range of 0 to `num_subgroup_ids - 1`, inclusive, conforms.

`level_info_entity_idx` specifies the track containing the associated subpicture level information sample group. The track with `track_ID` equal to `level_info_entity_idx-th entity_id` value in this `SubpicMultipleGroupsBox` should be a track that contains picture header NAL units.

`num_subgroup_ids` specifies the number of separate subgroups, each identified by the same value of `track_subgroup_id[i]`. Different subgroups are identified by different values of `track_subgroup_id[i]`.

`track_subgroup_id[i]` specifies the subgroup ID for the `i-th` track in this entity group. subgroup ID values shall range from 0 to `num_subgroup_ids - 1`, inclusive.

`num_active_tracks[i]` specifies the number of tracks among the subgroup with ID equal to `i`.

11.5.2 Operating point entity group

11.5.2.1 General

The operating point entity group is defined to provide a mapping of tracks to operating points and information of the operating points, such as profiles and levels.

When an 'mpeg' entity group is present, the file shall contain a file-level `ExtendedTypeBox` with a brand combination including the 'unif' brand specified in ISO/IEC 14496-12.

The implicit reconstruction process when aggregating samples of the tracks mapped to an operating point described in this entity group does not require removing any further VCL NAL units to result in a conforming VVC bitstream. Tracks belonging to an Operating Point Entity Group shall have a track reference of type 'vreg' to the `group_id` indicated in the Operating Point Entity Group and shall not carry a 'vopi' sample group.

All the `entity_id` values included in the operating point entity groups shall represent track IDs of tracks that belong to the same VVC bitstream. When present, `OperatingPointGroupBox` shall be contained in the `GroupsListBox` in the file-level `MetaBox` and shall not be contained in `MetaBoxes` of other levels.

11.5.2.2 Syntax

```

aligned(8) class OperatingPointGroupBox extends EntityToGroupBox('opeg',0,0)
{
    unsigned int(8) num_profile_tier_level_minus1;
    for (i=0; i<=num_profile_tier_level_minus1; i++)
        VvcPTLRecord(0) opeg_ptl[i];
    bit(6) reserved = 0;
    unsigned int(1) incomplete_operating_points_flag;
    unsigned int(9) num_olss;
    for (i=0; i<num_olss; i++) {
        unsigned int(8) ptl_idx[i];
        unsigned int(9) ols_idx[i];
        unsigned int(6) layer_count[i];
        bit(1) reserved = 0;
        unsigned int(1) layer_info_present_flag[i];
        if (layer_info_present_flag) {
            for (j=0; j<layer_count; j++) {
                unsigned int(6) layer_id[i][j];
                unsigned int(1) is_output_layer[i][j];
                bit(1) reserved = 0;
            }
        }
    }
    bit(4) reserved = 0;
    unsigned int(12) num_operating_points;
    for (i=0; i<num_operating_points; i++) {
        unsigned int(9) ols_loop_entry_idx;
        unsigned int(3) max_temporal_id;
        unsigned int(1) frame_rate_info_flag;
        unsigned int(1) bit_rate_info_flag;
        if (incomplete_operating_points_flag)
            unsigned int(2) op_availability_idc;
        else
            bit(2) reserved = 0;
        bit(3) reserved = 0;
        unsigned int(2) chroma_format_idc;
        unsigned int(3) bit_depth_minus8;
        unsigned int(16) max_picture_width;
        unsigned int(16) max_picture_height;
        if (frame_rate_info_flag) {
            unsigned int(16) avg_frame_rate;
            bit(6) reserved = 0;
            unsigned int(2) constant_frame_rate;
        }
        if (bit_rate_info_flag) {
            unsigned int(32) max_bit_rate;
            unsigned int(32) avg_bit_rate;
        }
        unsigned int(8) entity_count;
        for (j=0; j<entity_count; j++)
            unsigned int(8) entity_idx;
    }
}

```

11.5.2.3 Semantics

num_profile_tier_level_minus1 plus 1 gives the number of following profiles, tier, and level combinations as well as the associated fields.

`mpeg_ptl[i]` specifies the *i*-th profile, tier, and level structure.

`incomplete_operating_points_flag` equal to 1 specifies that some operating points declared in the structure may be incomplete (i.e., some output layer sets or temporal sublayers may be missing in some operating points). `incomplete_operating_points_flag` equal to 0 specifies that all operating points declared in the structure are complete (i.e., all output layer sets and temporal sublayers associated with the operating points are present in the file).

`num_olss` specifies the number of output layer sets signalled in this syntax structure. The value of `num_olss` shall be less than or equal to the value of `TotalNumOlss` as specified in ISO/IEC 23090-3.

`ptl_idx[i]` specifies the zero-based index of the listed profile, tier, and level structure for the *i*-th output layer set signalled in this syntax structure.

`ols_idx[i]` is the output layer set index of the *i*-th output layer set signalled in this syntax structure.

`layer_count[i]` specifies the number of layers in the *i*-th output layer set signalled in this syntax structure.

`layer_info_present_flag[i]` specifies whether the list of layers that belong to the *i*-th output layer set is present or not.

`layer_id[i][j]` specifies the `nuh_layer_id` value for the *j*-th layer in the *i*-th output layer set signalled in this syntax structure.

`is_output_layer[i][j]` equal to 1 specifies that the *j*-th layer is an output layer in the *i*-th output layer set signalled in this syntax structure. `is_output_layer[i][j]` equal to 0 specifies that the *j*-th layer is not an output layer in the *i*-th output layer set signalled in this syntax structure.

`num_operating_points` gives the number of operating points for which the information follows. The list of operating points shall include all operating points that are present in sample entry of tracks representing the bitstream.

`ols_loop_entry_idx` is the index to the list of output layer sets signalled in this syntax structure for the operating point. The variable `olsIdx` is set equal to `ols_idx[ols_loop_entry_idx]` for the present loop entry.

`max_temporal_id` gives the maximum `TemporalId` of NAL units of this operating point.

`frame_rate_info_flag` equal to 0 indicates that no frame rate information is present for the operating point. The value 1 indicates that frame rate information is present for the operating point.

`bit_rate_info_flag` equal to 0 indicates that no bitrate information is present for the operating point. The value 1 indicates that bitrate information is present for the operating point.

`op_availability_idc` equal to 0 indicates that the operating point is complete.

`op_availability_idc` equal to 1 indicates that at least one layer is missing in the operating point. `op_availability_idc` equal to 2 indicates that at least one temporal sublayer is missing in the operating point. `op_availability_idc` equal to 3 indicates that at least one layer is missing in the operating point and additionally at least one temporal sublayer is missing in the operating point.

NOTE The process specified in subclause 11.5.4 is specified for operating points when `incomplete_operating_points_flag` is equal to 0 or `op_availability_idc` is equal to 0 (operating point is complete). When `op_availability_idc` is equal to 1, 2 or 3, a file reader needs to inspect the file to determine if the operating point can be processed and might need to alter the bitstream reconstruction process compared to what is specified in subclause 11.5.4.

`chroma_format_idc` indicates the chroma format that applies to this operating point. The following constraints apply for `chroma_format_idc`:

— If this operating point contains only one layer, the value of `sps_chroma_format_idc`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `chroma_format_idc` shall be equal to that value of `sps_chroma_format_idc`.

— Otherwise (this operating point contains more than one layer), the value of `chroma_format_idc` shall be equal to the value of `vps_ols_dpb_chroma_format[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`bit_depth_minus8` indicates the bit depth that applies to this operating point. The following constraints apply for `bit_depth_minus8`:

— If this operating point contains only one layer, the value of `sps_bitdepth_minus8`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `bit_depth_minus8` shall be equal to that value of `sps_bitdepth_minus8`.

— Otherwise (this operating point contains more than one layer), the value of `bit_depth_minus8` shall be equal to the value of `vps_ols_dpb_bitdepth_minus8[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`max_picture_width` indicates the maximum picture width, in units of luma samples, that applies to this operating point. The following constraints apply for `max_picture_width`:

— If this operating point contains only one layer, the value of `sps_pic_width_max_in_luma_samples`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `max_picture_width` shall be equal to that value of `sps_pic_width_max_in_luma_samples`.

— Otherwise (this operating point contains more than one layer), the value of `max_picture_width` shall be equal to the value of `vps_ols_dpb_pic_width[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`max_picture_height` indicates the maximum picture height, in units of luma samples, that applies to this operating point. The following constraints apply for `max_picture_height`:

— If this operating point contains only one layer, the value of `sps_pic_height_max_in_luma_samples`, as defined in ISO/IEC 23090-3, shall be the same in all SPSs referenced by the VCL NAL units in the VVC bitstream of this operating point, and the value of `max_picture_height` shall be equal to that value of `sps_pic_height_max_in_luma_samples`.

— Otherwise (this operating point contains more than one layer), the value of `max_picture_height` shall be equal to the value of `vps_ols_dpb_pic_height[ MultiLayerOlsIdx[ olsIdx ] ]`, as defined in ISO/IEC 23090-3.

`avg_frame_rate` gives the average frame rate in units of frames/(256 seconds) for the operating point. Value 0 indicates an unspecified average frame rate.

`constant_frame_rate` equal to 1 indicates that the bitstream of the operating point is of constant frame rate. Value 2 indicates that the representation of each sublayer in the bitstream of the operating point is of constant frame rate. Value 0 indicates that the bitstream of the operating point may or may not be of constant frame rate. Value 3 is reserved.

`max_bit_rate` gives the maximum bit rate in bits/second of the bitstream of the operating point, over any window of one second.

`avg_bit_rate` gives the average bit rate in bits/second of the bitstream of the operating point.

`entity_count` specifies the number of tracks that are present in an operating point.

`entity_idx` specifies the index to the `entity_id` list in the entity group that belongs to an operating point.

11.5.3 VVC bitstream entity group

The VVC bitstream entity group is defined to provide the VVC tracks of a single VVC elementary stream.

When an `EntityToGroupBox` with `grouping_type` equal to 'vvcb' is present, the following constraints apply:

- a) The `EntityToGroupBox` shall be contained in the `GroupsListBox` in the file-level `MetaBox` and shall not be contained in `MetaBoxes` of other levels.
- b) The `entity_id` values included in the `EntityToGroupBox` shall represent the track IDs of all the VVC tracks carrying one VVC elementary stream.
- c) The first `entity_id` value included in the `EntityToGroupBox` shall represent the track ID of a VVC track that carries the 'vopi' sample group.
- d) The VVC tracks with track ID values represented by the second and later `entity_id` values included in the `EntityToGroupBox` shall not carry the 'vopi' sample group.

