cellar
Internet-Draft
Intended status: Standards Track
Expires: 14 October 2026

S. Lhomme

M. Bunkus

D. Rice

12 April 2026

Matroska Media Container Codec Specifications
draft-ietf-cellar-codec-18

Abstract

This document defines the Matroska multimedia container codec mappings, including the codec ID, layout of data in a Block element and in an optional CodecPrivate element.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 October 2026.

Copyright Notice

Table of Contents

1. Introduction

Matroska is a multimedia container format. It stores interleaved and timestamped audiovisual data using various codecs. To interpret the codec data, a mapping between the way the data is stored in Matroska and how it is understood by such a codec is necessary.

This document defines this mapping for many commonly used codecs in Matroska.

2. Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Codec Mappings

A Codec Mapping is a set of attributes to identify, name, and contextualize the format and characteristics of encoded data that can be contained within Matroska Clusters.

Each TrackEntry used within Matroska MUST reference a defined Codec Mapping using the CodecID to identify and describe the format of the encoded data in its associated Clusters. This CodecID is an identifier, uniquely defined in the Matroska Codec IDs Registry Section 9.1, that represents the encoding stored within the Track. Certain encodings also require some form of codec initialization to provide its decoder with context and technical metadata.

The intention behind this list is not to list all existing audio and video codecs, but rather to list those codecs that are currently supported in Matroska and therefore need a well defined CodecID so that all developers supporting Matroska will use the same CodecID. If you feel a very important codec is missing, please tell us on our development mailing list (cellar at ietf.org).

3.1. Defining Matroska Codec Support

Support for a codec is defined in Matroska with the following values.

3.1.1. Codec ID

Each codec supported for storage in Matroska MUST have a unique CodecID. Each CodecID MUST be prefixed with the string from the following table according to the associated Track Type of the codec. All characters of a Codec ID Prefix MUST be capital letters (A-Z) except for the last character of a Codec ID Prefix which MUST be an underscore ("_").

Track Type	Codec ID Prefix
Video (1)	"V_"
Audio (2)	"A_"
Complex (3)	"O_"
Logo (16)	"L_"
Subtitle (17)	"S_"
Button (18)	"B_"
Control (32)	"C_"
Metadata (33)	"M_"

Table 1: Codec ID Prefix by Track Type

Each CodecID MUST include a Major Codec ID immediately following the Codec ID Prefix. A Major Codec ID MAY be followed by an OPTIONAL Codec ID Suffix to communicate a refinement of the Major Codec ID. If a Codec ID Suffix is used, then the CodecID MUST include a forward slash ("/") as a separator between the Major Codec ID and the Codec ID Suffix. The Major Codec ID MUST be composed of only capital letters (A-Z) and numbers (0-9). The Codec ID Suffix MUST be composed of only capital letters (A-Z), numbers (0-9), underscore ("_"), and forward slash ("/").

The following table provides examples of valid Codec IDs and their components:

Codec ID Prefix	Major Codec ID	Separator	Codec ID Suffix	Codec ID
A_	AAC	/	MPEG2/LC/ SBR	A_AAC/MPEG2/LC/ SBR
V_	MPEG4	/	ISO/ASP	V_MPEG4/ISO/ASP
V_	MPEG1			V_MPEG1

Table 2: Codec ID Components

3.1.2. Codec Name

Each encoding supported for storage in Matroska MUST have a Codec Name. The Codec Name provides a readable label for the encoding.

3.1.3. Description

An optional description for the encoding. This value is only intended for human consumption.

3.1.4. Initialization

Each encoding supported for storage in Matroska MUST have a defined Initialization. The Initialization MUST describe the storage of data necessary to initialize the decoder, which MUST be stored within the CodecPrivate element. When the Initialization is updated within a track, then that updated Initialization data MUST be written into the CodecState element of the first Cluster to require it. If the encoding does not require any form of Initialization, then none MUST be used to define the Initialization and the CodecPrivate element SHOULD NOT be written and MUST be ignored. If the TrackEntry contains a CodecPrivate element, its data MUST be provided to the decoder.

3.1.5. Codec BlockAdditions

Additional data that contextualizes or supplements a Block can be stored within the BlockAdditional element of a BlockMore element Section 5.1.3.5.2.1 of [RFC9559]. Each BlockAdditional is coupled with a BlockAddID that identifies the kind of data it contains.

A BlockAddID of 1 means the data in the BlockAdditional element are tied to the codec. This BlockAdditional data with a BlockAddID of 1 MAY be passed to the associated decoder alongside the Block content .

A codec definition MUST contain a "Codec BlockAdditions" section if the codec can use BlockAdditional data with a BlockAddID of 1.

The BlockAddID values are defined in Section 4.

3.1.6. Citation

Documentation of the associated normative and informative references for the codec is RECOMMENDED.

3.1.7. Superseded By

When a Superseded By is set, the superseding CodecID value MUST be used instead of the superseded CodecID.

Files MAY exist with the superseded CodecID and MAY be supported by Matroska Players.

3.2. Recommendations for the Creation of New Codec Mappings

Creators of new Codec Mappings to be used in the context of Matroska:

* MUST assume that all Codec Mappings they create might become standardized, public, commonly deployed, or usable across multiple implementations.

* MUST employ meaningful values for CodecID and Codec Name that are not already included in the Matroska Codec IDs Registry, and are not otherwise known or suspected to be in use, even if they are not already registered.

* MUST NOT prefix their CodecID with "X_" or similar constructs.

These recommendations are based on Section 3 of [RFC6648].

3.3. Video Codec Mappings

All codecs described in this section MUST have a TrackType (Section 5.1.4.1.3 of [RFC9559]) value of "1" for video. The track using these codecs MUST contain a Video element -- i.e., EBML Path \Segment\Tracks\TrackEntry\Video.

Most video codec contain meta information about the data they contain, like encoded width and height, chroma subsampling, etc. Whenever possible these information inside the codec SHOULD be extracted and repeated at the Matroska level with the appropriate element(s) inside the \Segment\Tracks\TrackEntry\Video and \Segment\Tracks\TrackEntry elements. These values MUST be valid for the whole Segment.

3.3.1. V_AV1

Codec ID: V_AV1

Codec Name: Alliance for Open Media AV1 Video codec

Description: Only one Sequence Header OBU, as defined in section 6.4 of [AV1], is supported per Matroska Segment. Each Block contains one Temporal Unit containing one or more OBUs. Each OBU stored in the Block MUST contain its header and its payload. The OBUs in the Block follow the Low Overhead Bitstream Format syntax. They MUST have the obu_has_size_field set to 1 except for the last OBU in the frame, for which obu_has_size_field MAY be set to 0, in which case it is assumed to fill the remainder of the frame. A SimpleBlock MUST NOT be marked as a Keyframe if it doesn’t contain a Frame OBU. A SimpleBlock MUST NOT be marked as a Keyframe if the first Frame OBU doesn’t have a frame_type of KEY_FRAME. A SimpleBlock MUST NOT be marked as a Keyframe if it doesn’t contains a Sequence Header OBU. A Block inside a BlockGroup MUST use ReferenceBlock elements if the first Frame OBU in the Block has a frame_type other than KEY_FRAME. A Block inside a BlockGroup MUST use ReferenceBlock elements if the Block doesn’t contain a Sequence Header OBU. A Block with frame_header_obu where the frame_type is INTRA_ONLY_FRAME MUST use a ReferenceBlock with a value of 0 to reference itself.

Initialization: The CodecPrivate consists of the AV1CodecConfigurationRecord described in section 2.3 of [AV1-ISOBMFF].

PixelWidth: MUST be max_frame_width_minus_1+1 of the Sequence Header OBU.

PixelHeight: MUST be max_frame_height_minus_1+1 of the Sequence Header OBU.

3.3.2. V_AVS2

Codec ID: V_AVS2

Codec Name: AVS2-P2/IEEE.1857.4

Description: Individual pictures of AVS2-P2 stored as described in the second part of [IEEE.1857-4].

Initialization: none

3.3.3. V_AVS3

Codec ID: V_AVS3

Codec Name: AVS3-P2/IEEE.1857.10

Description: Individual pictures of AVS3-P2 stored as described in the second part of [IEEE.1857-10].

Initialization: none

3.3.4. V_CAVS

Codec ID: V_CAVS

Codec Name: AVS1-P2, JiZhun profile

Description: Individual pictures of AVS1-P2 stored as described in [IEEE.1857-3].

Initialization: none

3.3.5. V_DIRAC

Codec ID: V_DIRAC

Codec Name: BBC Dirac

Description: A video codec developed by the BBC [Dirac]. The Intra-only version of Dirac, also known as Dirac Pro, resulted in SMPTE VC-2 [SMPTE.ST2042-1]. Each Matroska frame corresponds to a Sequence as defined in [Dirac].

Initialization: none

3.3.6. V_FFV1

Codec ID: V_FFV1

Codec Name: FF Video Codec 1

Description: FFV1 is a lossless intra-frame video encoding format designed to efficiently compress video data in a variety of pixel formats. Compared to uncompressed video, FFV1 offers storage compression, frame fixity, and self-description, which makes FFV1 useful as a preservation or intermediate video format. [RFC9043]

Initialization: For FFV1 versions 0 or 1, CodecPrivate SHOULD NOT be written. For FFV1 version 3 or greater, the CodecPrivate MUST contain the FFV1 Configuration Record structure, as defined in Section 4.3 of [RFC9043], and no other data.

3.3.7. V_JPEG2000

Codec ID: V_JPEG2000

Codec Name: JPEG 2000

Description: Each Matroska frame corresponds to a JPEG 2000 image, as defined in [JPEG2000].

Initialization: none

3.3.8. V_MJPEG

Codec ID: V_MJPEG

Codec Name: Motion JPEG

Description: Motion JPEG is a video compression format in which each video frame or interlaced field is compressed separately as a [JPEG] image.

Initialization: none

3.3.9. V_MPEGH/ISO/HEVC

Codec ID: V_MPEGH/ISO/HEVC

Codec Name: HEVC/H.265

Description: Individual pictures (which could be a frame, a field, or 2 fields having the same timestamp) of HEVC/H.265 stored as described in [ISO.14496-15].

Initialization: The CodecPrivate contains a HEVCDecoderConfigurationRecord structure, as defined in [ISO.14496-15].

3.3.10. V_MPEGI/ISO/VVC

Codec ID: V_MPEGI/ISO/VVC

Codec Name: VVC/H.266

Description: Individual pictures (which could be a frame, a field, or 2 fields having the same timestamp) of VVC/H.266 stored as described in [ISO.14496-15].

Initialization: The CodecPrivate contains a VVCDecoderConfigurationRecord structure, as defined in [ISO.14496-15].

3.3.11. V_MPEG1

Codec ID: V_MPEG1

Codec Name: MPEG 1

Description: Frames correspond to a Video Sequence as defined in [ISO.11172-2].

Initialization: none

3.3.12. V_MPEG2

Codec ID: V_MPEG2

Codec Name: MPEG 2

Description: Frames correspond to a Video Sequence as defined in [ITU-T.H.262].

Initialization: none

3.3.13. V_MPEG4/ISO/AVC

Codec ID: V_MPEG4/ISO/AVC

Codec Name: AVC/H.264

Description: Individual pictures (which could be a frame, a field, or 2 fields having the same timestamp) of AVC/H.264 stored as described in [ISO.14496-15].

Initialization: The CodecPrivate contains an AVCDecoderConfigurationRecord structure, as defined in [ISO.14496-15]. For legacy reasons, because Block Additional Mappings are preferred; see Section 4, the AVCDecoderConfigurationRecord structure MAY be followed by an extension block beginning with a 4-byte extension block size field in big-endian byte order which is the size of the extension block minus 4 (excluding the size of the extension block size field) and a 4-byte field corresponding to a BlockAddIDType of "mvcC" followed by a content corresponding to the content of BlockAddIDExtraData for mvcC; see Section 4.2.10.

3.3.14. V_MPEG4/ISO/AP

Codec ID: V_MPEG4/ISO/AP

Codec Name: MPEG4 ISO Advanced Profile

Description: Frames correspond to frames defined in [ISO.14496-2]. Stream was created via improved codec API (UCI) or transmuxed from MP4, not simply transmuxed from AVI. Note there are differences how b-frames are handled in these original streams, when being compared to a VfW created stream, as here there are no dummy frames inserted.

Initialization: none

3.3.15. V_MPEG4/ISO/ASP

Codec ID: V_MPEG4/ISO/ASP

Codec Name: MPEG4 ISO Advanced Simple Profile (DivX5, XviD)

Description: Frames correspond to frames defined in [ISO.14496-2]. Stream was created via improved codec API (UCI) or transmuxed from MP4, not simply transmuxed from AVI. Note there are differences how b-frames are handled in these original streams, when being compared to a VfW created stream, as here there are no dummy frames inserted.

