
**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*



Reference number
ISO/IEC 14496-15:2022(E)

Documents
Document Sharing Hub
© ISO/IEC 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

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.

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.

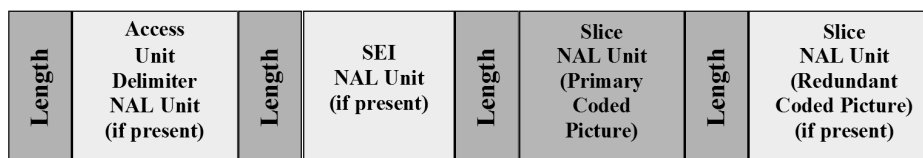


Figure 1 — Example structure of a sample

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 $(\text{flags} \& 1)$ 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:

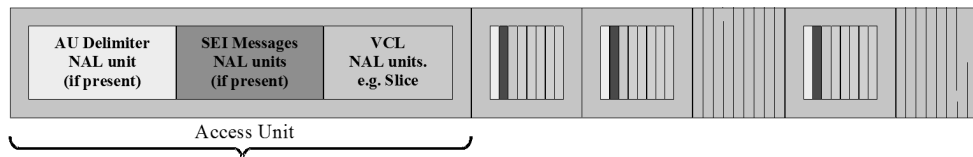
- **Video elementary stream with no parameter sets:** In this case, sequence and picture parameter set NAL units shall be stored in the sample entries of this track. Sequence and picture parameter set NAL units shall not be part of AVC samples within the stream itself.
- **Video elementary stream possibly 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.

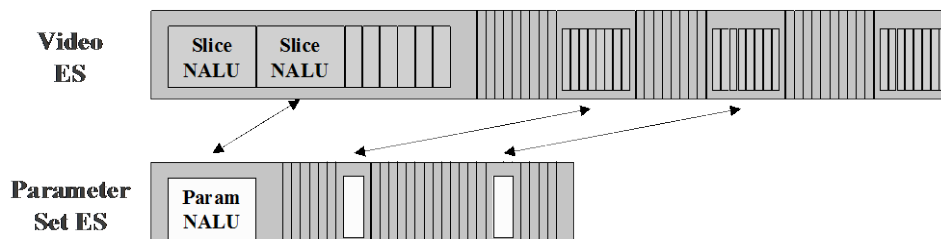
Table 1 — NAL Unit types in elementary Streams

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_partitionin g_rbsp()	Yes	Yes	No
2	Coded slice data partition A slice_data_partition_a_layer_rb sp()	Yes	Yes	No
3	Coded slice data partition B slice_data_partition_b_layer_rb sp()	Yes	Yes	No
4	Coded slice data partition C slice_data_partition_c_layer_rbs p()	Yes	Yes	No
5	Coded slice of an IDR picture slice_layer_without_partitionin g_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 `AVCDecoderConfigurationRecord`, 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 `AVCConfigurationBox`. 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 AVCConfigurationBox extends Box('avcC') {
    AVCDecoderConfigurationRecord() AVCConfig;
}

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

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

class AVC2SampleEntry() extends VisualSampleEntry (type) {
    // type is 'avc2' or 'avc4'
    AVCConfigurationBox 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.

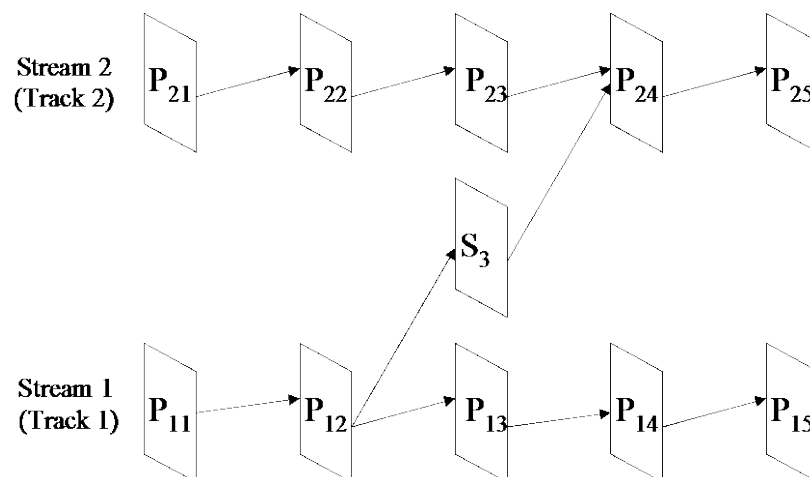


Figure 3 — Stream switching

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').