11.5.4 VVC switchable tracks entity group

11.5.4.1 General

The VVC switchable tracks entity group defines a set of tracks among which it is possible to switch using the sample description box of a single track included in the entity group.

In the following, track *k* refers to the track with `track_ID` equal to the *k*-th `entity_id` value of the entity group, where *k* is a variable. The sample description box of the track *i* may be used as follows. For a particular value of `track_switch_hierarchy_id[i]`, any track *j* with `track_switch_hierarchy_id[j]` less than or equal to `track_switch_hierarchy_id[i]` may be used with the sample description box in the track *i*. Switching from one track to another is guaranteed to lead to a valid VVC bitstream when the switch is performed at a SAP sample with a SAP type less than or equal to 3. When a 'defi' sample group is present in track *i*, it also applies to any track *j* of the entity group with value `track_switch_hierarchy_id[j]` smaller than or equal to `track_switch_hierarchy_id[i]`.

When an `EntityToGroupBox` with `grouping_type` equal to 'swtk' is present, all the following constraints apply:

- The `EntityToGroupBox` shall be contained in the `GroupsListBox` in the file-level `MetaBox` and shall not be contained in any `MetaBox` of other levels.
- All the `entity_id` values in the entity group shall be `track_ID` values of VVC tracks.

- `lengthSizeMinusOne` shall have the same value in all instances of `VvcConfigurationBox` and `VvcNALUConfigBox` contained in the sample entries of the VVC tracks included in the entity group and of all VVC tracks, VVC subpicture tracks, and VVC non-VCL tracks referenced by the VVC tracks included in the entity group.

A `TrackSelectionBox` may be present in the set of tracks included in the same VVC switchable tracks entity group. Tracks belonging to the same group can be differentiated by the differentiating attribute `'swtk'`. In this case, the pointer to the field or information that distinguishes the track is the `track_switch_hierarchy_id[entityIdx]` value in the VVC switchable tracks entity group for which `track_ID` is equal to `entity_id[entityIdx]`.

11.5.4.2 Syntax

```
aligned(8) class SwitchableTracks extends EntityToGroupBox('swtk', 0, 0)
{
    for (i = 0; i < num_entities_in_group; i++)
        unsigned int(16) track_switch_hierarchy_id[i];
}
```

11.5.4.3 Semantics

`track_switch_hierarchy_id[i]` specifies the hierarchy of a track within the tracks belonging to the entity group.

11.6 Data sharing and VVC bitstream reconstruction

11.6.1 General

The output of the process specified in this subclause is a VVC bitstream, which is referred to as the output bitstream.

A file reader shall invoke the process specified in this subclause when one or more of the following invocation conditions are true:

- a) A VVC bitstream (`'vvcb'`) entity group is present in a file and the file reader processes the VVC bitstream represented by the entity group to generate the output bitstream.
- b) An operating point (`'opep'`) entity group is present in a file and the file reader uses any of the operating points described by the entity group to generate the output bitstream.
- c) The file reader extracts a subset of the layers or sublayers of a VVC track with `ptl_present_flag` equal to 1 to generate the output bitstream.
- d) The file reader processes a VVC merge base track.
- e) The file reader processes a VVC extraction base track.
- f) The file reader processes a VVC track that has an associated VVC non-VCL track.
- g) The file reader processes a VVC track that contains a `'pase'` sample group.

When a VVC switchable tracks entity group ('swtk') is present in a file, a file reader initially selects a `VvcDecoderConfigurationRecord` among the tracks that belong to the VVC switchable tracks entity group with a particular value of `track_switch_hierarchy_id[i]`. The file reader may switch at SAP samples with SAP type less than or equal to 3 to any other track that belongs to the VVC switchable tracks entity group and has value of `track_switch_hierarchy_id[j]` less than or equal to the particular value of `track_switch_hierarchy_id[i]`. The initially selected `VvcDecoderConfigurationRecord` is used for any track with a value of `track_switch_hierarchy_id[j]` less than or equal to the particular value of `track_switch_hierarchy_id[i]`.

The process specified in this subclause consists of the following ordered steps:

- a) When the invocation condition a, b, or c above is true, an operating point is determined at the start of the VVC bitstream and may be determined again for any IRAP or GDR access unit.

The method of determining the operating point is outside the scope of this document.

When `incomplete_operating_points_flag` is equal to 1 in an 'mpeg' entity group or a 'vopi' sample group, the selected operating point shall be such that `op_availability_idc` is equal to 0 for the selected operating point.

When the file reader either selects an operating point the first time or selects a different operating point than what was selected earlier, the file reader shall indicate the output layer set index and the highest TemporalId value of the selected operating point to the VVC decoder either by including them in an OPI NAL unit inserted into the output bitstream (as the first NAL unit following the AU delimiter NAL unit, if any, in the first access unit where the operating point is taken into use) or by other means.

NOTE It is possible to down-switch and up-switch layers and sublayers at AUs that do not start a CVS as long as the set of layers and sublayers being decoded is within the latest operating point that was indicated to the decoder through an OPI NAL unit or by other means. Layer up-switching can take place at an IRAP, GDR, or STSA picture unit with TemporalId equal to 0. Sublayer up-switching can take place at an STSA picture unit.

The subsequent ordered steps apply to the sequence of access units in decoding order starting from an access unit wherein the operating point is determined, inclusive, to the end of the bitstream or the access unit wherein the operating point is determined next time, exclusive, whichever is earlier in decoding order.

- b) When a VVC bitstream is represented by multiple VVC tracks, a file parser identifies the tracks needed for the chosen operating point as follows:

- 1) When an operating point from the 'mpeg' entity group contains the selected operating point, select the tracks that belong to the operating point as indicated by the 'mpeg' entity group.
- 2) When an 'mpeg' entity group is not present (i.e. when a 'vopi' sample group is present), determine, from the 'vvc' entity group which VVC tracks represent a VVC bitstream. The first `entity_id` of the 'vvc' entity group identifies the track containing the 'vopi'

sample group. The mapping of operating points to layers and sublayers is concluded from the 'vopi' sample group. The set of tracks containing the layers and sublayers of the selected operating point and hence needed for decoding the selected operating point are concluded from the 'linfo' sample groups present in the VVC tracks of the VVC bitstream.

Since a particular layer or sublayer may be represented by more than one track, when figuring out the required tracks for an operating point, a selection may need to be made among the set of tracks that all carry the particular layer or sublayer.

- c) Access units to the output bitstream are reconstructed in the decoding time order of samples among the VVC tracks needed for the chosen operating point (when the invocation condition a, b, or c above is true), or within the VVC merge base track (when the invocation condition d above is true), or within the VVC extraction base track (when the invocation condition e above is true), or within the VVC track (when the invocation condition f above is true).

If several tracks contain data for the access unit, the alignment of respective samples in tracks is performed based on the sample decoding times.

A sequence of access units is reconstructed from the respective samples in the required tracks by invoking the implicit reconstruction process specified in subclause 11.6.2 repetitively.

The reconstructed access units are placed into the output bitstream in increasing order of decoding time.

- d) When both of the following conditions are true:
- the sequence of access units is followed by a selection a different operating point than what was selected earlier, and
 - the sequence of access units does not end with EOS NAL units in each layer of the applied operating point or an EOB NAL unit,
 - the file reader shall include an EOS NAL unit for each layer of the applied operating point into the output bitstream. The included EOS NAL units may be placed at the end of the access unit in the increasing order of the nuh_layer_id value of the EOS NAL units.

When a VVC bitstream is represented by multiple VVC tracks, the decoding times of the samples shall be such that if the tracks were combined into a single bitstream ordered by increasing decoding time, the access unit order would be correct as specified in ISO/IEC 23090-3.

11.6.2 Implicit reconstruction of a VVC bitstream

This process specifies the reconstruction of an access unit from the time-aligned samples having the current decoding time among:

- the required VVC track(s),
- the associated VVC non-VCL tracks, if any, and

- the referenced VVC subpicture tracks, if any.

A time-aligned sample in each required VVC track or each associated VVC non-VCL track, if any, is the sample with the same decoding time as the current decoding time. A time-aligned sample in each referenced VVC subpicture track, if any, is the sample that has the greatest decoding time that is less than or equal to the current decoding time.

When reconstructing a bitstream containing a sublayer for which the VCL NAL units have TemporalId greater than 0, all lower sublayers (i.e., those for which the VCL NAL units have smaller TemporalId) within the same layer are also included in the resulting bitstream.

When the sample(s) with the current decoding time contain VCL NAL units with TemporalId greater than the greatest TemporalId included in the selected operating point, no access unit is reconstructed from the current decoding time.

When a sample of the track is marked as belonging to a sample group 'pase' and the grouping_type_parameter includes max_width and max_height equal to max_picture_width and max_picture_height in the VvcDecoderConfigurationRecord initially selected to reconstruct the bitstream, the following applies:

- The SPS NAL unit within the 'pase' sample group is placed before the NAL units present in the sample and after the AUD NAL unit, if any.
- Any SPS NAL units in the reconstructed access unit as specified in this subclause that are not inserted by a 'pase' sample group are removed.

NOTE 1 When a VVC switchable tracks entity group is present, the file reader initially selects a VvcDecoderConfigurationRecord among the tracks that belong to the VVC switchable tracks entity group and this VvcDecoderConfigurationRecord is also used for extracting the SPS NAL unit within the 'pase' sample group in a different track belonging to the VVC switchable tracks entity group when reconstructing a sample of that track.

When reconstructing an access unit, picture units (as specified in ISO/IEC 23090-3) from samples having the same decoding time are placed into the access unit in increasing order of the nuh_layer_id value. The following steps may be performed when the condition applies:

- When a sample of the track containing the first picture unit of a sample is marked as belonging to a sample group 'aud' with ols_idx and lower_ols_idx_inclusive corresponding to the target operating point, the AUD NAL unit within the 'aud' sample group description entry is placed into the AU as the first NAL unit of the AU.
- When a sample in a track is marked as belonging to a sample group 'eos' with ols_idx, max_tid and lower_ols_idx_inclusive corresponding to the target operating point, the EOS NAL units within the 'eos' sample group description entry are placed into the AU after all NAL units from the sample, excluding the EOB NAL unit, when present.
- When a sample in a track is marked as belonging to a sample group 'eob' with ols_idx, max_tid, and lower_ols_idx_inclusive corresponding to the target

operating point, the EOB NAL unit within the 'eob' sample group description entry is placed into the AU after all NAL units of the AU (including the EOS NAL units, if any).