Initialization: none

3.3.16. V_MPEG4/ISO/SP

Codec ID: V_MPEG4/ISO/SP

Codec Name: MPEG4 ISO Simple Profile (DivX4)

Description: Frames correspond to frames defined in [ISO.14496-2]. Stream was created via improved codec API (UCI) or even transmuxed from AVI (no b-frames in Simple Profile).

Initialization: none

3.3.17. V_MPEG4/MS/V3

Codec ID: V_MPEG4/MS/V3

Codec Name: Microsoft MPEG4 V3

Description: Microsoft MPEG4 V3 and derivatives, means DivX3, Angelpotion, SMR, etc.; stream was created using VfW codec or transmuxed from AVI; note that V1/V2 are covered by the "V_MS/VFW/FOURCC" CodecID Section 3.3.18.

Initialization: none

3.3.18. V_MS/VFW/FOURCC

Codec ID: V_MS/VFW/FOURCC

Codec Name: Microsoft Video Codec Manager (VCM)

Description: Video frames originating from Video For Windows, using the Microsoft Video Codec Manager codecs. This is a codec designed to be transmuxed back and forth from AVI sources.

Initialization: The CodecPrivate contains the VCM structure BITMAPINFOHEADER including the extra private bytes, as defined in [BITMAPINFOHEADER]. The data are stored in little-endian format (like on IA32 machines).

3.3.19. V_QUICKTIME

Codec ID: V_QUICKTIME

Codec Name: Video taken from QuickTime files

Description: Several codecs as stored in QuickTime (e.g., Sorenson or Cinepak).

Initialization: The CodecPrivate contains all additional data that is stored in the ’stsd’ (sample description) atom in the QuickTime file *after* the mandatory video descriptor structure (starting with the size and FourCC fields). For an explanation of the QuickTime file format read [QTFF].

3.3.20. V_PRORES

Codec ID: V_PRORES

Codec Name: Apple ProRes

Initialization: The CodecPrivate contains the FourCC as found in MP4 movies:

* ap4x: ProRes 4444 XQ
* ap4h: ProRes 4444
* apch: ProRes 422 High Quality
* apcn: ProRes 422 Standard Definition
* apcs: ProRes 422 LT
* apco: ProRes 422 Proxy
* aprh: ProRes RAW High Quality
* aprn: ProRes RAW Standard Definition

ProRes is defined as [SMPTE.RDD36].

3.3.21. V_REAL/RV10

Codec ID: V_REAL/RV10

Codec Name: RealVideo 1.0 aka RealVideo 5

Description: Individual slices from the Real container are combined into a single frame.

Initialization: The CodecPrivate contains a real_video_props_t structure in big-endian byte order as found in [librmff].

3.3.22. V_REAL/RV20

Codec ID: V_REAL/RV20

Codec Name: RealVideo G2 and RealVideo G2+SVT

Description: Individual slices from the Real container are combined into a single frame.

Initialization: The CodecPrivate contains a real_video_props_t structure in big-endian byte order as found in [librmff].

3.3.23. V_REAL/RV30

Codec ID: V_REAL/RV30

Codec Name: RealVideo 8

Description: Individual slices from the Real container are combined into a single frame.

Initialization: The CodecPrivate contains a real_video_props_t structure in big-endian byte order as found in [librmff].

3.3.24. V_REAL/RV40

Codec ID: V_REAL/RV40

Codec Name: rv40 : RealVideo 9

Description: Individual slices from the Real container are combined into a single frame.

Initialization: The CodecPrivate contains a real_video_props_t structure in big-endian byte order as found in [librmff].

3.3.25. V_THEORA

Codec ID: V_THEORA

Codec Name: Theora

Description: Frames correspond to a Theora Frame as defined in [Theora].

Initialization: The CodecPrivate contains the first three Theora packets in order. The lengths of the packets precedes them. The actual layout is:

* Byte 1: number of distinct packets #p minus one inside the CodecPrivate block. This MUST be "2" for current (as of 2016-07-08) Theora headers.

* Bytes 2..n: lengths of the first #p packets, coded in Xiph-style lacing. The length of the last packet is the length of the CodecPrivate block minus the lengths coded in these bytes minus one.

* Bytes n+1..: The Theora identification header, followed by the commend header followed by the codec setup header. Those are described in the [Theora].

3.3.26. V_UNCOMPRESSED

Codec ID: V_UNCOMPRESSED

Codec Name: Video, raw uncompressed video frames

Description: The codec doesn't use any form of compression. All the relevant fields of the TrackEntry\Video element MUST be filled to play this content correctly. In addition the packing of RGB, YUV, etc. pixels MUST be declared with a TrackEntry\Video\UncompressedFourCC element with values defined in Section 5.1.4.1.28.15 of [RFC9559].

Initialization: none

3.3.27. V_VC1

Codec ID: V_VC1

Codec Name: VC-1

Description: Frames correspond to VC-1 samples as defined in [SMPTE.RP2025-2007].

Initialization: The CodecPrivate contains a VC1DecSpecStruc structure, as defined in [SMPTE.RP2025-2007].

3.3.28. V_VP8

Codec ID: V_VP8

Codec Name: VP8 Codec format

Description: VP8 is an open and royalty free video compression format developed by Google and created by On2 Technologies as a successor to VP7. [RFC6386]

Codec BlockAdditions: A single-channel encoding of an alpha channel MAY be stored in BlockAdditions. The BlockAddID of the BlockMore containing these data MUST be 1.

Initialization: none

3.3.29. V_VP9

Codec ID: V_VP9

Codec Name: VP9 Codec format

Description: VP9 is an open and royalty free video compression format developed by Google as a successor to VP8. [VP9]

Codec BlockAdditions: A single-channel encoding of an alpha channel MAY be stored in BlockAdditions. The BlockAddID of the BlockMore containing these data MUST be 1.

Initialization: Encoders are strongly encouraged to provide a CodecPrivate that contains a list of specific VP9 codec features as described in the "VP9 Codec Feature Metadata" section of [WebMContainer]. This piece of data helps to select a decoder on playback, but as many muxers don’t provide the CodecPrivate for "V_VP9" it is not a hard requirement. It is possible for the decoder to reconstruct the "VP9 Codec Feature Metadata" from the first frame in case the CodecPrivate is not present.

Note that the format differs from the VPCodecConfigurationRecord structure, as defined in [VP-ISOBMFF].

3.4. Audio Codec Mappings

All codecs described in this section MUST have a TrackType (Section 5.1.4.1.3 of [RFC9559]) value of "2" for audio. The track using these codecs MUST contain an Audio element -- i.e., EBML Path \Segment\Tracks\TrackEntry\Audio.

Most audio codec contain meta information about the data they contain, like encoded sampling frequency, channel count, etc. Whenever possible these information inside the codec SHOULD be extracted and repeated at the Matroska level with the appropriate element(s) inside the \Segment\Tracks\TrackEntry\Audio and \Segment\Tracks\TrackEntry elements. These values MUST be valid for the whole Segment.

3.4.1. A_AAC

Codec ID: A_AAC

Codec Name: Advanced Audio Coding (AAC)

Description: Individual frames of AAC raw_data_block(), stored as defined in subpart 4 of [ISO.14496-3].

Initialization: The CodecPrivate contains an AudioSpecificConfig structure, as defined in [ISO.14496-3].

3.4.2. A_AAC/MPEG2/LC

Codec ID: A_AAC/MPEG2/LC

Codec Name: Low Complexity

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.3. A_AAC/MPEG2/LC/SBR

Codec ID: A_AAC/MPEG2/LC/SBR

Codec Name: Low Complexity with Spectral Band Replication

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.4. A_AAC/MPEG2/MAIN

Codec ID: A_AAC/MPEG2/MAIN

Codec Name: MPEG2 Main Profile

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.5. A_AAC/MPEG2/SSR

Codec ID: A_AAC/MPEG2/SSR

Codec Name: Scalable Sampling Rate

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.6. A_AAC/MPEG4/LC

Codec ID: A_AAC/MPEG4/LC

Codec Name: Low Complexity

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.7. A_AAC/MPEG4/LC/SBR

Codec ID: A_AAC/MPEG4/LC/SBR

Codec Name: Low Complexity with Spectral Band Replication

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.8. A_AAC/MPEG4/LTP

Codec ID: A_AAC/MPEG4/LTP

Codec Name: Long Term Prediction

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.9. A_AAC/MPEG4/MAIN

Codec ID: A_AAC/MPEG4/MAIN

Codec Name: MPEG4 Main Profile

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.10. A_AAC/MPEG4/SSR

Codec ID: A_AAC/MPEG4/SSR

Codec Name: Scalable Sampling Rate

Description: The audio stream is stripped from ADTS headers and normal Matroska frame based muxing scheme is applied.

Initialization: none

Superseded By: A_AAC (Section 3.4.1)

3.4.11. A_AC3

Codec ID: A_AC3

Codec Name: Dolby Digital / AC-3

Description: Individual frames of AC-3 syncframe() stored as described in [ATSC.A52] or [ETSI.TS102-366] when the value of the bsid field defined in Section 5.4.2.1 of [ATSC.A52] or Section 4.4.2.1 of [ETSI.TS102-366] is 10 or below. Channel number have to be read from the corresponding audio element

Initialization: none

3.4.12. A_AC3/BSID9

Codec ID: A_AC3/BSID9

Codec Name: Dolby Digital / AC-3

Description: Individual frames of AC-3 syncframe() stored as described in [ATSC.A52] or [ETSI.TS102-366] when the value of the bsid field defined in Section 5.4.2.1 of [ATSC.A52] or Section 4.4.2.1 of [ETSI.TS102-366] is 9. Note that the value 9 in the bsid field is not standard but it is defacto used for dividing the sampling rate defined in Section 5.4.1.3 of [ATSC.A52] or Section 4.4.2.1 of [ETSI.TS102-366] by 2.

Using this Codec ID is NOT RECOMMENDED as many Matroska Players don’t support it. The generic A_AC3 Codec ID SHOULD be used instead as it supports a bsid of 9 as well.

Initialization: none

3.4.13. A_AC3/BSID10

Codec ID: A_AC3/BSID10

Codec Name: Dolby Digital / AC-3

Description: Individual frames of AC-3 syncframe() stored as described in [ATSC.A52] or [ETSI.TS102-366] when the value of the bsid field defined in Section 5.4.2.1 of [ATSC.A52] or Section 4.4.2.1 of [ETSI.TS102-366] is 10. Note that the value 10 in the bsid field is not standard but it is defacto used for dividing the sampling rate defined in Section 5.4.1.3 of [ATSC.A52] or Section 4.4.2.1 of [ETSI.TS102-366] by 4.

Using this Codec ID is NOT RECOMMENDED as many Matroska Players don’t support it. The generic A_AC3 Codec ID SHOULD be used instead as it supports a bsid of 10 as well.

Initialization: none

3.4.14. A_ALAC

Codec ID: A_ALAC

Codec Name: ALAC (Apple Lossless Audio Codec)

Initialization: The CodecPrivate contains ALAC’s magic cookie (both the codec specific configuration as well as the optional channel layout information). Its format is described in the "Magic Cookie" defined in [ALAC].

3.4.15. A_ATRAC/AT1

Codec ID: A_ATRAC/AT1

Codec Name: Sony ATRAC1 Codec

Description: The original ATRAC codec by Sony, mainly used in MiniDisc platforms. The core technical details on ATRAC1 can be found in [AtracAES]. An example encoder/decoder can be found at [atracdenc].

Initialization: None

3.4.16. A_DTS

Codec ID: A_DTS

Codec Name: Digital Theatre System

Description: Supports DTS, DTS-ES, DTS-96/26, DTS-HD High Resolution Audio and DTS-HD Master Audio. It corresponds to the base codec defined in [ETSI.TS102-114].

Initialization: none

3.4.17. A_DTS/EXPRESS

Codec ID: A_DTS/EXPRESS

Codec Name: Digital Theatre System Express

Description: DTS Express (a.k.a. LBR) audio streams. It corresponds to the LBR extension of the DTS codec defined in section 9 of [ETSI.TS102-114].

Initialization: none

3.4.18. A_DTS/LOSSLESS

Codec ID: A_DTS/LOSSLESS

Codec Name: Digital Theatre System Lossless

Description: DTS Lossless audio that does not have a core substream. It corresponds to the Lossless extension (XLL) of the DTS codec defined in section 8 of [ETSI.TS102-114].

Initialization: none

3.4.19. A_EAC3

Codec ID: A_EAC3

Codec Name: Dolby Digital Plus / E-AC-3

Description: Individual frames of E-AC-3 syncframe() stored as described in [ATSC.A52] or [ETSI.TS102-366] when the value of the bsid field defined in Annex E Section 2.1 of [ATSC.A52] or Section E.1.3.1.6 of [ETSI.TS102-366] is 11, 12, 13, 14, 15 or 16.