Only the picture units from the layers and sublayers that are in the target operating point are included in the output bitstream.

NOTE 2 When the Operating Points Entity Group is not present, there could be layers or sublayers that are carried in the tracks required for bitstream reconstruction but do not belong to the target operating point.

NOTE 3 VVC decoder implementations take as input a bitstream corresponding to the target output layer set index and the highest TemporalId value of the target operating point, which respectively correspond to the TargetOlsIdx and HighestTid variables of ISO/IEC 23090-3:2021, Clause 8. A file parser needs to ensure that the reconstructed bitstream does not contain any other layers and sublayers than those included in the target operating point before sending it to a VVC decoder.

When reconstructing an access unit based on an operating point associated with output layer set index i (in the for loop over `num_ols` in the 'vopi' sample group), the following applies:

- When reconstructing an access unit, for each layer in the output layer set with index j in the range from 0 to `layer_count[i]-1`, inclusive, if `num_ref_sublayers_in_layer_in_ols[i][j]` is greater than 0, then the VCL NAL units belongs to the sublayers of that layer for which the VCL NAL units have TemporalId less than or equal to $\text{Min}(\text{num_ref_sublayers_in_layer_in_ols}[i][j]-1, \text{max_temporal_id})$, where `max_temporal_id` is the value of the corresponding syntax element for the operating point, are included in the resulting bitstream and the required tracks are selected accordingly.
- When reconstructing an access unit, for each layer in the output layer set with index j in the range from 0 to `layer_count[i]-1`, inclusive, if `num_ref_sublayers_in_layer_in_ols[i][j]` is equal to 0, from all the picture units of the reference layers only IRAP picture units and GDR picture units with `ph_recovery_poc_cnt` equal to 0 are included in the resulting bitstream and the required tracks are selected accordingly.

When a VVC track contains a 'subp' or 'vvcN' track reference, each picture unit is reconstructed as specified in subclause 11.6.3. When a VVC track contains a 'recr' track reference, each picture unit is reconstructed as specified in subclause 11.6.6. The process in subclause 11.6.3 or 11.6.6 is repeated for each layer of the target operating point in increasing `nuh_layer_id` order.

NAL units and NAL-unit-like structures with unspecified NAL unit types (`nal_unit_type` in the range of UNSPEC_28 to UNSPEC_31, inclusive, i.e. `nal_unit_type` values in the range of 28 to 31, inclusive, as defined in ISO/IEC 23090-3), are discarded from the reconstructed access units.

The reconstructed access units are placed into the VVC bitstream in increasing order of decoding time.

Since a particular layer or sublayer may be represented by more than one track, when figuring out the required tracks for an operating point, a selection may need to be made among the set of tracks that all carry the particular layer or sublayer.

11.6.3 Reconstructing a picture unit from a sample in a VVC track with 'subp' or 'vvcN' track references

A sample of a VVC track is resolved to a picture unit in the order of the following numbered bullets:

- a) The AUD NAL unit, when present in the sample or in the time-aligned sample of the associated VVC non-VCL track, is included in the picture unit.

NOTE 1 When an AUD NAL unit is present in a sample, it is the first NAL unit in the sample.

- b) When the sample is the first sample of a sequence of samples associated with the same sample entry, the DCI, OPI, parameter set and SEI NAL units contained in the sample entry, if any, are included in the picture unit.
- c) When there is an associated VVC non-VCL track and the picture unit is the first picture unit in this access unit that is reconstructed from the sample, the following NAL units are included in the picture unit:
 - 1) If there is at least one NAL unit in the time-aligned sample of the associated VVC non-VCL track with nal_unit_type equal to EOS_NUT, EOB_NUT, SUFFIX_APS_NUT, SUFFIX_SEI_NUT, FD_NUT, or RSV_NVCL_27 (a NAL unit with such a NAL unit type cannot precede the first VCL NAL unit in a picture unit), the NAL units (excluding the AUD NAL unit, if any) in the time-aligned sample of the associated VVC non-VCL track up to and excluding the first of these NAL units.
 - 2) Otherwise all NAL units in the time-aligned sample of the associated VVC non-VCL track.
- d) If there is at least one NAL unit in the sample with nal_unit_type equal to EOS_NUT, EOB_NUT, SUFFIX_APS_NUT, SUFFIX_SEI_NUT, FD_NUT, or RSV_NVCL_27 (a NAL unit with such a NAL unit type cannot precede the first VCL NAL unit in a picture unit), the NAL units in the sample up to and excluding the first of these NAL units are included in the picture unit, otherwise all NAL units in the sample are included in the picture unit.
- e) If the reader has selected an operating point, the reader shall exclude any OPI NAL units stored in sample entries and samples from the reconstructed access unit in all of the steps above.

NOTE 2 It is discouraged to include OPI NAL units in sample entries or samples when multiple tracks are used to store layers or sublayers of a VVC bitstream.

- f) If the VVC track does not reference VVC subpicture tracks, the NAL units of the sample from the VVC track are included in the picture unit.

Otherwise, the following applies:

- 1) The track references are resolved as specified in subclause 11.6.4.
- 2) Parameter sets are updated when necessary as specified in subclause 11.6.5.
- 3) The picture unit is appended by the content of the time-aligned (in decoding time) resolved sample from each referenced VVC subpicture track in the order of the VVC

subpicture tracks being referenced in the 'subp' track reference (when num_subpic_ref_idx in the same group entry of the 'spor' sample group entry mapped to this sample is equal to 0) or in the order as specified in the 'spor' sample group description entry mapped to this sample (when num_subpic_ref_idx in the same group entry of the 'spor' sample group entry mapped to this sample is greater than 0), excluding

- all DCI, OPI, VPS, SPS, PPS, AUD, PH, EOS, and EOB NAL units; and
- when a 'vvc1'/'vvi1' VVC subpicture track is referenced, all SEI NAL units that do not contain a scalable nesting SEI message with sn_subpic_flag equal to 1.

If a referenced VVC subpicture track is associated with a VVC non-VCL track, the resolved sample of the VVC subpicture track contains the following NAL units:

- If there is at least one NAL unit in the time-aligned sample of the associated VVC non-VCL track with nal_unit_type equal to EOS_NUT, EOB_NUT, SUFFIX_APS_NUT, SUFFIX_SEI_NUT, FD_NUT, or RSV_NVCL_27 (a NAL unit with such a NAL unit type cannot precede the first VCL NAL unit in a picture unit), the NAL units (excluding the AUD NAL unit, if any) in the time-aligned sample of the associated VVC non-VCL track up to and excluding the first of these NAL units.
- Otherwise all NAL units in the time-aligned sample of the associated VVC non-VCL track.
- The NAL units from the sample of the referenced VVC subpicture track.
- The remaining NAL units, if any, from the time-aligned sample of the associated VVC non-VCL track.

Otherwise (a referenced VVC subpicture track is not associated with a VVC non-VCL track), the resolved sample of the VVC subpicture track is the sample of the VVC subpicture track.

- 4) After appending the picture unit by the content of the time-aligned (in decoding time) resolved sample from all the referenced VVC subpicture tracks, the following applies:

- The prefix APS NAL units are reordered prior to the first VCL NAL unit of the picture unit.
- The suffix APS NAL units are reordered after to the last VCL NAL unit of the picture unit.

- g) All NAL units in the sample with nal_unit_type equal to EOS_NUT, EOB_NUT, SUFFIX_APS_NUT, SUFFIX_SEI_NUT, FD_NUT, or RSV_NVCL_27 are included in the picture unit.

- h) When there is an associated VVC non-VCL track and the picture unit is the last picture unit in this access unit that is reconstructed from the sample, all NAL units in the time-aligned sample of the associated VVC non-VCL track with `nal_unit_type` equal to `EOS_NUT`, `EOB_NUT`, `SUFFIX_APS_NUT`, `SUFFIX_SEI_NUT`, `FD_NUT`, or `RSV_NVCL_27` are included in the picture unit.
- i) All NAL units and NAL-unit-like structures in the sample with `nal_unit_type` in the range of `UNSPEC_28` to `UNSPEC_31`, inclusive, are discarded from the picture unit.

11.6.4 Resolving subpicture track references

If `num_subpic_ref_idx` in the 'spor' sample group description entry mapped to this sample is equal to 0, each track reference in the 'subp' box is resolved as follows. Otherwise, the track reference `track_IDs[subp_track_ref_idx[i]]` in the 'subp' `TrackReferenceTypeBox` is resolved for each instance of `subp_track_ref_idx[i]` in the 'spor' sample group description entry mapped to this sample as follows.

NOTE 1 Each sample of the VVC merge base track resolved from the 'subp' track references forms a rectangular region without holes (i.e., all samples in the rectangular region are covered) and without overlaps (i.e., all samples in the rectangular region are covered only once).

- If the track reference points to a track ID of a VVC subpicture track, the track reference is resolved to the VVC subpicture track.
- Otherwise (the track reference points to an 'alte' track group), the track reference is resolved to any of the tracks of the 'alte' track group.

NOTE 2 A reference to an 'alte' track group is advised to be resolved to a track with `(flags & 1)` equal to 1 in a `VvcConfigurationBox` or `VvcNALUConfigBox` included in a sample entry only when the application omits the displaying of the track. For example, a player can select such a track when the content of the track is not used to produce a viewport for omnidirectional video represented by the VVC merge base track.

- The track reference shall not be resolved to a particular subpicture track more than once per a single picture unit.
- When a particular track reference index value was resolved to a particular track in the previous sample, it shall be resolved in the current sample to either of the following:
 - the same particular track, or
 - any other track in the same 'alte' track group that contains a sync sample that is time-aligned with the current sample.

NOTE 3 The VVC subpicture tracks in the same 'alte' track group are necessarily independent of any other VVC subpicture tracks referenced by the same VVC merge base track to avoid decoding mismatches and could therefore be constrained as follows:

- All the VVC subpicture tracks contain VVC subpictures.
- The subpicture boundaries are like picture boundaries.