Initialization: none

3.4.20. A_FLAC

Codec ID: A_FLAC

Codec Name: FLAC (Free Lossless Audio Codec)

Description: The mapping of the FLAC framing and CodecPrivate is described in Section 10.2 of [RFC9639].

Initialization: All FLAC data before the first audio frame; see Section 10.2 of [RFC9639].

3.4.21. A_MLP

Codec ID: A_MLP

Codec Name: Meridian Lossless Packing / MLP

Description: A lossless audio codec used in DVD-Audio discs. The format is similar to Dolby TrueHD (Section 3.4.39) but with fewer channels.

Initialization: none

3.4.22. A_MPEG/L1

Codec ID: A_MPEG/L1

Codec Name: MPEG Audio 1, 2 Layer I

Description: Frames correspond to Audio Frames of a Layer I bitstream as defined in [ISO.11172-3].

Initialization: none

3.4.23. A_MPEG/L2

Codec ID: A_MPEG/L2

Codec Name: MPEG Audio 1, 2 Layer II

Description: Frames correspond to Audio Frames of a Layer II bitstream as defined in [ISO.11172-3].

Initialization: none

3.4.24. A_MPEG/L3

Codec ID: A_MPEG/L3

Codec Name: MPEG Audio 1, 2, 2.5 Layer III

Description: Frames correspond to Audio Frames of a Layer III bitstream as defined in [ISO.11172-3].

Initialization: none

3.4.25. A_MS/ACM

Codec ID: A_MS/ACM

Codec Name: Microsoft Audio Codec Manager (ACM)

Description: The data are stored in little-endian format (like on IA32 machines).

Initialization: The CodecPrivate contains the [WAVEFORMATEX] structure including the extra format information bytes. The structure is stored without packing or padding bytes. A WORD corresponds to a signed 2 octets integer, DWORD corresponds to a signed 4 octets integer. The extra format information are appended after the WAVEFORMATEX octets.

3.4.26. A_REAL/14_4

Codec ID: A_REAL/14_4

Codec Name: Real Audio 1

Initialization: The CodecPrivate contains either the "real_audio_v4_props_t" or the "real_audio_v5_props_t" structure (differentiated by their "version" field; big-endian byte order) as found in [librmff].

3.4.27. A_REAL/28_8

Codec ID: A_REAL/28_8

Codec Name: Real Audio 2

Initialization: The CodecPrivate contains either the "real_audio_v4_props_t" or the "real_audio_v5_props_t" structure (differentiated by their "version" field; big-endian byte order) as found in [librmff].

3.4.28. A_REAL/ATRC

Codec ID: A_REAL/ATRC

Codec Name: Sony Atrac3 Codec

Initialization: The CodecPrivate contains either the "real_audio_v4_props_t" or the "real_audio_v5_props_t" structure (differentiated by their "version" field; big-endian byte order) as found in [librmff].

3.4.29. A_REAL/COOK

Codec ID: A_REAL/COOK

Codec Name: Real Audio Cook Codec (codename: Gecko)

Initialization: The CodecPrivate contains either the "real_audio_v4_props_t" or the "real_audio_v5_props_t" structure (differentiated by their "version" field; big-endian byte order) as found in [librmff].

3.4.30. A_REAL/RALF

Codec ID: A_REAL/RALF

Codec Name: Real Audio Lossless Format

Initialization: The CodecPrivate contains either the "real_audio_v4_props_t" or the "real_audio_v5_props_t" structure (differentiated by their "version" field; big-endian byte order) as found in [librmff].

3.4.31. A_REAL/SIPR

Codec ID: A_REAL/SIPR

Codec Name: Sipro Voice Codec

Initialization: The CodecPrivate contains either the "real_audio_v4_props_t" or the "real_audio_v5_props_t" structure (differentiated by their "version" field; big-endian byte order) as found in [librmff].

3.4.32. A_OPUS

Codec ID: A_OPUS

Codec Name: Opus interactive speech and audio codec

Description: The OPUS audio codec defined by [RFC6716] using a similar encapsulation as the Ogg Encapsulation [RFC7845].

Initialization: The track CodecPrivate MUST be present and contain the Identification Header defined in Section 5.1 of [RFC7845].

Channels: The track Channels element value MUST be the "Output Channel Count" value of the Identification Header.

SamplingFrequency: The track SamplingFrequency element value MUST be the "Input Sample Rate" value of the Identification Header.

CodecDelay: The track CodecDelay element MUST be present and set to the "Pre-skip" value of the Identification Header translated to Matroska Ticks. The "Pre-skip" value is in samples at 48,000 Hz. The formula to get the CodecDelay is:

CodecDelay = pre-skip * 1,000,000,000 / 48,000.

SeekPreRoll: The track SeekPreRoll element SHOULD be present and set to 80,000,000 -- 80 ms in Matroska Ticks -- in order to ensure that the output audio is correct by the time it reaches the seek target.

3.4.33. A_PCM/FLOAT/IEEE

Codec ID: A_PCM/FLOAT/IEEE

Codec Name: Floating-Point, IEEE compatible

Description: The audio bit depth MUST be read and set from the BitDepth element (32 bits in most cases). The floats are stored as defined in [IEEE.754] and in little-endian order.

Initialization: none

3.4.34. A_PCM/INT/BIG

Codec ID: A_PCM/INT/BIG

Codec Name: PCM Integer Big Endian

Description: The audio bit depth MUST be read and set from the BitDepth element. Audio samples MUST be considered as signed values, unless the audio bit depth is 8 which MUST be interpreted as unsigned values.

Initialization: none

3.4.35. A_PCM/INT/LIT

Codec ID: A_PCM/INT/LIT

Codec Name: PCM Integer Little Endian

Description: The audio bit depth MUST be read and set from the BitDepth element. Audio samples MUST be considered as signed values, unless the audio bit depth is 8 which MUST be interpreted as unsigned values.

Initialization: none

3.4.36. A_QUICKTIME

Codec ID: A_QUICKTIME

Codec Name: Audio taken from QuickTime files

Description: Several codecs as stored in QuickTime (e.g., QDesign Music v1 or v2).

Initialization: The CodecPrivate contains all additional data that is stored in the ’stsd’ (sample description) atom in the QuickTime file *after* the mandatory sound descriptor structure (starting with the size and FourCC fields). For an explanation of the QuickTime file format read [QTFF].

3.4.37. A_QUICKTIME/QDMC

Codec ID: A_QUICKTIME/QDMC

Codec Name: QDesign Music

Description:

Initialization: The CodecPrivate contains all additional data that is stored in the ’stsd’ (sample description) atom in the QuickTime file *after* the mandatory sound descriptor structure (starting with the size and FourCC fields). For an explanation of the QuickTime file format read [QTFF].

Superseded By: A_QUICKTIME (Section 3.4.36)

3.4.38. A_QUICKTIME/QDM2

Codec ID: A_QUICKTIME/QDM2

Codec Name: QDesign Music v2

Description:

Initialization: The CodecPrivate contains all additional data that is stored in the ’stsd’ (sample description) atom in the QuickTime file *after* the mandatory sound descriptor structure (starting with the size and FourCC fields). For an explanation of the QuickTime file format read [QTFF].

Superseded By: A_QUICKTIME (Section 3.4.36)

3.4.39. A_TRUEHD

Codec ID: A_TRUEHD

Codec Name: Dolby TrueHD

Description: Lossless audio codec from Dolby. Each Matroska frame corresponds to a single Access Unit as defined in [TRUEHD].

Initialization: none

3.4.40. A_TTA1

Codec ID: A_TTA1

Codec Name: The True Audio lossless audio compressor

Description: The format is described in [TTA]. Each frame is kept intact, including the CRC32. The header and seektable are dropped. SamplingFrequency, Channels and BitDepth are used in the TrackEntry.

Initialization: The CodecPrivate contains the TTA Header Structure, as defined in [TTA].

3.4.41. A_VORBIS

Codec ID: A_VORBIS

Codec Name: Vorbis

Initialization: The CodecPrivate contains the first three Vorbis packet in order. The lengths of the packets precedes them. The actual layout is:

* Byte 1: number of distinct packets #p minus one inside the CodecPrivate block. This MUST be "2" for current (as of 2016-07-08) Vorbis headers.

* Bytes 2..n: lengths of the first #p packets, coded in Xiph-style lacing. The length of the last packet is the length of the CodecPrivate block minus the lengths coded in these bytes minus one.

* Bytes n+1..: The "Identification Header" as defined in Section 4.2.2 of [VORBIS], followed by the "Comment Header" as defined in Section 5 of [VORBIS], followed by the "Setup Header" as defined in Section 4.2.4 of [VORBIS].

3.4.42. A_WAVPACK4

Codec ID: A_WAVPACK4

Codec Name: WavPack lossless audio compressor

Description: The WavPack packets consist of a block defined in [WAVPACK] with a WavpackHeader header. For multichannel (> 2 channels) a frame consists of many packets. For more details, check the WavPack muxing description Section 5.1.

Codec BlockAdditions: For hybrid A_WAVPACK4 encodings (that include a lossy encoding with a supplemental correction to produce a lossless encoding), the correction part is stored in BlockAdditional. The BlockAddID of the BlockMore containing these data MUST be 1.

Initialization: The CodecPrivate contains the version 16-bit integer from the WavpackHeader of [WAVPACK] stored in little-endian.

3.5. Subtitle Codec Mappings

All codecs described in this section MUST have a TrackType (Section 5.1.4.1.3 of [RFC9559]) value of "17" for subtitles.

Subtitle codec often contain meta information about the data they contain, like expected output dimension, language, etc. Whenever possible these information inside the codec SHOULD be extracted and repeated at the Matroska level with the appropriate element(s) inside the \Segment\Tracks\TrackEntry\Video and \Segment\Tracks\TrackEntry elements. These values MUST be valid for the whole Segment.

3.5.1. S_ARIBSUB

Codec ID: S_ARIBSUB

Codec Name: ARIB STD-B24 subtitles

Description: This is the textual subtitle format used in the ISDB/ ARIB broadcasting standard. For more information see Section 6.8 on ARIB (ISDB) subtitles.

Initialization: The CodecPrivate data are defined in Section 6.8.1.

3.5.2. S_DVBSUB

Codec ID: S_DVBSUB

Codec Name: Digital Video Broadcasting (DVB) subtitles

Description: This is the graphical subtitle format used in the Digital Video Broadcasting standard. For more information see Section 6.7 on Digital Video Broadcasting (DVB).

Initialization: The CodecPrivate data are defined in Section 6.7.1.

3.5.3. S_HDMV/PGS

Codec ID: S_HDMV/PGS

Codec Name: HDMV presentation graphics subtitles (PGS)

Description: This is the graphical subtitle format used on Blu-rays. For more information, see Section 6.6 on HDMV text presentation.

Initialization: none

3.5.4. S_HDMV/TEXTST

Codec ID: S_HDMV/TEXTST

Codec Name: HDMV text subtitles

Description: This is the textual subtitle format used on Blu-rays. For more information, see Section 6.5 on HDMV graphics presentation.

Initialization: The CodecPrivate data are defined in Section 6.6.1.

3.5.5. S_KATE

Codec ID: S_KATE

Codec Name: Karaoke And Text Encapsulation

Description: A subtitle format developed for ogg. The mapping for Matroska is described on the "Matroska mapping" section of [OggKate].

The codec MAY use embedded fonts from attachments, as defined in Section 21.2 of [RFC9559], in that case the TrackEntry MUST contain an AttachmentLink element.

Initialization: Kate headers are stored in the CodecPrivate as xiph-laced packets. The length of the last packet isn’t encoded, it is deduced from the sizes of the other packets and the total size of the CodecPrivate.

3.5.6. S_IMAGE/BMP

Codec ID: S_IMAGE/BMP

Codec Name: Bitmap

Description: Basic image based subtitle format; The subtitles are stored as images, like in the DVD [DVD-Video]. The timestamp in the block header of Matroska indicates the start display time, the duration is set with the BlockDuration element. The full data for the subtitle bitmap is stored in the Block’s data section.

Initialization: none

3.5.7. S_TEXT/ASS

Codec ID: S_TEXT/ASS

Codec Name: Advanced SubStation Alpha Format

Description: Each event is stored in its own Block. For more information see Section 6.3 on SSA/ASS.

This codec ID MUST be used when "ScriptType: v4.00+" or "[V4+ Styles]" sections are found in the original SSA script.

The codec MAY use embedded fonts from attachments, as defined in Section 21.2 of [RFC9559], in that case the TrackEntry MUST contain an AttachmentLink element.