11.6.5 Parameter set updating

If a reader selects VVC subpicture tracks containing VVC subpictures with a set of subpicture ID values that is the initial selection or differs from the previous selection, the following steps are taken:

- The 'spor' sample group description entry is studied to conclude whether a PPS or SPS NAL unit needs to be changed.

NOTE 1 An SPS change is only possible at the start of a CLVS.

- If the 'spor' sample group description entry indicates that start code emulation prevention bytes are present before or within the subpicture IDs in the containing NAL unit, an RBSP is derived from the NAL unit (i.e., start code emulation prevention bytes are removed). After the overriding in the next step, start code emulation prevention is re-done.
- The reader uses the bit position and subpicture ID length information in the 'spor' sample group entry to conclude which bits are overwritten to update the subpicture IDs to the selected ones.
- When the subpicture ID values of a PPS or SPS are initially selected, the reader needs to rewrite the PPS or SPS, respectively, with the selected subpicture ID values in the reconstructed access unit.
- When the subpicture ID values of a PPS or SPS are changed compared to the previous PPS or SPS (respectively) with the same PPS ID value or SPS ID value (respectively), the reader needs to include a copy of that previous PPS and SPS (if the PPS or SPS with that same PPS or SPS ID value, respectively, is not present in the access unit otherwise) and rewrite the PPS or SPS (respectively) with the updated subpicture ID values in the reconstructed access unit.

When there is a 'minp' sample group description entry mapped to the sample of the VVC merge base track, the following operations are applied:

- The 'minp' sample group description entry is studied to conclude the value of `pps_mixed_nalu_types_in_pic_flag` as follows:
 - The following examination is performed for each pair of track ID values of subpicture tracks selected to reconstruct a picture unit to determine whether they map to the track reference `mix_subp_track_idx1[i]` or `mix_subp_track_idx2[i]` in the 'minp' sample group description entry mapped to this sample:
 - If the track reference points to the track ID directly, the track ID maps to `mix_subp_track_idx1[i]` or `mix_subp_track_idx2[i]`.
 - Otherwise, if both of the following are true:
 - the track reference points to an 'alte' track group or a 'snut' track group, and

- the 'alte' track group or the 'snut' track group, respectively, contains a track having the track ID value,

the track ID value maps to `mix_subp_track_idx1[i]` or `mix_subp_track_idx2[i]`.

- Otherwise, the track ID value does not map to `mix_subp_track_idx1[i]` or `mix_subp_track_idx2[i]`.
- If any pair of subpicture tracks selected to reconstruct a picture unit have track ID values that map to `mix_subp_track_idx1[i]` and `mix_subp_track_idx2[i]` with the same value of `i`, `pps_mixed_nalu_types_in_pic_flag` is concluded to be equal to 1.
- Otherwise, `pps_mixed_nalu_types_in_pic_flag` is concluded to be equal to 0.
- If the value of `pps_mixed_nalu_types_in_pic_flag` differs from that in the previous PPS NAL unit with the same PPS ID in the reconstructed bitstream, the following applies:
 - When no PPS was included in the picture unit by the steps above, the reader needs to include a copy of the PPS with the updated `pps_mixed_nalu_types_in_pic_flag` value in the reconstructed picture unit.
 - The reader uses the bit position in the 'minp' sample group entry to conclude which bit is overwritten to update the `pps_mixed_nalu_types_in_pic_flag`.

11.6.6 Reconstructing a picture unit from a sample in a VVC track with 'recr' track reference

A sample of a VVC track is resolved to a picture unit in the order of the following numbered bullets:

- The AUD NAL unit, when present in the sample, is included in the picture unit.

NOTE 1 When an AUD NAL unit is present in a sample, it is the first NAL unit in the sample.

- When the sample is the first sample of a sequence of samples associated with the same sample entry, the DCI, OPI, parameter set and SEI NAL units contained in the sample entry, if any, are included in the picture unit.
- If there is at least one NAL unit in the sample with `nalu_unit_type` equal to `EOS_NUT`, `EOB_NUT`, `SUFFIX_APS_NUT`, `SUFFIX_SEI_NUT`, `FD_NUT`, or `RSV_NVCL_27` (a NAL unit with such a NAL unit type cannot precede the first VCL NAL unit in a picture unit), the NAL units in the sample up to and excluding the first of these NAL units is included in the picture unit, otherwise all NAL units in the sample are included in the picture unit.
- The picture unit is appended by the content of the time-aligned (in decoding time) sample with NAL units of the resolved regions from the referenced VVC track being referenced in the 'recr' track reference. The order of NAL units is specified by the order of regions in the 'rpor' sample group description entry mapped to this sample. The NAL units are resolved as specified below.

- 1) Each region is resolved to the rectangular region indicated by `region_id[i]` in the 'error' sample group description entry mapped to this sample as specified below.

- The region shall not be resolved to a particular region more than once per a single picture unit.
- If the value of `region_id[i]` is not equal to any `alternate_region_set_id[j]` defined in the 'error' sample group description entry, the resolved region corresponds to the rectangular region with unique identifier `groupID` equal `region_id[i]`.
- Otherwise (`region_id[i]` is equal to a value of `alternate_region_set_id[j]` defined in the 'error' sample group description entry), the region is resolved to any rectangular region with unique identifier `groupID` equal to any value of `groupID[j][k]` that belongs to the alternate region set with `alternate_region_set_id[j]` equal to `region_id[i]`, and if a particular `region_id[i]` value was resolved to a particular region in the previous sample, it shall be resolved in the current sample to either of the following:
 - the same particular region, or
 - any other region in the same alternate region set if a sync sample in the referenced VVC track is time-aligned with the current sample.

NOTE 2 The regions in the same alternate region set are necessarily independent of any other regions referenced by the same VVC extraction base track to avoid decoding mismatches and could therefore be constrained as follows:

- All the regions contain VVC subpictures.
- The subpicture boundaries are like picture boundaries.

- 2) NAL units of the resolved regions are identified by `groupID` in the 'nal' sample group description entry mapped to the time-aligned sample in the referenced VVC track.

- 3) After appending the picture unit with the NAL units of all the resolved regions, the following applies:

- The prefix APS NAL units are reordered prior to the first VCL NAL unit of the picture unit.
- The suffix APS NAL units are reordered after the last VCL NAL unit of the picture unit.

- e) All NAL units in the sample with `nal_unit_type` equal to `EOS_NUT`, `EOB_NUT`, `SUFFIX_APS_NUT`, `SUFFIX_SEI_NUT`, `FD_NUT`, or `RSV_NVCL_27` are included in the picture unit.

- f) All NAL units and NAL-unit-like structures in the sample with `nal_unit_type` in the range of UNSPEC_28 to UNSPEC_31, inclusive, are discarded from the picture unit.

If a reader selects regions containing VVC subpictures with a set of subpicture ID values that is the initial selection or differs from the previous selection, the following steps may be taken:

- The '`error`' sample group description entry is studied to conclude whether a PPS or SPS NAL unit needs to be changed.

NOTE 3 An SPS change is only possible at the start of a CLVS.

- If the '`error`' sample group description entry indicates that start code emulation prevention bytes are present before or within the subpicture IDs in the containing NAL unit, an RBSP is derived from the NAL unit (i.e., start code emulation prevention bytes are removed). After the overriding in the next step, start code emulation prevention is re-done.
- The reader uses the bit position and subpicture ID length information in the '`error`' sample group entry to conclude which bits are overwritten to update the subpicture IDs to the selected ones.
- When the subpicture ID values of a PPS or SPS are initially selected, the reader needs to rewrite the PPS or SPS, respectively, with the selected subpicture ID values in the reconstructed access unit.
- When the subpicture ID values of a PPS or SPS are changed compared to the previous PPS or SPS (respectively) with the same PPS ID value or SPS ID value (respectively), the reader needs to include a copy of that previous PPS and SPS (if the PPS or SPS with that same PPS or SPS ID value, respectively, is not present in the access unit otherwise) and rewrite the PPS or SPS (respectively) with the updated subpicture ID values in the reconstructed access unit.

12 EVC elementary streams and sample definitions

12.1 Overview

This clause defines the carriage of EVC elementary streams, as defined in ISO/IEC 23094-1, in the ISO base media file format. The elementary stream structure is provided in subclause 12.2. Sample and configuration definition is provided in subclause 12.3. Derivation from ISO base media file format are provided in subclause 12.4.

Internet media 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 in Clause E.9.

12.2 Elementary stream structure

EVC elementary streams are structured as NAL units and their storage in the ISO Base Media File Format follows similar principles as other NAL structured video formats.

The storage of EVC elementary streams is subdivided into two parts: static information that is globally used in the elementary stream and dynamic information that may vary per sample. The SPS and PPS are considered to be part of the information that rarely changes and is as such considered to be static. A set of flags are used to indicate if the parameter sets are expected to change in the stream. In such case, a sample grouping is defined that indicates the samples at which the parameter sets do change.

The Adaptation Parameter Set (APS) is considered to be dynamic information. The APS is used to carry the Adaptive Loop Filter (ALF) and Dynamic Range Adjustment (DRA) information. Usage of ALF or DRA to process samples of video data is signalled by a flag and referred APS identification in the slice header for ALF and in the picture parameters header for DRA.

12.3 Sample and configuration definition

12.3.1 Overview

An EVC sample contains an access unit as defined in ISO/IEC 23094-1:2020, subclause 3.1, and conforms to the sample format specified in subclause 4.2.3.

12.3.2 Canonical order and restrictions

The canonical stream format is an EVC elementary stream that satisfies the following conditions in addition to the general conditions in subclause 4.2.2:

- If all VCL NAL units for a single picture are stored in the same track, they shall be contained within the sample whose decoding time and composition time are those of the picture. Otherwise (i.e., when pieces of a coded picture are stored in samples of different tracks), each of these samples shall have the decoding time and composition time that are those of the coded picture.
- **Access unit delimiter NAL units:** The constraints about access unit delimiter NAL units defined in ISO/IEC 23094-1 shall be obeyed.
- **SPSs and PPSs:** An SPS or PPS to be used in a picture shall be present prior to the sample containing that picture or in the sample for that picture. The sample entry shall carry at least the SPS and PPS needed to process the first sample associated with it. Other SPSs and/or PPSs may be present in-band in the stream if allowed by the sample entry.
- **APSs:** EVC constraints for APS usage shall apply, for details refer to ISO/IEC 23094-1. The APSs may be stored in the sample entry and/or in the samples.
- **SEI messages:** SEI messages of declarative nature may be stored in the sample entry; there is no prescription about removing such SEI messages from the samples.
- **Filler data.** Video data is naturally represented as variable bit rate in the file format and should be filled for transmission if needed.