The codec MAY also be found with the Codec ID S_ASS in legacy media containers, but using that value is NOT RECOMMENDED.

Initialization: The "[Script Info]" and "[V4 Styles]" sections are stored in the CodecPrivate.

3.5.8. S_TEXT/ASCII

Codec ID: S_TEXT/ASCII

Codec Name: ASCII Plain Text

Description: Basic text subtitles with only ASCII characters allowed.

Initialization: none

3.5.9. S_TEXT/SSA

Codec ID: S_TEXT/SSA

Codec Name: SubStation Alpha Format

Description: Each event is stored in its own Block. For more information see Section 6.3 on SSA/ASS.

This codec ID MUST NOT be used when "ScriptType: v4.00+" or "[V4+ Styles]" sections are found in the original SSA script.

The codec MAY use embedded fonts from attachments, as defined in Section 21.2 of [RFC9559], in that case the TrackEntry MUST contain an AttachmentLink element.

The codec MAY also be found with the Codec ID S_SSA, but using that value is NOT RECOMMENDED.

Initialization: The "[Script Info]" and "[V4+ Styles]" sections are stored in the CodecPrivate.

3.5.10. S_TEXT/USF

Codec ID: S_TEXT/USF

Codec Name: Universal Subtitle Format

Description: An XML based subtitle format. Each BlockGroup contains XML data from a "subtitle" XML element as defined in section 3.4 of [USF], without the "subtitle" element itself and with the start, stop duration mapped to the BlockGroup timestamp and BlockDuration element. The "image" XML elements are turned into Matroska attachments and replaced in the stream with their attachment filename.

The codec MAY use embedded fonts from attachments, as defined in Section 21.2 of [RFC9559], in that case the TrackEntry MUST contain an AttachmentLink element.

Initialization: The CodecPrivate element MAY be present. If present it MAY contain "metadata", "styles" and "effects" XML elements usable in the whole stream inside a parent "USFSubtitles" XML parent element, similar to the "USFSubtitles" element of a standalone USF file but without the "subtitles" XML element.

3.5.11. S_TEXT/UTF8

Codec ID: S_TEXT/UTF8

Codec Name: UTF-8 Plain Text

Description: Basic text subtitles. For more information see Section 6 on Subtitles.

Initialization: none

3.5.12. S_TEXT/WEBVTT

Codec ID: S_TEXT/WEBVTT

Codec Name: Web Video Text Tracks Format (WebVTT)

Description: Advanced text subtitles defined by [WebVTT]. For more information see Section 6.4.

Initialization: The CodecPrivate contains the WebVTT file body up to the first WebVTT cue block.

Codec BlockAdditions: Intermediate non-Cue Blocks SHOULD be stored in BlockAdditions. The BlockAddID of the BlockMore containing these data MUST be 1.

3.5.13. S_VOBSUB

Codec ID: S_VOBSUB

Codec Name: VobSub subtitles

Description: Uses data from [VobSub] files. The data represent subtitle data used on DVDs [DVD-Video]. VobSubs consist of two files, the .idx containing information, and the .sub, containing the actual data. Only version 7 and newer of VobSubs files are supported.

The line of the .idx file beginning with "id:" MUST be transformed into the appropriate Matroska track language element.

For each line of the .idx file containing a "timestamp:" and "filepos:" data is read from the appropriate position in the .sub file. This data consists of a MPEG program stream which in turn contains SPU packets. The MPEG program stream data is discarded, and each SPU packet is put into one Matroska frame.

Initialization: The CodecPrivate contains the "palette:" and "size:" lines from the .idx file. Other lines from the .idx file not containing empty lines, comments, or starting with "alt:"; "langidx:", "id:", or "timestamp:" MAY be added in the CodecPrivate data for preservation.

3.6. Button Codec Mappings

All codecs described in this section MUST have a TrackType (Section 5.1.4.1.3 of [RFC9559]) value of "18" for buttons.

3.6.1. B_VOBBTN

Codec ID: B_VOBBTN

Codec Name: VobBtn Buttons

Description: Based on MPEG/VOB PCI packets. The frame contains a header consisting of the string "butonDVD" followed by the width and height in pixels (16-bit unsigned integer each) and 4 reserved bytes. The rest is a full PCI packet described in [DVD-Info.PCI].

4. Block Addition Mappings

This section describes the various types of BlockAdditionMapping that can be stored in Matroska. These help the player interpret the multiple BlockAdditions that can be added to each Matroska BlockGroup. More details can be found in section Section 7.

4.1. Defining Block Addition Mappings

Support for a Block Addition mapping is defined in Matroska with the following values.

4.1.1. Block Type Identifier

Each BlockAdditionMapping supported in Matroska MUST have a unique BlockAddIDType. It MUST be defined for each Block Addition Mapping.

4.1.2. Block Type Name

Each BlockAdditionMapping supported in Matroska MAY have a BlockAddIDName. The BlockAddIDName provides a readable label for the encoding.

4.1.3. Description

An optional description for the encoding. This value is only intended for human consumption.

4.2. Initial Block Addition Mappings

4.2.1. Use BlockAddIDValue

Block type identifier: 0

Block type name: "BlockAddIDValue"

Description: This value indicates that the actual type is stored in BlockAddIDValue instead. This value is used when it is important to have a strong compatibility with players or derived formats not supporting BlockAdditionMapping but using BlockAdditions with an unknown BlockAddIDValue, and SHOULD NOT be used if it is possible to use another value.

4.2.2. Opaque Data

Block type identifier: 1

Block type name: "Opaque data"

Description: the BlockAdditional data is interpreted as opaque additional data passed to the codec with the Block data. The usage of these BlockAdditional data is defined in the "Codec BlockAdditions" section of the codec; see Section 3.1.5.

4.2.3. ITU T.35 Metadata

Block type identifier: 4

Block type name: "ITU T.35 metadata"

Description: the BlockAdditional data is interpreted as ITU T.35 metadata, as defined by [ITU-T.35] terminal codes. BlockAddIDValue MUST be 4.

HDR10+ dynamic metadata can be stored as ITU T.35 terminal codes as defined in Table 8 of [CTA.861-4].

4.2.4. SMPTE ST 12-1 Timecode

Block type identifier: 121

Block type name: "SMPTE ST 12-1 timecode"

Description: the BlockAdditional data is defined in Section 7.1.

4.2.5. avcE

Block type identifier: 0x61766345

Block type name: Dolby Vision enhancement-layer AVC configuration

Description: the BlockAddIDExtraData data is interpreted as the Dolby Vision enhancement-layer AVC configuration box as described in [DolbyVision-ISOBMFF]. This extension MUST NOT be used if CodecID is not V_MPEG4/ISO/AVC.

4.2.6. hvcE

Block type identifier: 0x68766345

Block type name: "Dolby Vision enhancement-layer HEVC configuration"

Description: the BlockAddIDExtraData data is interpreted as the Dolby Vision enhancement-layer HEVC configuration as described in [DolbyVision-ISOBMFF]. This extension MUST NOT be used if CodecID is not V_MPEGH/ISO/HEVC.

4.2.7. dvcC

Block type identifier: 0x64766343

Block type name: "Dolby Vision configuration dvcC"

Description: the BlockAddIDExtraData data is interpreted as DOVIDecoderConfigurationRecord structure, as defined in [DolbyVision-ISOBMFF], for Dolby Vision profiles 0 to 7 inclusive.

4.2.8. dvvC

Block type identifier: 0x64767643

Block type name: "Dolby Vision configuration dvvC"

Description: the BlockAddIDExtraData data is interpreted as DOVIDecoderConfigurationRecord structure, as defined in [DolbyVision-ISOBMFF], for Dolby Vision profiles 8 to 10 inclusive and 20.

4.2.9. dvwC

Block type identifier: 0x64767743

Block type name: "Dolby Vision configuration dvwC"

Description: the BlockAddIDExtraData data is interpreted as DOVIDecoderConfigurationRecord structure, as defined in [DolbyVision-ISOBMFF], for Dolby Vision profiles 11 to 19 inclusive.

4.2.10. mvcC

Block type identifier: 0x6D766343

Block type name: "MVC configuration"

Description: the BlockAddIDExtraData data is interpreted as MVCDecoderConfigurationRecord structure, as defined in [ISO.14496-15]. This extension MUST NOT be used if CodecID is not V_MPEG4/ISO/AVC.

5. Audio Codecs

5.1. WavPack

WavPack is an audio codec primarily designed for lossless audio, but it can also be used as a lossy codec.

[WAVPACK] stores each data in variable length frames. That means each frame can have a different number of samples.

Each WavPack block starts with a WavpackHeader header as defined in [WAVPACK], stored in little-endian.

To save space and avoid redundant information in Matroska some data from the WavpackHeader header are removed, when saved in Matroska. All the data from the WavpackHeader are kept in little-endian.

The CodecPrivate contains the version 16-bit integer from the WavpackHeader of [WAVPACK] stored in little-endian.

Depending on the number of audio channels and whether the hybrid mode is kept or not, the storage of WavPack blocks in Matroska differ.

5.1.1. Lossless And Lossy Storage

For multichannel files (more than 2 channels, like for 5.1), a frame consists of multiple WavPack blocks. The first one having the INITIAL_BLOCK (bit 11) flag set and the last one the FINAL_BLOCK (bit 12) flag set. For a mono or stereo file, both flags are set in each WavPack block.

5.1.1.1. Mono/Stereo

A Block or SimpleBlock frame contains the following header with the some fields taken from the WavpackHeader of a single WavPack block followed by the data of that WavPack block.

```
{
  uint32_t block_samples; // # samples in this block
  uint32_t flags;         // various flags for id and decoding
  uint32_t crc;           // crc for actual decoded data
}
[ block data ]
```

5.1.1.2. Multichannel

For multichannel files, a WavPack file uses multiple WavPack block to store all channels of a frame. The WavPack blocks for each channels of a frame are stored consecutively into a Matroska Block or SimpleBlock.

Each WavPack block is preceded by a header. The header for the first WavPack block is similar to the mono/stereo one (Section 5.1.1.1) with the addition of a "blocksize" field, which is the size of the first WavPack block minus the WavpackHeader size. The header for the

following WavPack blocks use the "flags" and "crc" of the WavpackHeader of each respective WavPack block, followed with the size of each respective WavPack block minus the WavpackHeader size.

```
{
  uint32_t block_samples; // # samples in this block
  uint32_t flags;         // various flags for id and decoding
  uint32_t crc;           // crc for actual decoded data
  uint32_t blocksize;     // size of the data to follow
}
[ block data # 1 ]
{
  uint32_t flags;         // various flags for id and decoding
  uint32_t crc;           // crc for actual decoded data
  uint32_t blocksize;     // size of the data to follow
}
[ block data # 2 ]
{
  uint32_t flags;         // various flags for id and decoding
  uint32_t crc;           // crc for actual decoded data
  uint32_t blocksize;     // size of the data to follow
}
[ block data # 3 ]
...
```

5.1.2. Hybrid Storage

WavPack has a hybrid mode that splits the audio frames between lossy and correction packets. Adding both gives a lossless version of the original audio. It is possible to only store the lossy part in Matroska or both together. Storing only the lossy part is equivalent to the format described in Section 5.1.1. This section explains how to store all hybrid data in Matroska.

Hybrid WavPack is encoded in 2 files. The first one has a lossy part and the second file has the correction part to reconstruct the original audio losslessly.

Each WavPack block in the correction file corresponds to a WavPack block in the lossy file with the same number of samples, that’s also true for a multichannel file. This means that if a frame is made of 4 WavPack blocks, the correction file will have 4 WavPack blocks in the corresponding frame. The header of the correction WavPack block is exactly the same as in the lossy WavPack block, except for the CRC.

In Matroska, the correction part is stored as an additional data available to the Block (see Section 7). This way a file could be remuxed and not keep the Block Additional data and still be usable as a lossy WavPack file. The Block data of the lossy file are stored exactly the same as for lossy storage defined in Section 5.1.1.

A BlockAdditionMapping MUST be used for hybrid WavPack TrackEntry’.

The BlockAddIDType of that BlockAdditionMapping MUST be set to 1 for hybrid WavPack, corresponding to Opaque data; see Section 4.2.2.

Each WavPack frame is stored in a BlockGroup that MUST have at least a BlockMore to hold the correction data.

The BlockAddID of that BlockMore MUST be 1, i.e., the default value.

5.1.2.1. Mono/Stereo

The BlockAdditional element of the correction data BlockMore contains the following header with the "crc" field from the WavpackHeader of the WavPack block of the correction file matching the WavPack block of the lossy frame used to fill the Block data, followed by the data of that correction file WavPack block.

```
{
  uint32_t crc;           // crc for actual decoded data
}
[ correction block data ]
```

5.1.2.2. Multichannel

The BlockAdditional element of the correction data BlockMore contains the following header with the data from the each WavpackHeader of the WavPack block of the correction file matching the WavPack block in the lossy file used to fill the Block data, followed by the data of the correction file WavPack block.

```
{
  uint32_t crc;           // crc for actual decoded data
  uint32_t blocksize;     // size of the data to follow
}
[ correction block data # 1 ]
{
  uint32_t crc;           // crc for actual decoded data
  uint32_t blocksize;     // size of the data to follow
}
[ correction block data # 2 ]
{
  uint32_t crc;           // crc for actual decoded data
  uint32_t blocksize;     // size of the data to follow
}
[ correction block data # 3 ]
...
```

6. Subtitles

Here is a list of guidelines for storing subtitles in Matroska:

* As a general rule of thumb for all codecs, information that is global to an entire stream SHOULD be stored in the CodecPrivate element, although not all codec mappings are designed this way.

* As subtitles usually come with a start and stop timestamps or a start timestamp and a duration, SimpleBlock is usually not used as it doesn't allow storing the BlockDuration. One exception would be if the subtitle track has a DefaultDuration which doesn't require a BlockDuration.

* Start and stop timestamps that are used in a timestamps original storage format SHOULD be removed when being placed in Matroska as they could interfere if the file is edited afterwards. Instead, the Block's timestamp and BlockDuration SHOULD be used to say when the timestamp is displayed.

* Because a "subtitle" stream is actually just an overlay stream, anything with a transparency layer could be used, including video.

6.1. Images Subtitles

A common image format imported into Matroska is the VobSub subtitle format. This subtitle type is generated by exporting the subtitles from a DVD [DVD-Video].

If the subtitle version in the .IDX file is less than v7, the content has to be remuxed as the S_VOBSUB CodecID only supports version 7 and newer of VobSubs files; see Section 3.5.13. One way to remux the subtitles is to use the SubResync utility from VobSub 2.23 (or MPC) into v7 format. Generally any newly created subs will be in v7 format.

The .IFO file will not be used at all.

If there is more than one subtitle stream in the VobSub set, each stream is separated into separate tracks for storage in Matroska. E.g. the VobSub file contains streams for both English and German subtitles. Then the resulting Matroska file will contain multiple tracks, and language information can be mapped to Matroska's language tags and dropped from the streams.

The .IDX file is reformatted (see below) and placed in the CodecPrivate.

Each .BMP will be stored in its own Block. The Timestamp will be stored in the Block timestamp and the duration will be stored in the Default Duration.

Here is an example .IDX file:

```
 # VobSub index file, v7 (do not modify this line!)
 #
 # To repair desynchronization, you can insert gaps this way:
 # (it usually happens after vob id changes)
 #
 # delay: [sign]hh:mm:ss:ms
 #
 # Where:
 # [sign]: +, - (optional)
 # hh: hours (0 <= hh)
 # mm/ss: minutes/seconds (0 <= mm/ss <= 59)
 # ms: milliseconds (0 <= ms <= 999)
 #
 # You can also modify timestamps or delete a few subs you don't
 # like. Just make sure they stay in increasing order.

 # Settings

 # Original frame size
 size: 720x480

 # Origin, relative to the upper-left corner, can be overloaded by
 # alignment
```

```
    org: 0, 0

    # Image scaling (hor,ver), origin is at the upper-left corner or at
    # the alignment coord (x, y)
    scale: 100%, 100%

    # Alpha blending
    alpha: 100%

    # Smoothing for very blocky images (use OLD for no filtering)
    smooth: OFF

    # In millisecs
    fadein/out: 50, 50

    # Force subtitle placement relative to (org.x, org.y)
    align: OFF at LEFT TOP

    # For correcting non-progressive desync. (in millisecs or
    # hh:mm:ss:ms)
    # Note: Not effective in DirectVobSub, use "delay: ... " instead.
    time offset: 0

    # ON: displays only forced subtitles, OFF: shows everything
    forced subs: OFF

    # The original palette of the DVD
    palette: 000000, 7e7e7e, fbff8b, cb86f1, 7f74b8, e23f06, 0a48ea, \
    b3d65a, 6b92f1, 87f087, c02081, f8d0f4, e3c411, 382201, e8840b, \
    fdfdfd

    # Custom colors (transp idxs and the four colors)
    custom colors: OFF, tridx: 0000, colors: 000000, 000000, 000000, \
    000000

    # Language index in use
    langidx: 0

    # English
    id: en, index: 0
    # Uncomment next line to activate alternative name in DirectVobSub /
    # Windows Media Player 6.x
    # alt: English
    # Vob/Cell ID: 1, 1 (PTS: 0)
    timestamp: 00:00:01:101, filepos: 000000000
    timestamp: 00:00:08:708, filepos: 000001000
```

First, lines beginning with "#" are removed. These are comments to make text file editing easier, and as this is not a text file, they aren't needed.

Next remove the "langidx" and "id" lines. These are used to differentiate the subtitle streams and define the language. As the streams will be stored separately anyway, there is no need to differentiate them here. Also, the language setting will be stored in the Matroska tags, so there is no need to store it here.

Finally, the "timestamp" will be used to set the Block's timestamp. Once it is set there, there is no need for it to be stored here. Also, as it may interfere if the file is edited, it SHOULD NOT be stored here and it MUST NOT be used by the decoder.

Once all of these items are removed, the data to store in the CodecPrivate SHOULD look like this:

```
size: 720x480
org: 0, 0
scale: 100%, 100%
alpha: 100%
smooth: OFF
fadein/out: 50, 50
align: OFF at LEFT TOP
time offset: 0
forced subs: OFF
palette: 000000, 7e7e7e, fbff8b, cb86f1, 7f74b8, e23f06, 0a48ea, \
b3d65a, 6b92f1, 87f087, c02081, f8d0f4, e3c411, 382201, e8840b, \
fdfdfd
custom colors: OFF, tridx: 0000, colors: 000000, 000000, 000000, \
000000
```

There SHOULD also be two Blocks containing one image each with the timestamps "00:00:01:101" and "00:00:08:708".

6.2. SRT Subtitles

SRT is perhaps the most basic of all subtitle formats.

It consists of four parts, all in text:

1. A number indicating which subtitle it is in the sequence.

2. The time that the subtitle appears on the screen, and then disappears.

3. The subtitle itself.

4. A blank line indicating the start of a new subtitle.

When placing SRT in Matroska, part 3 is converted to UTF-8 (S_TEXT/UTF8) and placed in the data portion of the Block. Part 2 is used to set the timestamp of the Block, and BlockDuration element. Nothing else is used.

Here is an example SRT file:

```
1
00:02:17,440 --> 00:02:20,375
Senator, we’re making
our final approach into Coruscant.

2
00:02:20,476 --> 00:02:22,501
Very good, Lieutenant.
```

In this example, the text "Senator, we’re making our final approach into Coruscant." would be converted into UTF-8 and placed in the Block. The timestamp of the block would be set to "00:02:17,440". And the BlockDuration element would be set to "00:00:02,935".

The same is repeated for the next subtitle.

Because there are no general settings for SRT, the CodecPrivate is left blank.

6.3. SSA/ASS Subtitles

SSA stands for Sub Station Alpha. It’s the file format used by the popular subtitle editor SubStation Alpha. It allows you to do some advanced display features, like positioning, karaoke, or style managements...

For detailed information on SSA/ASS, see the SSA specs [SSA]. It includes an SSA specs description and the advanced features added by ASS format (standing for Advanced SSA). Because SSA and ASS are so similar, they are treated the same here.

Like SRT, this format is text based with a particular syntax.

A file consists of 4 or 5 parts, declared similar to an INI file.

The first, "[Script Info]" contains some information about the subtitle file, such as its title, who created it, type of script and "PlayResY", which is very important, because everything in your script (font size, positioning) is scaled by it. Sub Station Alpha

uses your desktops Y resolution to write this value, so if a friend with a large monitor and a high screen resolution gives you an edited script, you can mess everything up by saving the script in SSA with your low-resolution monitor.

The second, "[V4 Styles]" or "[V4+ Styles]", is a list of style definitions. A style describes how a text will look on the screen. It defines font, font size, primary/.../outline color, position, alignment, etc.

For example, this:

```
Format: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, \
TertiaryColour, BackColour, Bold, Italic, BorderStyle, Outline, \
Shadow, Alignment, MarginL, MarginR, MarginV, AlphaLevel, Encoding
Style: Wolf main,Wolf_Rain,56,15724527,15724527,15724527,4144959,0,\
0,1,1,2,2,5,5,30,0,0
```

The third, "[Events]", is the list of text you want to display at the right timing. You can specify some attributes here, such as the style to use for this event (MUST be defined in the list), the position of the text (Left, Right, Vertical Margin), or some effect. The Name is used by the translator to know who said this sentence. Timing is in h:mm:ss.cc (centisec).

```
Format: Marked, Start, End, Style, Name, MarginL, MarginR, MarginV, \
Effect, Text
Dialogue: Marked=0,0:02:40.65,0:02:41.79,Wolf main,Cher,0000,0000,\
0000,,Et les enregistrements de ses ondes delta ?
Dialogue: Marked=0,0:02:42.42,0:02:44.15,Wolf main,autre,0000,0000,\
0000,,Toujours rien.
```

"[Pictures]" or "[Fonts]" part can be found in some SSA files. These parts contain UUE-encoded pictures/font. These features are only used by Sub Station Alpha -- i.e., no filter (Vobsub/Avery Lee Subtiler filter) uses them.

Now, how are they stored in Matroska?

* All text is converted to UTF-8

* All the headers, "[Script Info]" and the "[V4 Styles]"/"[V4+ Styles]" list, are stored in CodecPrivate.

* Start & End field are used to set TimeStamp and the BlockDuration element. the data stored is:

* Events are stored in the Block in this order: ReadOrder, Layer, Style, Name, MarginL, MarginR, MarginV, Effect, Text (Layer comes from ASS specs ... it's empty for SSA.) "ReadOrder field is needed for the decoder to be able to reorder the streamed samples as they were placed originally in the file."

Here is an example of an SSA file.

```
[Script Info]
; This is a Sub Station Alpha v4 script.
Title: Wolf's rain 2
Original Script: Anime-spirit Ishin-francais
Original Translation: Coolman
Original Editing: Spikewolfwood
Original Timing: Lord_alucard
Original Script Checking: Spikewolfwood
ScriptType: v4.00
Collisions: Normal
PlayResY: 1024
PlayDepth: 0
Wav: 0, 128697,D:\Alex\Anime\- Fansub -\- TAFF -\WR_-_02_Wav.wav
Wav: 0, 120692,H:\team truc\WR_-_02.wav
Wav: 0, 116504,E:\sub\wolf's_rain\WOLF'S RAIN 02.wav
LastWav: 3
Timer: 100,0000

[V4 Styles]
Format: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, \
TertiaryColour, BackColour, Bold, Italic, BorderStyle, Outline, \
Shadow, Alignment, MarginL, MarginR, MarginV, AlphaLevel, Encoding
Style: Default,Arial,20,65535,65535,65535,-2147483640,-1,0,1,3,0,2,\
30,30,30,0,0
Style: Titre_episode,Akbar,140,15724527,65535,65535,986895,-1,0,1,1,\
0,3,30,30,30,0,0
Style: Wolf main,Wolf_Rain,56,15724527,15724527,15724527,4144959,0,\
0,1,1,2,2,5,5,30,0,0

[Events]
Format: Marked, Start, End, Style, Name, MarginL, MarginR, MarginV, \
Effect, Text
Dialogue: Marked=0,0:02:40.65,0:02:41.79,Wolf main,Cher,0000,0000,\
0000,,Et les enregistrements de ses ondes delta ?
Dialogue: Marked=0,0:02:42.42,0:02:44.15,Wolf main,autre,0000,0000,\
0000,,Toujours rien.
```

Here is what would be placed into the CodecPrivate element.

```
[Script Info]
; This is a Sub Station Alpha v4 script.
Title: Wolf’s rain 2
Original Script: Anime-spirit Ishin-francais
Original Translation: Coolman
Original Editing: Spikewolfwood
Original Timing: Lord_alucard
Original Script Checking: Spikewolfwood
ScriptType: v4.00
Collisions: Normal
PlayResY: 1024
PlayDepth: 0
Wav: 0, 128697,D:\Alex\Anime\- Fansub -\- TAFF -\WR_-_02_Wav.wav
Wav: 0, 120692,H:\team truc\WR_-_02.wav
Wav: 0, 116504,E:\sub\wolf’s_rain\WOLF’S RAIN 02.wav
LastWav: 3
Timer: 100,0000

[V4 Styles]
Format: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, \
TertiaryColour, BackColour, Bold, Italic, BorderStyle, Outline, \
Shadow, Alignment, MarginL, MarginR, MarginV, AlphaLevel, Encoding
Style: Default,Arial,20,65535,65535,65535,-2147483640,-1,0,1,3,0,2,\
30,30,30,0,0
Style: Titre_episode,Akbar,140,15724527,65535,65535,986895,-1,0,1,1,\
0,3,30,30,30,0,0
Style: Wolf main,Wolf_Rain,56,15724527,15724527,15724527,4144959,0,\
0,1,1,2,2,5,5,30,0,0

And here are the two blocks that would be generated.

Block’s timestamp: 00:02:40.650 BlockDuration: 00:00:01.140

1,,Wolf main,Cher,0000,0000,0000,,Et les enregistrements de ses \
ondes delta ?

Block’s timestamp: 00:02:42.420 BlockDuration: 00:00:01.730

2,,Wolf main,autre,0000,0000,0000,,Toujours rien.
```

6.4. WebVTT

The "Web Video Text Tracks Format" (short: WebVTT) is developed by the World Wide Web Consortium (W3C). Its specifications are freely available at [WebVTT].