NOTE The removal or addition of Filler Data NAL units, start codes, or SEI messages can change the bitstream characteristics with respect to conformance with the HRD when operating the HRD in CBR mode as specified in ISO/IEC 23094-1:2020, Annex C.

12.3.3 Decoder configuration information: EVC decoder configuration record

12.3.3.1 Definition

This subclause specifies the decoder configuration information for ISO/IEC 23094-1 video content.

This record contains a version field. This version of the specification defines version 1 of this record. Incompatible changes to the record will be indicated by a change of version number. Readers shall not attempt to decode this record or the streams to which it applies if the version number is unrecognized.

Compatible extensions to this record will extend it and will not change the configuration version code. Readers should be prepared to ignore unrecognized data beyond the definition of the data they understand.

The values for `profile_idc`, `level_idc`, `toolset_idc_h`, `toolset_idc_l`, `chroma_format_idc`, `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, `bit_depth_luma_minus8`, and `bit_depth_chroma_minus8` shall be valid for all parameter sets that are activated when the stream described by this record is decoded (referred to as "all the parameter sets" in throughout this subclause).

Specifically, the following restrictions apply:

- The profile indication `profile_idc` shall indicate a profile to which the stream associated with this configuration record conforms.

NOTE 1 If the SPSs are marked with different profiles, then the stream could need examination to determine which profile, if any, the entire stream conforms to. If the entire stream is not examined, or the examination reveals that there is no profile to which the entire stream conforms, then the entire stream has to be split into two or more sub-streams with separate configuration records in which these rules can be met.

- The level indication `level_idc` shall indicate a level of capability equal to or greater than the highest level indicated in all the parameter sets of this configuration record.
- The `pic_width_in_luma_samples` and `pic_height_in_luma_samples` shall contain the highest values of all parameter sets of this configuration record.
- The toolset indication `toolset_idc_l` and `toolset_idc_h` shall signal all tools that are required to decode the stream associated with this configuration record. The tools flags shall comply with the conformance requirements as provided in ISO/IEC 23094-1:2020, Table A.6 and shall be identical with the `toolset_idc_h` and `toolset_idc_l` field that is signalled in the SPS.
- The value of `chroma_format_idc` in all the parameter sets shall be identical.
- The value of `bit_depth_luma_minus8` in all the parameter sets shall be identical.
- The value of `bit_depth_chroma_minus8` in all the parameter sets shall be identical.

Explicit indication is provided in the EVC Decoder Configuration Record about the chroma format and bit depth used by the EVC video elementary stream. Each type of such information shall be identical in all parameter sets, if present, in a single EVC configuration record. If two sequences differ in any type of such information, two different EVC sample entries shall be used.

There is a set of arrays to carry initialization NAL units. The NAL unit types are restricted to indicate SPS, PPS, APS, and SEI NAL units only. NAL unit types that are reserved in ISO/IEC 23094-1 and in this specification may acquire a definition in the future specification, and readers should ignore NAL units with unrecognized NAL unit types.

NOTE 2 This 'tolerant' behaviour is designed so that errors are not raised, allowing the possibility of backwards-compatible extensions to these arrays in future specifications.

It is recommended that the arrays in the decoder configuration record be in the order SPS, PPS, APS, SEI.

12.3.3.2 Syntax

```
aligned(8) class EVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(8) profile_idc;
    unsigned int(8) level_idc;
    unsigned int(32) toolset_idc_h;
    unsigned int(32) toolset_idc_l;
    unsigned int(2) chroma_format_idc;
    unsigned int(3) bit_depth_luma_minus8;
    unsigned int(3) bit_depth_chroma_minus8;
    unsigned int(16) pic_width_in_luma_samples;
    unsigned int(16) pic_height_in_luma_samples;
    unsigned int(6) reserved = '000000'b;
    unsigned int(2) LengthSizeMinusOne;
    unsigned int(8) num_of_arrays;
    for (j=0; j < num_of_arrays; j++) {
        unsigned int(1) array_completeness;
        bit(1) reserved = '0'b;
        unsigned int(6) NAL_unit_type;
        unsigned int(16) num_nalus;
        for (i=0; i < num_nalus; i++) {
            unsigned int(16) nal_unit_length;
            bit(8*nal_unit_length) nal_unit;
        }
    }
}
```

12.3.3.3 Semantics

`profile_idc`, `level_idc`, `toolset_idc_h`, `toolset_idc_l`, `chroma_format_idc`, `toolset_idc`, `bit_depth_luma_minus8` and `bit_depth_chroma_minus8` contain the matching values for the fields in the PPS for all parameter sets of the configuration record.

`pic_width_in_luma_samples` and `pic_height_in_luma_samples` contain the largest values for the fields in all SPS of this configuration record when the stream does not contain additional SPS, that are not included in the array of NAL units of this configuration record. They will contain the largest values for the fields in all SPS of this configuration record and all SPS in the stream when the stream contains additional SPS, that are not included in the array of NAL units of this configuration record. The value '0' shall be used if the largest value of these fields in the SPS for all the parameter sets in this record is not indicated through this field when the stream

does not contain additional SPS, that are not included in the array of NAL units of this configuration record or the value of these fields in the SPS in the stream has the value larger than the largest value of the field in this record when the stream contains additional SPS, that are not included in the array of NAL units of this configuration record.

`LengthSizeMinusOne` plus 1 indicates the length in bytes of the `NALUnitLength` field in a EVC video stream sample in the stream to which this configuration record applies. For example, a size of one byte is indicated with a value of 0. The value of this field shall be one of 0, 1, or 3 corresponding to a length encoded with 1, 2, or 4 bytes, respectively.

`numArrays` indicates the number of arrays of NAL units of the indicated type(s).

`array_completeness` when equal to 1 indicates that all NAL units of the given type are in the following array and none are present in the samples used for reconstructing the referenced stream when equal to 0 indicates that additional NAL units of the indicated type may be present in the samples used for reconstructing the referenced stream; the permitted values are constrained by the sample entry name.

`NAL_unit_type` indicates the type of the NAL units in the following array (which shall be all of that type); it takes a value as defined in ISO/IEC 23094-1; it is restricted to take one of the values indicating a SPS, PPS, APS, or SEI NAL unit.

`num_nalus` indicates the number of NAL units of the indicated type included in the configuration record for the stream to which this configuration record applies.

`nal_unit_length` indicates the length in bytes of the NAL unit.

`nal_unit` contains an SPS, PPS, APS or a SEI NAL unit, as specified in ISO/IEC 23094-1.

12.4 Derivation from ISO base media file format

12.4.1 EVC video stream definition: sample entry name and format

12.4.1.1 Definition

Sample Entry Type: 'evc1'

Container: Sample Table Box ('stbl')

Mandatory: An 'evc1' sample entry is mandatory for EVC bitstream

Quantity: One or more sample entries may be present

Box Type: 'evcC'

Container: EVC Sample Entry ('evc1')

Mandatory: Yes

Quantity: One

An EVC visual sample entry shall contain an `EVCCConfigurationBox`. This includes an `EVCDecoderConfigurationRecord`, as defined in subclause 12.3.3.

An optional `BitRateBox` may be present in the EVC visual sample entry to signal the bit rate information of the EVC video stream.

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.

When the sample entry name is 'evc1', the stream to which this sample entry applies shall be a compliant EVC stream as viewed by an EVC decoder operating under the configuration (including profile, level, and toolset) given in the `EVCCConfigurationBox`.

The 'evc1' sample entry allows for both storage of the parameter sets in the sample entry and in the stream. The `sps_in_stream`, `pps_in_stream`, and `aps_in_stream` when set to 0, indicate that the array of the NAL units of the corresponding type is complete.

If the sample of a 'evc1' track contains unspecified NAL unit types (NAL units that have `NalUnitType` values in the range of 56 to 62 as defined in ISO/IEC 23090-3), then the NAL units or NAL-unit-like structures having unspecified NAL unit types shall be discarded from the sample. The canonical order and restrictions as defined in subclause 12.3.2 shall apply after the removal of NAL units or NAL-unit-like structures specified above.

12.4.1.2 Syntax

```
class EVCConfigurationBox extends Box('evcC') {
    EVCDecoderConfigurationRecord() EVCConfig;
}

class EVCSampleEntry() extends VisualSampleEntry('evc1'){
    EVCConfigurationBox config;
    MPEG4ExtensionDescriptorsBox(); // optional
}
```

12.4.1.3 Semantics

`Compressorname` in the base class `VisualSampleEntry` indicates the name of the compressor used with the value "`\012EVC Coding`" being recommended (`\012` is 10, the length of the string in bytes).

`EVCDecoderConfigurationRecord` is defined in subclause 12.3.3.

12.4.2 Parameter sets

12.4.2.1 Overview

An EVC sample entry shall carry at least the SPS and PPS needed to process the first sample associated with it. If `sps_in_stream` and/or `pps_in_stream` is set to '1', additional SPSs and/or PPSs may be present in-band in the stream. Samples that carry one or more parameter sets shall belong to a sample group of type 'pss1'. The sample group description entry associated with the sample gives the type of parameter sets present in the sample (SPS and/or PPS and/or APS).

12.4.2.2 In-band parameter set indication sample group entry

12.4.2.2.1 Definition

Group Types: 'pss1'
 Container: Sample Group Description Box ('sgpd')
 Mandatory: No
 Quantity: Zero or more

A Parameter Set sample group identifies samples that contain one or more parameter set of type SPS, PPS, or APS. The `grouping_type_parameter` is not defined for the `SampleToGroupBox` with grouping type 'pss1'.