The guiding principles for the storage of WebVTT in Matroska are:

* Consistency: store data in a similar way to other subtitle codecs

* Simplicity: making decoding and remuxing as easy as possible for existing infrastructures

* Completeness: keeping as much data as possible from the original WebVTT file

6.4.1. Track Parameters

The CodecID to use is S_TEXT/WEBVTT.

This CodecPrivate contains all global blocks before the first subtitle entry. This starts at the "WEBVTT" file identification marker but excludes the optional byte order mark.

6.4.2. Storage of non-global WebVTT blocks

Non-global WebVTT blocks (e.g., "NOTE") before a WebVTT caption or subtitle cue text are stored in Matroska’s BlockAddition element together with the Matroska Block containing the WebVTT caption or subtitle cue text these blocks precede (see below for the actual format).

6.4.3. Storage of Cues in Matroska blocks

Each WebVTT caption or subtitle cue text is stored directly in the Matroska Block.

A muxer MUST change all WebVTT cue timestamp(s) present within the WebVTT caption or subtitle cue text to be relative to the Matroska Block’s timestamp.

The Cue’s start timestamp is used as the Matroska Block’s timestamp.

The difference between the Cue’s end timestamp and its start timestamp is used as the Matroska BlockDuration.

6.4.4. BlockAdditions

Each Matroska Block may be accompanied by one BlockAdditions element. Its format is as follows:

1. The first line contains the WebVTT caption or subtitle cue text’s optional WebVTT cue settings list followed by one line feed character (U+0x000a). The WebVTT cue settings list may be empty, in which case the line consists of the line feed character only.

2. The second line contains the WebVTT caption or subtitle cue text’s optional WebVTT cue identifier followed by one line feed character (U+0x000a). The line may be empty indicating that there was no WebVTT cue identifier in the source file, in which case the line consists of the line feed character only.

3. The third and all following lines contain all WebVTT comment block(s) that precede the current WebVTT cue block. These may be absent. Each WebVTT comment block includes its WebVTT line terminator and is followed by one line feed character (U+0x000a). The last WebVTT comment block MAY omit the WebVTT line terminator and the line feed character.

If there is no Matroska BlockAddition element stored together with the Matroska Block, then WebVTT cue settings list, WebVTT cue identifier, and WebVTT comment block(s) MUST be assumed to be absent.

6.4.5. Example of Matroska Muxing

Here’s an example how a WebVTT is transformed.

Consider the following example WebVTT file:

```
WEBVTT with text after the signature

STYLE
::cue {
  background-image: linear-gradient(to bottom, dimgray, lightgray);
  color: papayawhip;
}
/* Style blocks cannot use blank lines nor "dash dash greater \
than" */

NOTE comment blocks can be used between style blocks.

STYLE
::cue(b) {
  color: peachpuff;
}

REGION
id:bill
width:40%
lines:3
regionanchor:0%,100%
viewportanchor:10%,90%
scroll:up
```

```
NOTE
Notes always span a whole block and can cover multiple
lines. Like this one.
An empty line ends the block.

hello
00:00:00.000 --> 00:00:10.000
Example entry 1: Hello <b>world</b>.

NOTE style blocks can’t appear after the first cue.

00:00:25.000 --> 00:00:35.000
Example entry 2: Another entry.
This one has multiple lines.

00:01:03.000 --> 00:01:06.500 position:90% align:right size:35%
Entry 3: That stuff to the right of the \
timestamps are cue settings.

00:03:10.000 --> 00:03:20.000
Entry 4: Entries can even include timestamps.
For example:<00:03:15.000>This becomes visible five seconds
after the first part.
```

6.4.5.1. CodecPrivate

The following XML depicts the CodecPrivate element contains the UTF-8 text of all global WebVTT blocks before the first subtitle entry:

```
<TrackEntry>
  <CodecPrivate>WEBVTT with text after the signature

STYLE
::cue {
  background-image: linear-gradient(to bottom, dimgray, lightgray);
  color: papayawhip;
}
/* Style blocks cannot use blank lines nor "dash dash greater \
than" */

NOTE comment blocks can be used between style blocks.

STYLE
::cue(b) {
  color: peachpuff;
}

REGION
id:bill
width:40%
lines:3
regionanchor:0%,100%
viewportanchor:10%,90%
scroll:up

NOTE
Notes always span a whole block and can cover multiple
lines. Like this one.
An empty line ends the block.</CodecPrivate>
</TrackEntry>
```

6.4.5.2. Cue Block 1

The following XML depicts the nested elements of a BlockGroup element with of the first WebVTT cue block. The cue block timings are turned into Matroska timestamps. The last line feed character (U+0x000a) is stripped.

The BlockAddition content starts with one empty line as there’s no WebVTT cue settings list:

```
<BlockGroup>
  <Block timestamp="0">Example entry 1: Hello <b>world</b>.</Block>
  <BlockDuration>10000</BlockDuration> <!-- 10000 Ticks of 1 ms -->
  <BlockAdditions>
    <BlockMore>
      <BlockAddID>1</BlockAddID>
      <BlockAdditional>

hello</BlockAdditional>
    </BlockMore>
  </BlockAdditions>
</BlockGroup>
```

6.4.5.3. Cue Block 2

The following XML depicts the nested elements of a BlockGroup element with of the second WebVTT cue block. The last line feed character (U+0x000a) is stripped.

The BlockAddition content starts with two empty lines as there’s neither a WebVTT cue settings list nor a WebVTT cue identifier, Then follows the content of the WebVTT comment block(s). The last line feed character (U+0x000a) is stripped.

```
<BlockGroup>
  <Block timestamp="25000">Example entry 2: Another entry.
This one has multiple lines.</Block>
  <BlockDuration>10000</BlockDuration>
  <BlockAdditions>
    <BlockMore>
      <BlockAddID>1</BlockAddID>
      <BlockAdditional>

NOTE style blocks can’t appear after the first cue.</BlockAdditional>
    </BlockMore>
  </BlockAdditions>
</BlockGroup>
```

6.4.5.4. Cue Block 3

The following XML depicts the nested elements of a BlockGroup element with of the third WebVTT cue block. The last line feed character (U+0x000a) is stripped.

The BlockAddition content ends with an empty line as there is no WebVTT cue identifier and there were no WebVTT comment block.

```
<BlockGroup>
  <Block timestamp="63000">Entry 3: That stuff to the right of the \
timestamps are cue settings.</Block>
  <BlockDuration>3500</BlockDuration>
  <BlockAdditions>
    <BlockMore>
      <BlockAddID>1</BlockAddID>
      <BlockAdditional>
position:90% align:right size:35%

</BlockAdditional>
    </BlockMore>
  </BlockAdditions>
</BlockGroup>
```

6.4.5.5. Cue Block 4

The following XML depicts the nested elements of a BlockGroup element with of the fourth WebVTT cue block. The last line feed character (U+0x000a) is stripped.

No BlockAddition is used.

```
<BlockGroup>
  <Block timestamp="190000">Entry 4: Entries can even include
  timestamps.
For example:<00:03:15.000>This becomes visible five seconds
after the first part.</Block>
  <BlockDuration>10000</BlockDuration>
</BlockGroup>
```

6.4.6. Storage of WebVTT in Matroska vs. WebM

Note: the storage of WebVTT in Matroska is not the same as the design document for storage of WebVTT in WebM [WebM-WebVTT]. There are several reasons for this including but not limited to: the WebM document is old (from February 2012) and was based on an earlier draft of WebVTT and ignores several parts that were added to WebVTT later; WebM does still not support subtitles at all [WebMContainer]; the proposal suggests splitting the information across multiple tracks making demuxer’s and remuxer’s life very difficult.

WebM uses the "D_WEBVTT/SUBTITLES", "D_WEBVTT/CAPTIONS", "D_WEBVTT/DESCRIPTIONS", and "D_WEBVTT/METADATA" CodecID with different tracks depending on the data type and without a CodecPrivate.

6.5. HDMV Presentation Graphics Subtitles

The specifications for the HDMV Presentation Graphics Subtitle format (short: HDMV PGS) can be found in in section 9.14 "HDMV graphics streams" of the Blu-ray specifications [Blu-ray.Part3].

6.5.1. Track Parameters

The CodecID to use is S_HDMV/PGS. A CodecPrivate element is not used.

6.5.2. Matroska Blocks

Each HDMV PGS Segment (short: Segment) will be stored in a Matroska Block. A Segment is the data structure described in section 9.14.2.1 "Segment coding structure and parameters" of the Blu-ray specifications [Blu-ray.Part3].

Each Segment contains a presentation timestamp. This timestamp will be used as the timestamp for the Matroska Block.

A Segment is normally shown until a subsequent Segment is encountered. Therefore, the Matroska Block MAY have no Duration. In that case, a player MUST display a Segment within a Matroska Block until the next Segment is encountered.

A muxer MAY use a Duration, e.g., by calculating the distance between two subsequent Segments. If a Matroska Block has a Duration, a player MUST display that Segment only for the duration of the BlockDuration.

6.6. HDMV Text Subtitles

The specifications for the HDMV Text Subtitle format (short: HDMV TextST) can be found in section 9.15 "HDMV text subtitle streams" of the Blu-ray specifications [Blu-ray.Part3].

6.6.1. Track Parameters

The CodecID to use is S_HDMV/TEXTST.

A CodecPrivate element is required. It MUST contain the stream’s Dialog Style Segment as described in section 9.15.4.2 "Dialog Style Segment" of the Blu-ray specifications [Blu-ray.Part3].

6.6.2. Matroska Blocks

Each HDMV Dialog Presentation Segment (short: Segment) will be stored in a Matroska Block. A Segment is the data structure described in section 9.15.4.3 "Dialog presentation segment" of the Blu-ray specifications [Blu-ray.Part3].

Each Segment contains a start and an end presentation timestamp (short: start PTS & end PTS). The start PTS will be used as the timestamp for the Matroska Block. The Matroska Block MUST have a Duration, and that Duration is the difference between the end PTS and the start PTS.

A player MUST use the Matroska Block’s timestamp and BlockDuration instead of the Segment’s start and end PTS for determining when and how long to show the Segment.

6.6.3. Character set

When TextST subtitles are stored inside Matroska, the only allowed character set is UTF-8.

Each HDMV text subtitle stream in a Blu-ray can use one of a handful of character sets. This information is not stored in the MPEG2 Transport Stream itself but in the accompanying Clip Information file.

Therefore, a muxer MUST parse the accompanying Clip Information file. If the information indicates a character set other than UTF-8, it MUST re-encode all text Dialog Presentation Segments from the indicated character set to UTF-8 prior to storing them in Matroska.

6.7. Digital Video Broadcasting (DVB) subtitles

The specifications for the Digital Video Broadcasting subtitle bitstream format (short: DVB subtitles) can be found in the [ETSI.EN300-743] document. The storage of DVB subtitles in MPEG transport streams is specified in the [ETSI.EN300-468] document.

6.7.1. Track Parameters

The CodecID to use is S_DVBSUB.

The CodecPrivate element is five bytes long and has the following structure:

* 2 bytes: composition page ID (bit string, left bit first)

* 2 bytes: ancillary page ID (bit string, left bit first)

* 1 byte: subtitling type (bit string, left bit first)

The semantics of these bytes are the same as the ones described in section 6.2.41 "Subtitling descriptor" of [ETSI.EN300-468].

6.7.2. Matroska Blocks

Each Matroska Block consists of one or more DVB Subtitle Segments as described in section 7.2 "Syntax and semantics of the subtitling segment" of [ETSI.EN300-743].

Each Matroska Block SHOULD have a Duration indicating how long the DVB Subtitle Segments in that Block SHOULD be displayed.