12.4.2.2.2 Syntax

```
class PSSampleGroupEntry() extends VisualSampleGroupEntry ('pss1')
{
    unsigned int(1) sps_present;
    unsigned int(1) pps_present;
    unsigned int(1) aps_present;
    bit(5) reserved = 0;
}
```

12.4.2.2.3 Semantics

`sps_present` indicates that a sample associated with this sample group description entry contains an SPS NAL unit if set to 1, or does not contain any SPS NAL unit if set to 0.

`pps_present` indicates that a sample associated with this sample group description entry contains an PPS NAL unit if set to 1, or does not contain any PPS NAL unit if set to 0.

`aps_present` indicates that a sample associated with this sample group description entry contains an APS NAL unit if set to 1, or does not contain any APS NAL unit if set to 0. When `sps_present` and `pps_present` are both equal to 0, `aps_present` shall be equal to 1.

12.4.3 Sync sample

An EVC sync sample shall contain VCL NAL units with NALUnitType IDR_NUT only; all SPS, PPS and APS required to process the bitstream from this sync sample on shall either be in this sync sample or in the associated sample description entry.

Table 15 indicates the mapping between EVC VCL NAL unit types, ISOBMFF sync sample status and SAP types as documented in ISO/IEC 14496-12.

Table 15 — Mapping of sync sample status and SAP types to NAL unit type

NAL Unit Type	ISOBMFF sync sample status	SAP type
IDR_NUT	true	1 or 2
IDR_NUT after which some pictures are not decodable	false	3

12.4.4 Definition of a sub-sample for EVC

For the use of the SubSampleInformationBox as defined in ISO/IEC 14496-12 in a EVC stream, a sub-sample is defined on the basis of the value of the `flags` field of the sub-sample information box as specified below. The presence of this box is optional; however, if present in a track containing EVC data, the `codec_specific_parameters` field in the box shall have the semantics defined here.

`flags` specifies the type of sub-sample information given in this box as follows:

- 0: NAL-unit-based sub-samples: A sub-sample contains one or more contiguous NAL units.
- 1: Tile-based sub-samples: A sub-sample contains the VCL NAL units with all CTUs of one tile together with any associated non-VCL NAL units, if any.
- 2: Slice-based sub-samples; A sub-sample contains one slice (i.e., one VCL NAL unit) and the associated non-VCL NAL units, if any.
- 3: Temporal ID-based sub-sample; A sub-sample contains all the VCL NAL units (of the sample) with the same TemporalId value, with any associated non-VCL NAL units if any.

Other values of flags are reserved.

The subsample_priority field shall be set to a value in accordance with the specification of this field in ISO/IEC 14496-12.

The discardable field shall be set to 1 only if this sample would still be decodable if this sub-sample is discarded (e.g. the sub-sample consists of an SEI NAL unit).

The length field, as defined in subclause 4.2.3, of each NAL unit in the sub-sample shall be included in the sub-sample size.

The codec_specific_parameters field of the SubSampleInformationBox is defined for EVC as follows:

```
if (flags == 1) {
    unsigned int(16) tile_col_idx;
    unsigned int(16) tile_row_idx;
}
else if (flags == 3) {
    unsigned int(3) temporal_id
    bit(29) reserved = 0;
}
else {
    bit(32) reserved = 0;
}
```

tile_col_idx for sub-samples based on tiles, this parameter indicates the 0-based index of the tile column that contains the tile of this sub-sample.

tile_row_idx for sub-samples based on tiles, this parameter indicates the 0-based index of the tile row that contains the tile of this sub-sample.

temporal_id for sub-samples based on Temporal ID, this parameter indicates the TemporalID of the VCL NAL units in the sub-sample.

12.5 EVC slice track

12.5.1 Overview

ISO/IEC 23094-1 EVC supports coding of rectangular regions within a picture as independently decodable units, so called slices. EVC slices can be encoded to have no coding dependencies with other

slices in the same picture but with other slices from the previous coded pictures so that they can be independently decoded. To support fast and efficient access of one or more slices for certain use cases requiring a subset of a picture instead of entire picture, a track containing data for one or more slice is defined in this section.

12.5.2 Implicit reconstruction of an EVC bitstream

If there are multiple tracks containing slice data for an access unit, the alignment of respective samples in tracks is performed based on the sample decoding times, i.e. using the time-to-sample table without considering edit lists. When reconstructing an access unit, the slices from samples having the same decoding time shall be placed into the access unit in increasing order of the smallest value of the tile ID of the first tile of the slice.

When the samples include SEI NAL units, then they shall be placed before any VCL NAL units of an access unit. When the sample is the first sample of a sequence of samples, the parameter set NAL units and SEI NAL units, if any, contained in the sample entry of the main slice track are placed before the VCL NAL units in an access unit. When the samples include NAL units with NALUnitType equal to FD_NUT, then they shall be placed after all VCL NAL units of an access unit.

When reconstructing an EVC bitstream, the reconstructed access units are placed into the EVC bitstream in increasing order of decoding time.

12.5.3 EVC slice component track

12.5.3.1 Definition

EVC slice component track is a track contains a subset of a picture composed of one or more EVC slices. EVC slice component track with sample entry type 'evs1' shall only contain VCL NAL units or non-VCL NAL but it shall not contain non-VCL NAL units whose NALUnitType is not equal to SPS_NUT, PPS_NUT or APS_NUT. EVC slice component track with sample entry type 'evs2' can contain any VCL NAL units or non-VCL NAL units. If a EVC slice component track contains any parameter set NAL units, then all parameter sets required to process VCL NAL units in the track shall be contained in either the same track or the EVC slice base track. This type of track shall have TrackReferenceBox with 'evcr' track reference type to link this type of track to the track containing non-VCL NAL units required to decode the VCL NAL units in this track.

12.5.3.2 Sample entry for EVC slice component track

12.5.3.2.1 Definition

Sample Entry Type: 'evs1' or 'evs2'

Container: Sample Table Box ('stbl')

Mandatory: An 'evs1' or 'evs2' sample entry is mandatory

Quantity: One or more sample entries may be present

Box Type: 'evsC'

Container: EVC Slice Component Sample Entry ('evs1' or 'evs2')

Mandatory: Yes

Quantity: One

An EVC visual sample entry shall contain an EVC Slice Component Track Configuration Box, as defined below when a track is an EVC slice component track. This includes an EVCSliceComonentTrackConfigurationRecord.

An optional BitRateBox may be present in the EVC visual sample entry to signal the bit rate information of the EVC slice data in this track.

Multiple sample entries may be used, as permitted by the ISO Base Media File Format specification, to indicate sections of video that use different configurations or parameter sets.

12.5.3.2.2 Syntax

```
class EVCSliceComponentTrackSampleEntry()
    extends VisualSampleEntry ('evs1' or 'evs2'){
    EVCSliceComponentTrackConfigurationBox config;
}

class EVCSliceComponentTrackConfigurationBox extends Box('evsC') {
    EVCSliceComponentTrackConfigurationRecord() config;
}

aligned(8) class EVCSliceComponentTrackConfigurationRecord {
    unsigned int(16) numOfSlice;
    for (i=0; i < numOfSlice; i++) {
        unsigned int(16) numOfTile;
        for (i=0; j< numOfTiles; j++) {
            unsigned int(16) tileID;
        }
    }
}
```

12.5.3.2.3 Semantics

numOfSlice indicates the number of slices contained in this track.

`numOfTile` indicates the number of tiles in a slice.

`tileID` indicates the tile ID.

12.5.4 EVC slice base track

12.5.4.1 Definition

EVC slice base track is a track contains non-VCL NAL units required to decode VCL NAL units in any EVC slice component tracks reference by 'evcr' track reference. This type of track can also include VCL NAL units of one or more slices belong to the same picture with the EVC slice component tracks referencing this track. The restrictions in the subclause 12.3.2 are applied to this type of track except that EVC slice base track can only include any SPS, PPS and APS which are not included in the array of NAL units of the configuration record of the track when they are required for decoding of VCL NAL units in the same track. The parameter sets required for decoding of any EVC slice component track shall only be carried in the configuration record of the EVC slice base track.

12.5.4.2 Sample entry for EVC slice base track

12.5.4.2.1 Definition

Sample Entry Type: 'evm1'

Container: Sample Table Box ('stbl')

Mandatory: An 'evm1' sample entry is mandatory for EVC slice base track

Quantity: One or more sample entries may be present

Box Type: 'evmC'

Container: EVC Slice Base Sample Entry ('evm1')

Mandatory: Yes

Quantity: One

An EVC visual sample entry shall contain an EVC Configuration Box as defined in the subclause 12.4.1.1 when a track is an EVC slice base track. This includes an `EVCDecoderConfigurationRecord` defined in the subclause 12.3.3.

An optional `BitRateBox` may be present in the EVC visual sample entry to signal the bit rate information of the EVC slice data in this track.

Multiple sample entries may be used, as permitted by the ISO Base Media File Format specification, to indicate sections of video that use different configurations or parameter sets.

Annex A (normative)

In-stream structures

A.1 General

Aggregators and Extractors are file format internal structures enabling efficient grouping of NAL units or extraction of NAL units from other tracks.

Aggregators and Extractors use 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 14496-10 or ISO/IEC 23008-2. These NAL-unit-like structures are seen as NAL units in the context of the sample structure. While accessing a sample, Aggregators shall be removed (leaving their contained or referenced NAL units) and Extractors shall be replaced by the data they reference. Aggregators and Extractors shall not be output by file parsers.

These structures use NAL unit types reserved for the application/transport layer by ISO/IEC 14496-10 or ISO/IEC 23008-2.

See Annex F for more information about use of “reserved”, “unspecified”, “not specified” and “registrant-defined” `nal_unit_type` values.

A.2 Aggregators

A.2.1 Definition

This subclause describes Aggregators, which enable NALU-map-group entries to be consistent and repetitive. (See Annex B).

Aggregators are used to group NAL units belonging to the same sample.

For storage of ISO/IEC 14496-10 video, the following rules apply:

- Aggregators use the same NAL unit header as SVC VCL NAL units, MVC VCL NAL units, MVC+D depth VCL NAL units, or 3D-AVC VCL NAL units, but with a different value of NAL unit type.
- If the sample entry contains the MVCD Configuration Box or the A3D Configuration Box, the NAL unit header of the aggregator follows the syntax of the NAL unit header for the NAL unit of `nal_unit_type` equal to 21. Otherwise, the NAL unit header of the aggregator follows the syntax of the NAL unit header for the NAL unit of `nal_unit_type` equal to 20.
- If the sample entry contains the SVC Configuration Box and the `svc_extension_flag` of the NAL unit syntax (specified in ISO/IEC 14496-10:2020, subclause 7.3.1) of an aggregator is equal to 1, the NAL unit header of SVC VCL NAL units is used for the aggregator.