6.8. ARIB (ISDB) subtitles

The specifications for the ARIB B-24 subtitle bitstream format (short: ARIB subtitles) and its storage in MPEG transport streams can be found in the documents [ARIB.STD-B24], [ARIB.STD-B10], and [ARIB.TR-B14].

6.8.1. Track Parameters

The CodecID to use is S_ARIBSUB.

The CodecPrivate element is three bytes long and has the following structure:

* 1 byte: component tag (bit string, left bit first)

* 2 bytes: data component ID (bit string, left bit first)

The semantics of the component tag are the same as those described in [ARIB.STD-B10], part 2, Annex J. The semantics of the data component ID are the same as those described in [ARIB.TR-B14], fascicle 2, Vol. 3, Section 2, 4.2.8.1.

6.8.2. Matroska Blocks

Each Matroska Block consists of a single synchronized PES data structure as described in chapter 5 "Independent PES transmission protocol" of [ARIB.STD-B24], volume 3, with a Synchronized_PES_data_byte block containing one or more ISDB Caption Data Groups as described in chapter 9 "Transmission of caption and superimpose" of [ARIB.STD-B24], volume 1, part 3. All of the Caption Statement Data Groups in a given Matroska Track MUST use the same

language index.

A Data Group is normally shown until a subsequent Group provides instructions to clear it. Therefore, the Matroska Block SHOULD NOT have a Duration. A player SHOULD display a Data Group within a Matroska Block until its internal duration elapses, or until a subsequent Data Group removes it.

7. Block Additional Mapping

Extra data or metadata can be added to each Block using BlockAdditional data. Each BlockAdditional contains a BlockAddID that identifies the kind of data it contains. When the BlockAddID is set to "1" the contents of the BlockAdditional element are defined by the "Codec BlockAdditions" section of the codec; see Section 3.1.5.

The following XML depicts the nested elements of a BlockGroup element with an example of BlockAdditions with a BlockAddID of "1":

```
<BlockGroup>
  <Block>{Binary data of a VP9 video frame in YUV}</Block>
  <BlockAdditions>
    <BlockMore>
      <BlockAddID>1</BlockAddID>
      <BlockAdditional>
        {alpha channel encoding to supplement the VP9 frame}
      </BlockAdditional>
    </BlockMore>
  </BlockAdditions>
</BlockGroup>
```

When the BlockAddID is set a value greater than "1", then the contents of the BlockAdditional element are defined by the BlockAdditionMapping element, within the associated TrackEntry element, where the BlockAddID element of BlockAdditional element equals the BlockAddIDValue of the associated TrackEntry’s BlockAdditionMapping element. That BlockAdditionMapping element identifies a particular Block Additional Mapping by the BlockAddIDType.

The values of BlockAddID that are 2 or greater have no semantic meaning, but simply associate the BlockMore element with a BlockAdditionMapping of the associated Track. See Section 7 on Block Additional Mappings for more information.

It is RECOMMENDED to not use a value of 4 for BlockAddID and BlockAddIDValue when BlockAddIDType is not 4 -- i.e., ITU T.35 metadata Section 4.2.3, as some WebM-oriented demuxers consider a block with BlockAddID of 4 as ITU T.35 metadata without checking the BlockAddIDType element.

The following XML depicts a use of a Block Additional Mapping to associate a timecode value with a Block:

```
<Segment>
  <!--Mandatory elements omitted for readability-->
  <Tracks>
    <TrackEntry>
      <TrackNumber>1</TrackNumber>
      <TrackUID>568001708</TrackUID>
      <TrackType>1</TrackType>
      <BlockAdditionMapping>
        <BlockAddIDValue>2</BlockAddIDValue><!--arbitrary value
          used in BlockAddID-->
        <BlockAddIDName>timecode</BlockAddIDName>
        <BlockAddIDType>121</BlockAddIDType>
      </BlockAdditionMapping>
      <CodecID>V_FFV1</CodecID>
      <Video>
        <PixelWidth>1920</PixelWidth>
        <PixelHeight>1080</PixelHeight>
      </Video>
    </TrackEntry>
  </Tracks>
  <Cluster>
    <Timestamp>3000</Timestamp>
    <BlockGroup>
      <Block>{binary video frame}</Block>
      <BlockAdditions>
        <BlockMore>
          <BlockAddID>2</BlockAddID><!--arbitrary value from
            BlockAdditionMapping-->
          <BlockAdditional>01:00:00:00</BlockAdditional><!--presented
           as a string for readability but should use binary encoding
           defined in the associated mapping -->
        </BlockMore>
      </BlockAdditions>
    </BlockGroup>
  </Cluster>
</Segment>
```

Block Additional Mappings detail how additional data is stored in the BlockMore element with a BlockAdditionMapping element, within the Track element, which identifies the BlockAdditional content. Block Additional Mappings define the BlockAddIDType value reserved to identify that type of data as well as providing an optional label stored within the BlockAddIDName element. When the Block Additional Mapping is dependent on additional contextual information, then the Mapping SHOULD describe how such additional contextual information is stored within the BlockAddIDExtraData element.

7.1. SMPTE ST 12-1 Timecode Description

SMPTE ST 12-1 timecode values can be stored in the BlockMore element to associate the content of a Matroska Block with a particular timecode value. If the Block uses Lacing, the timecode value is associated with the first frame of the Lace.

The Block Additional Mapping contains a full binary representation of a 64-bit SMPTE timecode value stored in big-endian format and expressed exactly as defined in Section 8 and 9 of SMPTE 12M [SMPTE.ST12-1], without the 16-bit synchronization word. For convenience, here are the time address bit assignments as described in Section 6.2 of [RFC5484]:

Bit Positions	Label
0--3	Units of frames
8--9	Tens of frames
16--19	Units of seconds
24--26	Tens of seconds
32--35	Units of minutes
40--42	Tens of minutes
48--51	Units of hours
56--57	Tens of hours

Table 3: SMPTE ST 12-1 Time Address Bit Positions

For example, a timecode value of "07:12:26;18" can be expressed as a 64-bit SMPTE 12M value as:

```
10000000 01100000 01100000 01010000
00100000 00110000 01110000 00000000
```

Or with the irrelevant bits marked with an "x" which gives 26 usable bits:

```
1000xxxx 01xxxxxx 0110xxxx 010xxxxx
0010xxxx 001xxxxx 0111xxxx 00xxxxxx
```

This is interpreted in hexadecimal:

* 0x8 units of frames
* 0x1 tens of frames
* 0x6 units of seconds
* 0x2 tens of seconds
* 0x2 units of minutes
* 0x1 tens of minutes
* 0x7 units of hours
* 0x0 tens of hours

Given no value is above 9, the BCD coding correspond to the actual values:

* 8 units of frames
* 1 tens of frames
* 6 units of seconds
* 2 tens of seconds
* 2 units of minutes
* 1 tens of minutes
* 7 units of hours
* 0 tens of hours

Or:

* 18 frames
* 26 seconds
* 12 minutes
* 07 hours

8. Security Considerations

This document inherits security considerations from the EBML [RFC8794] and Matroska [RFC9559] documents.

Codec handling may be one of the more error-prone aspect of using Matroska. The parsing and interpretation of binary data can lead to many types of security issues. Although these issues don't come from Matroska itself, it's worth noting some issues that need to be considered. Security reviews for each codec cannot be complete without reference to the appropriate references for each codec. While incorrect parsing and interpretation of binary data will not cause extraction from a Matroska container to fail, the application performing this extraction will be affected by these errors.

The CodecPrivate may be missing from the TrackEntry description. The TrackEntry MAY be discarded in that case.

An existing CodecPrivate data may be corrupted or incomplete or too big. The TrackEntry MAY be discarded in that case.

A lot of codec have internal fields to hold values that are already found in the TrackEntry like the video dimensions or the audio sampling frequency. If these values differ that can lead to playback issues and even crashes.

9. IANA Considerations

9.1. Matroska Codec IDs Registry

This document defines registries for Codec IDs stored in the CodecID element. A CodecID is a case-sensitive ASCII string with a prefix defined in Table 1. The details of the string format are found in Section 3.1.1.

"Matroska Codec IDs" are to be allocated according to the "Expert Review" policy [RFC8126].

To register a new Codec ID in this registry, one needs a Codec ID string, a TrackType value, a description, a Change Controller, and an optional Reference to a document describing the Codec ID.

Some Codec IDs values are deprecated. Such Codec IDs are marked as "Reclaimed" in the "Matroska Codec IDs" registry.

Table 4 shows the initial contents of the "Matroska Codec IDs" registry. The Change Controller for the initial entries is the IETF.

Codec ID	Track Type	Description	Reference
V_AV1	1	Alliance for Open Media AV1	This document, Section 3.3.1
V_AVS2	1	AVS2-P2/ IEEE.1857.4	This document, Section 3.3.2
V_AVS3	1	AVS3-P2/ IEEE.1857.10	This document, Section 3.3.3
V_CAVS	1	AVS1-P2/ IEEE.1857.3	This document, Section 3.3.4
V_DIRAC	1	Dirac / VC-2	This document, Section 3.3.5
V_FFV1	1	FFV1	This document, Section 3.3.6
V_JPEG2000	1	JPEG 2000	This document, Section 3.3.7
V_MJPEG	1	Motion JPEG	This document, Section 3.3.8
V_MPEGH/ISO/ HEVC	1	HEVC/H.265	This document, Section 3.3.9
V_MPEGI/ISO/VVC	1	VVC/H.266	This document, Section 3.3.10
V_MPEG1	1	MPEG 1	This document, Section 3.3.11
V_MPEG2	1	MPEG 2	This document, Section 3.3.12
V_MPEG4/ISO/AVC	1	AVC/H.264	This document, Section 3.3.13
V_MPEG4/ISO/AP	1	MPEG4 ISO advanced profile	This document, Section 3.3.14
V_MPEG4/ISO/ASP	1	MPEG4 ISO advanced simple profile	This document, Section 3.3.15

V_MPEG4/ISO/SP	1	MPEG4 ISO simple profile	This document, Section 3.3.16
V_MPEG4/MS/V3	1	Microsoft MPEG4 V3	This document, Section 3.3.17
V_MS/VFW/FOURCC	1	Microsoft Video Codec Manager	This document, Section 3.3.18
V_QUICKTIME	1	Video taken from QuickTime files	This document, Section 3.3.19
V_PRORES	1	Apple ProRes	This document, Section 3.3.20
V_REAL/RV10	1	RealVideo 1.0 aka RealVideo 5	This document, Section 3.3.21
V_REAL/RV20	1	RealVideo G2 and RealVideo G2+SVT	This document, Section 3.3.22
V_REAL/RV30	1	RealVideo 8	This document, Section 3.3.23
V_REAL/RV40	1	rv40 : RealVideo 9	This document, Section 3.3.24
V_THEORA	1	Theora	This document, Section 3.3.25
V_UNCOMPRESSED	1	Raw uncompressed video frames	This document, Section 3.3.26
V_VC1	1	VC-1	This document, Section 3.3.27
V_VP8	1	VP8 Codec format	This document, Section 3.3.28
V_VP9	1	VP9 Codec format	This document, Section 3.3.29
A_AAC	2	Advanced Audio Coding	This document, Section 3.4.1
A_AAC/MPEG2/LC	2	Low Complexity	This document, Section 3.4.2

A_AAC/MPEG2/LC/SBR	2	Low Complexity with Spectral Band Replication	This document, Section 3.4.3
A_AAC/MPEG2/MAIN	2	MPEG2 Main Profile	This document, Section 3.4.4
A_AAC/MPEG2/SSR	2	Scalable Sampling Rate	This document, Section 3.4.5
A_AAC/MPEG4/LC	2	Low Complexity	This document, Section 3.4.6
A_AAC/MPEG4/LC/SBR	2	Low Complexity with Spectral Band Replication	This document, Section 3.4.7
A_AAC/MPEG4/LTP	2	Long Term Prediction	This document, Section 3.4.8
A_AAC/MPEG4/MAIN	2	MPEG4 Main Profile	This document, Section 3.4.9
A_AAC/MPEG4/SSR	2	Scalable Sampling Rate	This document, Section 3.4.10
A_AC3	2	Dolby Digital / AC-3	This document, Section 3.4.11
A_AC3/BSID9	2	Dolby Digital / AC-3	This document, Section 3.4.12
A_AC3/BSID10	2	Dolby Digital / AC-3	This document, Section 3.4.13
A_ALAC	2	ALAC (Apple Lossless Audio Codec)	This document, Section 3.4.14
A_ATRAC/AT1	2	Sony ATRAC1 Codec	This document, Section 3.4.15
A_DTS	2	Digital Theatre System	This document, Section 3.4.16
A_DTS/EXPRESS	2	Digital Theatre System Express	This document, Section 3.4.17