- Otherwise, if the sample entry contains the MVCD Configuration Box or the A3D Configuration Box and additionally the `avc_3d_extension_flag` of the NAL unit syntax of an aggregator is equal to 1, the NAL unit header of 3D-AVC VCL NAL units is used for the aggregator.
- Otherwise, the NAL unit header of MVC and MVC+D depth VCL NAL units is used for the aggregator.

For storage of ISO/IEC 23008-2 video, Aggregators use the NAL unit header as defined in ISO/IEC 23008-2, which has the same syntax for plain HEVC and layered HEVC. An ISO/IEC 23008-2 aggregator shall not aggregate NAL units belonging to more than one picture unit (where picture unit is defined in ISO/IEC 23008-2).

Aggregators can both aggregate, by *inclusion*, NAL units within them (within the size indicated by their length) and also aggregate, by *reference*, NAL units that follow them (within the area indicated by the `additional_bytes` field within them). When the stream is scanned by a file reader that does not support aggregators, only the included NAL units are seen as “within” the aggregator. This permits such a file reader to skip a whole set of un-needed NAL units when they are aggregated by inclusion. This also permits such a file reader not to skip NAL units but let them remain in-stream when they are aggregated by reference.

Aggregators can be used to group base layer or base view NAL units. If these Aggregators are used in an 'avc1', 'hvc1', or 'hev1' track then an aggregator shall not use inclusion but reference of base layer or base view NAL units (the length of the Aggregator includes only its header and the NAL units referenced by the Aggregator are specified by `additional_bytes`).

When the aggregator is referenced by either an extractor with `data_length` equal to zero, or by a Map sample group, the aggregator is treated as aggregating both the included and referenced bytes.

An Aggregator may include or reference Extractors. An Extractor may extract from Aggregators. An aggregator shall not include or reference another aggregator directly; however, an aggregator may include or reference an extractor that references an aggregator.

An Aggregator shall not be empty, i.e., it shall include or reference to at least one NAL unit.

When scanning the stream:

- a) if the aggregator is unrecognized (e.g. by an AVC or HEVC reader or decoder) it is easily discarded with its included content;
- b) if the aggregator is not needed (i.e. it belongs to an undesired layer) it and its contents both by inclusion and reference are easily discarded (using its length and `additional_bytes` fields);
- c) if the aggregator is needed, its header is easily discarded and its contents retained.

An aggregator is stored within a sample like any other NAL unit.

All NAL units remain in decoding order within an aggregator.

A.2.2 Syntax

```
class aligned(8) Aggregator (AggregatorSize) {
    NALUnitHeader();
    unsigned int i = sizeof(NALUnitHeader());
    unsigned int((lengthSizeMinusOne+1)*8)
        additional_bytes;
    i += lengthSizeMinusOne+1;
    while (i<AggregatorSize) {
        unsigned int((lengthSizeMinusOne+1)*8)
            NALUnitLength;
        unsigned int(NALUnitLength*8) NALUnit;
        i += NALUnitLength+lengthSizeMinusOne+1;
    }
}
```

NOTE The syntax of Aggregators does not always follow the NAL unit syntax and the NAL unit constraints specified in ISO/IEC 14496-10 or ISO/IEC 23008-2. For example, there can be three continuous bytes equal to a value in the range of 0x000000 to 0x000010, inclusive. This specification disallows the presence of Aggregators in a video bitstream output from parsing a file, therefore formal non-compliance with the video specifications is immaterial as they will never be presented to a video decoder.

A.2.3 Semantics

The value of the variable `AggregatorSize` is equal to the size of the aggregator, and the function `sizeof(X)` returns the size of the field `X` in bytes.

`NALUnitHeader()`: the first four bytes of SVC, MVC, and MVC+D depth VCL NAL units, or the first three bytes of 3D-AVC NAL units, or the first two bytes of ISO/IEC 23008-2 NAL units.

`nal_unit_type` shall be set to 30 for ISO/IEC 14496-10 video and 48 for ISO/IEC 23008-2 video.

For an aggregator including or referencing SVC NAL units, the following shall apply.

`forbidden_zero_bit` and `reserved_three_2bits` shall be set as specified in ISO/IEC 14496-10.

Other fields (`nal_ref_idc`, `idr_flag`, `priority_id`, `no_inter_layer_pred_flag`, `dependency_id`, `quality_id`, `temporal_id`, `use_ref_base_pic_flag`, `discardable_flag`, and `output_flag`) shall be set as specified in A.4.

For an aggregator including or referencing MVC or MVC+D depth NAL units, the following shall apply.

`forbidden_zero_bit` and `reserved_one_bit` shall be set as specified in ISO/IEC 14496-10.

Other fields (`nal_ref_idc`, `non_idr_flag`, `priority_id`, `view_id`, `temporal_id`, `anchor_pic_flag`, and `inter_view_flag`) shall be set as specified in A.5.

For an aggregator including or referencing 3D-AVC NAL units, the following shall apply.

`forbidden_zero_bit` and `reserved_three_2bits` shall be set as specified in ISO/IEC 14496-10.

Other fields (`nal_ref_idc`, `view_idx`, `depth_flag`, `non_idr_flag`, `temporal_id`, `anchor_pic_flag`, and `inter_view_flag`) shall be set as specified in A.6.

For an aggregator including or referencing ISO/IEC 23008-2 NAL units, the following shall apply.

`forbidden_zero_bit` shall be set as specified in ISO/IEC 23008-2.

Other fields (`nuh_layer_id` and `nuh_temporal_id_plus1`) shall be set as specified in A.7.6.

`additional_bytes`: The number of bytes following this aggregator that should be considered as aggregated when this aggregator is referenced by an extractor with `data_length` equal to zero or Map sample group.

`NALUnitLength`: Specifies the size, in bytes, of the NAL unit following. The size of this field is specified with the `lengthSizeMinusOne` field.

`NALUnit`: a NAL unit as specified in ISO/IEC 14496-10 or ISO/IEC 23008-2, including the NAL unit header. The size of the NAL unit is specified by `NALUnitLength`.

A.3 Extractors for SVC, MVC, and MVD tracks

A.3.1 Definition

This subclause describes Extractors for SVC, MVC, or MVD tracks. Extractors enable compact formation of tracks that extract, by reference, NAL unit data from other tracks.

An Aggregator may include or reference Extractors. An Extractor may reference Aggregators. When an extractor is processed by a file reader that requires it, the extractor is logically replaced by the bytes it references. Those bytes shall not contain extractors; an extractor shall not reference, directly or indirectly, another extractor.

NOTE The track that is referenced can contain extractors even though the data that is referenced by the extractor cannot.

An extractor contains an instruction to extract data from another track, which is linked to the track in which the extractor resides, by means of a track reference of type '`scal`'.

The bytes copied shall be one of the following:

- a) One entire NAL unit; note that when an Aggregator is referenced, both the included and referenced bytes are copied
- b) More than one entire NAL unit

In both cases the bytes extracted start with a valid length field and a NAL unit header.

The bytes are copied only from the single identified sample in the track referenced through the indicated '`scal`' track reference. 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. The alignment is on decoding time, i.e. using the time-to-sample table only, followed by a counted offset in sample number. Extractors are a media-level concept and hence apply to the destination track before any edit list is considered. (However, one would normally expect that the edit lists in the two tracks would be identical).

A.3.2 Syntax

```
class aligned(8) Extractor () {
    NALUnitHeader();
    unsigned int(8) track_ref_index;
    signed int(8) sample_offset;
    unsigned int((lengthSizeMinusOne+1)*8)
        data_offset;
    unsigned int((lengthSizeMinusOne+1)*8)
        data_length;
}
```

NOTE The syntax of Extractors does not always follow the NAL unit syntax and the NAL unit constraints specified in ISO/IEC 14496-10. For example, there can be three continuous bytes equal to a value in the range of 0x000000 to 0x000010, inclusive. This specification disallows the presence of Extractors in a video bitstream output from parsing a file, therefore formal non-compliance with the video specifications is immaterial as they will never be presented to a video decoder.

A.3.3 Semantics

NALUnitHeader(): the first four bytes of SVC, MVC and MVC+D depth VCL NAL units, or the first three bytes of 3D-AVC NAL units.

nal_unit_type shall be set to 31 for ISO/IEC 14496-10 video.

For an extractor referencing SVC NAL units, the following shall apply.

forbidden_zero_bit and reserved_three_2bits shall be set as specified in ISO/IEC 14496-10.

Other fields (nal_ref_idc, idr_flag, priority_id, no_inter_layer_pred_flag, dependency_id, quality_id, temporal_id, use_ref_base_pic_flag, discardable_flag, and output_flag) shall be set as specified in A.4.

For an extractor referencing MVC or MVC+D depth NAL units, the following shall apply.

forbidden_zero_bit and reserved_one_bit shall be set as specified in ISO/IEC 14496-10.

Other fields (nal_ref_idc, non_idr_flag, priority_id, view_id, temporal_id, anchor_pic_flag, and inter_view_flag) shall be set as specified in A.5.

For an extractor referencing 3D-AVC NAL units, the following shall apply.

forbidden_zero_bit and reserved_three_2bits shall be set as specified in ISO/IEC 14496-10.

Other fields (nal_ref_idc, view_idx, depth_flag, non_idr_flag, temporal_id, anchor_pic_flag, and inter_view_flag) shall be set as specified in A.6.

track_ref_index specifies the index of the track reference of type 'scal' to use to find the track from which to extract data. The sample in that track from which data is extracted is temporally aligned or nearest preceding in the media decoding timeline, i.e. using the time-to-sample table only, adjusted by an offset specified by sample_offset with the sample containing the Extractor. The first track reference has the index value 1; the value 0 is reserved.

sample_offset gives the relative index of the sample in the linked track that shall be used as the source of information. Sample 0 (zero) is the sample with the same, or the closest preceding,

decoding time compared to the decoding time of the sample containing the extractor; sample 1 (one) is the next sample, sample -1 (minus 1) is the previous sample, and so on.

data_offset: The offset of the first byte within the reference sample to copy. If the extraction starts with the first byte of data in that sample, the offset takes the value 0. The offset shall reference the beginning of a NAL unit length field.

data_length: The number of bytes to copy. If this field takes the value 0, then the entire single referenced NAL unit is copied (i.e. the length to copy is taken from the length field referenced by the data offset, augmented by the additional_bytes field in the case of Aggregators). When $\text{data_offset} + \text{data_length}$ is greater than the size of the sample, the bytes from the byte pointed to by data_offset until the end of the sample, inclusive, are copied. Resolution of an extractor may result in a reconstructed payload for which there are fewer bytes than what is indicated in the NALUnitLength of the first NAL in that reconstructed payload. In such cases, readers shall assume that only a single NAL unit was reconstructed by the extractors, and shall rewrite the NALUnitLength of that NAL to the appropriate value (i.e. size of the reconstructed payload minus (LengthSizeMinusOne + 1)).

NOTE If the two tracks use different lengthSizeMinusOne values, then the extracted data will need re-formatting to conform to the destination track's length field size.

A.4 NAL unit header values for SVC

Both extractors and aggregators use NAL unit headers with the NAL unit header SVC extension. The NAL units extracted by an extractor or aggregated by an aggregator are all those NAL units that are referenced or included by recursively inspecting the contents of aggregators or extractors.

The fields `nal_ref_idc`, `idr_flag`, `priority_id`, `temporal_id`, `no_inter_layer_pred_flag`, `dependency_id`, `quality_id`, `temporal_id`, `use_ref_base_pic_flag`, `discardable_flag`, `output_flag`, `use_ref_base_pic_flag`, and `no_inter_layer_pred_flag` shall take the following values:

`nal_ref_idc` shall be set to the highest value of the field in all the extracted or aggregated NAL units.

`idr_flag` shall be set to the highest value of the field in all the extracted or aggregated NAL units.

`priority_id`, `temporal_id`, `dependency_id`, and `quality_id` shall be set to the lowest values of the fields, respectively, in all the extracted or aggregated NAL units.

`discardable_flag` shall be set to 1 if and only if all the extracted or aggregated NAL units have the `discardable_flag` set to 1, and set to 0 otherwise.

`output_flag` should be set to 1 if at least one of the aggregated or extracted NAL units has this flag set to 1, and otherwise set to 0.

`use_ref_base_pic_flag` shall be set to 1 if and only if at least one of the extracted or aggregated VCL NAL units have the `use_ref_base_pic_flag` set to 1, and set to 0 otherwise.

`no_inter_layer_pred_flag` shall be set to 1 if and only if all the extracted or aggregated VCL NAL units have the `no_inter_layer_pred_flag` set to 1, and set to 0 otherwise.

If the set of extracted or aggregated NAL units is empty, then each of these fields takes a value conformant with the mapped tier description.

NOTE 1 Aggregators can group NAL units with different scalability information.

NOTE 2 Aggregators can be used to group NAL units belonging to a level of scalability that is not signalled by the NAL unit header SVC extension (e.g. NAL units belonging to a region of interest). The description of such Aggregators can be done with the tier description and the NAL unit map groups. In this case more than one Aggregator with the same scalability information can occur in one sample.

NOTE 3 If multiple scalable tracks reference the same media data, then an aggregator might need to group NAL units with identical scalability information only. This ensures that the resulting pattern can be accessed by each of the tracks.

NOTE 4 If no NAL unit of a particular layer exists in an access unit then an empty Aggregator (in which the length of the Aggregator includes only the header, and additional_bytes is zero) can exist.

A.5 NAL unit header values for MVC and MVC+D depth NAL units

Both Aggregators and Extractors use the NAL unit header MVC extension. The NAL units extracted by an extractor or aggregated by an aggregator are all those NAL units that are referenced or included by recursively inspecting the contents of aggregators or extractors.

The fields `nal_ref_idc`, `non_idr_flag`, `priority_id`, `view_id`, `temporal_id`, `anchor_pic_flag`, and `inter_view_flag` shall take the following values:

- `nal_ref_idc` shall be set to the highest value of the field in all the aggregated or extracted NAL units.
- `non_idr_flag` shall be set to the lowest value of the field in all the aggregated or extracted NAL units.
- `priority_id` and `temporal_id` shall be set to the lowest values of the fields, respectively, in all the aggregated or extracted NAL units.
- `view_id` shall be set to the `view_id` value of the VCL NAL unit with the lowest view order index among all the aggregated or extracted VCL NAL units.
- `anchor_pic_flag` and `inter_view_flag` shall be set to the highest value of the fields, respectively, in all the aggregated or extracted VCL NAL units.

A.6 NAL unit header values for 3D-AVC NAL units

Both Aggregators and Extractors use the NAL unit header 3D-AVC extension specified in Annex J of ISO/IEC 14496-10:2020. The NAL units extracted by an extractor or aggregated by an aggregator are all those NAL units that are referenced or included by recursively inspecting the contents of aggregators or extractors. All the aggregated or extracted VCL NAL units shall have `nal_unit_type` equal to 21 and `avc_3d_extension_flag` equal to 1.

The fields `nal_ref_idc`, `view_idx`, `depth_flag`, `non_idr_flag`, `temporal_id`, `anchor_pic_flag`, and `inter_view_flag` shall take the following values:

- `nal_ref_idc` shall be set to the highest value of the field in all the aggregated or extracted NAL units.
- `view_idx` shall be set to the lowest view order index among all the aggregated or extracted VCL NAL units.
- `depth_flag` shall be set to the lowest `depth_flag` value among all the aggregated or extracted VCL NAL units.
- `non_idr_flag` shall be set to the lowest value of the field in all the aggregated or extracted VCL NAL units.

`temporal_id` shall be set to the lowest value of the field in all the aggregated or extracted VCL NAL units.

`anchor_pic_flag` and `inter_view_flag` shall be set to the highest value of the fields, respectively, in all the aggregated or extracted VCL NAL units.

A.7 Extractors for HEVC and L-HEVC tracks

A.7.1 Definition

This subclause describes extractors for HEVC and L-HEVC tracks. Extractors enable compact formation of tracks that extract NAL unit data by reference.

An extractor contains one or more constructors. The following constructors are specified:

- a) A sample constructor extracts, by reference, NAL unit data from a sample of another track.
- b) An in-line constructor includes NAL unit data.

An aggregator may include or reference extractors. An extractor may reference aggregators. When an extractor is processed by a file reader that requires it, the extractor is logically replaced by the bytes resulting when resolving the contained constructors in their appearance order.

The bytes referred to by a sample constructor shall not contain extractors; an extractor shall not reference, directly or indirectly, another extractor.

NOTE The track that is referenced can contain extractors even though the data that is referenced by the extractor cannot.

An extractor may contain one or more constructors for extracting data from the current track or from another track that is linked to the track in which the extractor resides by means of a track reference of type '`scal`'.

The bytes of a resolved extractor shall be one of the following:

- a) One entire NAL unit; note that when an Aggregator is referenced, both the included and referenced bytes are copied
- b) More than one entire NAL unit

In both cases the bytes of the resolved extractor start with a valid length field and a NAL unit header.

The bytes of a sample constructor are copied only from the single identified sample in the track referenced through the indicated '`scal`' track reference. The alignment is on decoding time, i.e. using the time-to-sample table only, followed by a counted offset in sample number. Extractors are a media-level concept and hence apply to the destination track before any edit list is considered. (However, one would normally expect that the edit lists in the two tracks would be identical).

A.7.2 Syntax

```
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() )
}
```

NOTE The syntax of Extractors does not always follow the NAL unit syntax and the NAL unit constraints specified in ISO/IEC 23008-2. For example, there can be three continuous bytes equal to a value in the range of 0x000000 to 0x000010, inclusive. This specification disallows the presence of Extractors in a video bitstream output from parsing a file, therefore formal non-compliance with the video specifications is immaterial as they will never be presented to a video decoder.

A.7.3 Semantics

NALUnitHeader(): The first two bytes of ISO/IEC 23008-2 NAL units.

nal_unit_type shall be set to 49 for ISO/IEC 23008-2 video. **forbidden_zero_bit** shall be set as specified in ISO/IEC 23008-2. Other fields (**nuh_layer_id** and **nuh_temporal_id_plus1**) shall be set as specified in A.7.6.

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.

EndOfNALUnit() is a function that returns 0 (false) when more data follows in this extractor; otherwise it returns 1 (true).

A.7.4 Sample constructor

A.7.4.1.1 Syntax

```
class aligned(8) SampleConstructor () {
    unsigned int(8) track_ref_index;
    signed int(8) sample_offset;
    unsigned int((lengthSizeMinusOne+1)*8)
        data_offset;
    unsigned int((lengthSizeMinusOne+1)*8)
        data_length;
}
```

A.7.4.1.2 Semantics

`track_ref_index`: as specified in A.3.3.

`sample_offset`: as specified in A.3.3.

`data_offset`: The offset of the first byte within the reference sample to copy. If the extraction starts with the first byte of data in that sample, the offset takes the value 0.

`data_length`: The number of bytes to copy. If this field takes the value 0, `data_offset` shall refer to the beginning of a NAL unit length field and the entire single referenced NAL unit is copied (i.e. the length to copy is taken from the length field referenced by `data_offset`, augmented by the `additional_bytes` field in the case of Aggregators). When `data_offset + data_length` is greater than the size of the sample, the bytes from the byte pointed to by `data_offset` until the end of the sample, inclusive, are copied, i.e. `data_length` is resolved as $(\text{sample_size} - \text{data_offset})$. Resolution of an extractor may result in a reconstructed payload for which there are fewer bytes than what is indicated in the `NALUnitLength` of the first NAL in that reconstructed payload. In such cases, readers shall assume that only a single NAL unit was reconstructed by the extractors, and shall rewrite the `NALUnitLength` of that NAL to the appropriate value (i.e. size of the reconstructed payload minus $(\text{LengthSizeMinusOne} + 1)$).

NOTE 1 If the two tracks use different `lengthSizeMinusOne` values, then the extracted data will need re-formatting to conform to the destination track's length field size.

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 In-line constructor

A.7.5.1.1 Syntax

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

A.7.5.1.2 Semantics

`length`: the number of bytes that belong to the `InlineConstructor` 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 in-line constructor.

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.