A_DTS/LOSSLESS	2	Digital Theatre System Lossless	This document, Section 3.4.18
A_EAC3	2	Dolby Digital Plus / E-AC-3	This document, Section 3.4.19
A_FLAC	2	FLAC	This document, Section 3.4.20
A_MLP	2	Meridian Lossless Packing / MLP	This document, Section 3.4.21
A_MPEG/L1	2	MPEG Audio 1, 2 Layer I	This document, Section 3.4.22
A_MPEG/L2	2	MPEG Audio 1, 2 Layer II	This document, Section 3.4.23
A_MPEG/L3	2	MPEG Audio 1, 2, 2.5 Layer III	This document, Section 3.4.24
A_MS/ACM	2	Microsoft Audio Codec Manager (ACM)	This document, Section 3.4.25
A_REAL/14_4	2	Real Audio 1	This document, Section 3.4.26
A_REAL/28_8	2	Real Audio 2	This document, Section 3.4.27
A_REAL/ATRC	2	Sony Atrac3 Codec	This document, Section 3.4.28
A_REAL/COOK	2	Real Audio Cook Codec	This document, Section 3.4.29
A_REAL/RALF	2	Real Audio Lossless Format	This document, Section 3.4.30
A_REAL/SIPR	2	Sipro Voice Codec	This document, Section 3.4.31
A_OPUS	2	Opus interactive speech and audio codec	This document, Section 3.4.32

A_PCM/FLOAT/IEEE	2	Floating-Point, IEEE compatible	This document, Section 3.4.33
A_PCM/INT/BIG	2	PCM Integer Big Endian	This document, Section 3.4.34
A_PCM/INT/LIT	2	PCM Integer Little Endian	This document, Section 3.4.35
A_QUICKTIME	2	Audio taken from QuickTime files	This document, Section 3.4.36
A_QUICKTIME/QDMC	2	QDesign Music	This document, Section 3.4.37
A_QUICKTIME/QDM2	2	QDesign Music v2	This document, Section 3.4.38
A_TRUEHD	2	Dolby TrueHD	This document, Section 3.4.39
A_TTA1	2	The True Audio	This document, Section 3.4.40
A_VORBIS	2	Vorbis	This document, Section 3.4.41
A_WAVPACK4	2	WavPack	This document, Section 3.4.42
S_ARIBSUB	17	ARIB STD-B24 subtitles	This document, Section 3.5.1
S_DVBSUB	17	Digital Video Broadcasting subtitles	This document, Section 3.5.2
S_HDMV/PGS	17	HDMV presentation graphics subtitles	This document, Section 3.5.3
S_HDMV/TEXTST	17	HDMV text subtitles	This document, Section 3.5.4
S_KATE	17	Karaoke And Text Encapsulation	This document, Section 3.5.5
S_IMAGE/BMP	17	Bitmap	This document, Section 3.5.6

S_ASS	17	Advanced SubStation Alpha Format	Reclaimed, Section 3.5.7
S_TEXT/ASS	17	Advanced SubStation Alpha Format	This document, Section 3.5.7
S_TEXT/ASCII	17	ASCII Plain Text	This document, Section 3.5.8
S_TEXT/SSA	17	SubStation Alpha Format	This document, Section 3.5.9
S_TEXT/USF	17	Universal Subtitle Format	This document, Section 3.5.10
S_TEXT/UTF8	17	UTF-8 Plain Text	This document, Section 3.5.11
S_TEXT/WEBVTT	17	Web Video Text Tracks (WebVTT)	This document, Section 3.5.12
S_SSA	17	SubStation Alpha Format	Reclaimed, Section 3.5.7
S_VOBSUB	17	VobSub subtitles	This document, Section 3.5.13
B_VOBBTN	18	VobBtn Buttons	This document, Section 3.6.1

Table 4: Initial Contents of "Matroska Codec IDs" Registry

9.2. Matroska BlockAdditional Type IDs Registry

This document defines registries for BlockAdditional Type IDs stored in the BlockAddIDType element. The values correspond to the unsigned integer BlockAddIDType value described in Section 5.1.4.1.17.3 of [RFC9559].

"Matroska BlockAdditional Type IDs" are to be allocated according to the "Expert Review" policy [RFC8126].

To register a new BlockAdditional Type ID in this registry, one needs a BlockAddIDType unsigned integer, a BlockAddIDName string value, a Change Controller, and an optional Reference to a document describing the BlockAdditional Type ID.

Table 5 shows the initial contents of the "Matroska BlockAdditional Type IDs" registry. The Change Controller for the initial entries is the IETF.

BlockAddIDType	BlockAddIDName	Reference
0	BlockAddIDValue	This document, Section 4.2.1
1	Opaque data	This document, Section 4.2.2
4	ITU T.35 metadata	This document, Section 4.2.3
121	SMPTE ST 12-1 timecode	This document, Section 4.2.4
0x61766345	Dolby Vision enhancement-layer AVC configuration	This document, Section 4.2.5
0x68766345	Dolby Vision enhancement-layer HEVC configuration	This document, Section 4.2.6
0x64766343	Dolby Vision configuration dvcC	This document, Section 4.2.7
0x64767643	Dolby Vision configuration dvvC	This document, Section 4.2.8
0x64767743	Dolby Vision configuration dvwC	This document, Section 4.2.9
0x6D766343	MVC configuration	This document, Section 4.2.10

Table 5: Initial Contents of "Matroska BlockAdditional Type IDs" Registry

10. References

10.1. Normative References

[ALAC] Apple Inc., "Apple Lossless Format "Magic Cookie" Description", 12 December 2012, <https://github.com/macosforge/alac/blob/master/ALACMagicCookieDescription.txt>.

[ARIB.STD-B10] ARIB, "Service Information for Digital Broadcasting System", 5 December 2019, <https://www.arib.or.jp/english/std_tr/broadcasting/desc/std-b10.html>.

[ARIB.STD-B24] ARIB, "Data Coding and Transmission Specification for Digital Broadcasting", 6 October 2022, <https://www.arib.or.jp/english/std_tr/broadcasting/desc/std-b24.html>.

[ARIB.TR-B14] ARIB, "Operational Guidelines for Digital Terrestrial Television Broadcasting", 6 October 2022, <https://www.arib.or.jp/english/std_tr/broadcasting/desc/tr-b14.html>.

[ATSC.A52] Advanced Television Systems Committee, "ATSC Standard: Digital Audio Compression (AC-3, E-AC-3)", 25 January 2018, <https://www.atsc.org/wp-content/uploads/2021/04/A52-2018.pdf>.

[AV1] Alliance for Open Media, "AV1 Bitstream & Decoding Process Specification", 8 January 2019, <https://aomediacodec.github.io/av1-spec/av1-spec.pdf>.

[AV1-ISOBMFF] Alliance for Open Media, "AV1 Codec ISO Media File Format Binding", 3 April 2024, <https://aomediacodec.github.io/av1-isobmff/>.

[BITMAPINFOHEADER] Microsoft Corporation, "BITMAPINFOHEADER structure", 26 January 2024, <https://learn.microsoft.com/en-us/windows/win32/api/wingdi/ns-wingdi-bitmapinfoheader>.

[Blu-ray.Part3]
Blu-ray Disc Association, "System Description Blu-ray Disc Read-Only Format - Part 3: Audio Visual Basic Specifications Ver 2.01", January 2007, <https://blu-raydisc.info/format-spec/re2-spec.php>.

[Dirac] British Broadcasting Corporation, "Dirac Specification", 23 September 2008, <https://web.archive.org/web/20150503015104/http://diracvideo.org/download/specification/dirac-spec-latest.pdf>.

[DolbyVision-ISOBMFF]
Dolby, "Dolby Vision Streams Within the ISO Base MediaFile Format", 7 November 2023, <https://dolby.my.salesforce.com/sfc/p/700000009YuG/a/4u000000l6FB/076wHYEmyEfz09m0V1bo85_25hlUJjaiWTbzorNmYY4>.

[DVD-Info.PCI]
"Presentation Control Information (PCI) Packet Layout", <https://dvd.sourceforge.net/dvdinfo/pci_pkt.html>.

[ETSI.EN300-468]
European Telecommunications Standards Institute, "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems", ETSI EN 300 468, December 2023, <https://www.etsi.org/deliver/etsi_en/300400_300499/300468/01.18.01_60/en_300468v011801p.pdf>.

[ETSI.EN300-743]
European Telecommunications Standards Institute, "Digital Video Broadcasting (DVB); Subtitling systems", ETSI EN 300 743, October 2018, <https://www.etsi.org/deliver/etsi_en/300700_300799/300743/01.06.01_60/en_300743v010601p.pdf>.

[ETSI.TS102-114]
European Telecommunications Standards Institute, "DTS Coherent Acoustics; Core and Extensions with Additional Profiles", ETSI TS 102 114, August 2019, <https://www.etsi.org/deliver/etsi_ts/102100_102199/102114/01.06.01_60/ts_102114v010601p.pdf>.

[ETSI.TS102-366]
European Telecommunications Standards Institute, "Digital Audio Compression (AC-3, Enhanced AC-3) Standard", ETSI TS 102 366, September 2017, <https://www.etsi.org/deliver/etsi_ts/102300_102399/102366/01.04.01_60/ts_102366v010401p.pdf>.

[IEEE.1857-10]
IEEE, "IEEE Standard for Third Generation Video Coding", 9 November 2021, <https://standards.ieee.org/ieee/1857.10/7722/>.

[IEEE.1857-3]
IEEE, "IEEE Standard for a System of Advanced Audio and Video Coding", 8 November 2023, <https://standards.ieee.org/ieee/1857.3/10645/>.

[IEEE.1857-4]
IEEE, "IEEE Standard for Second-Generation IEEE 1857 Video Coding", 23 October 2018, <https://standards.ieee.org/ieee/1857.4/5817/>.

[IEEE.754] IEEE, "IEEE Standard for Binary Floating-Point Arithmetic", 13 June 2019, <https://standards.ieee.org/standard/754-2019.html>.

[ISO.11172-2]
International Organization for Standardization, "Information technology - Coding of moving pictures and associated audio for digital storage median at up to about 1,5 Mbit/s - Part 2: Video", ISO 11172-2:1993, August 1993.

[ISO.11172-3]
International Organization for Standardization, "Information technology - Coding of moving pictures and associated audio for digital storage median at up to about 1,5 Mbit/s - Part 3: Audio", ISO 11172-2:1993, August 1993.

[ISO.14496-15]
International Organization for Standardization, "Information technology - Coding of audio-visual objects - Part 15: Carriage of network abstraction layer (NAL) unit structured video in ISO base media file format", ISO 14496-15:2024, October 2024.

[ISO.14496-2]
International Organization for Standardization, "Information technology - Coding of audio-visual objects - Part 2: Visual", ISO 14496-2:2004, June 2004.

[ISO.14496-3]
International Organization for Standardization, "Information technology - Coding of audio-visual objects - Part 3: Audio", ISO 14496-3:2019, December 2019.

[ITU-T.H.262]
ITU-T, "Procedure for the allocation of ITU-T defined codes for non-standard facilities", ITU-T Recommendation H.262, July 1995, <https://www.itu.int/rec/T-REC-H.262/en>.

[JPEG] ITU-T, "INFORMATION TECHNOLOGY - DIGITAL COMPRESSION AND CODING OF CONTINUOUS-TONE STILL IMAGES - REQUIREMENTS AND GUIDELINES", ITU-T Recommendation T.81, September 1992, <https://www.w3.org/Graphics/JPEG/itu-t81.pdf>.

[JPEG2000] ITU-T, "Information technology - JPEG 2000 image coding system: Core coding system", ITU-T Recommendation T.800, July 2024, <https://www.itu.int/rec/T-REC-T.800-202407-I/en>.

[librmff] Bunkus, M., "RealMedia file format access library", 20 February 2021, <https://gitlab.com/mbunkus/mkvtoolnix/-/blob/main/lib/librmff/librmff.h>.

[OggKate] Xiph.Org Foundation, "OggKate", 21 February 2023, <http://wiki.xiph.org/index.php/OggKate>.

[QTFF] Apple Inc., "QuickTime File Format", 26 January 2024, <https://developer.apple.com/documentation/quicktime-file-format>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6648] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating the "X-" Prefix and Similar Constructs in Application Protocols", BCP 178, RFC 6648, DOI 10.17487/RFC6648, June 2012, <https://www.rfc-editor.org/info/rfc6648>.

[RFC6716] Valin, JM., Vos, K., and T. Terriberry, "Definition of the Opus Audio Codec", RFC 6716, DOI 10.17487/RFC6716, September 2012, <https://www.rfc-editor.org/info/rfc6716>.

[RFC7845] Terriberry, T., Lee, R., and R. Giles, "Ogg Encapsulation for the Opus Audio Codec", RFC 7845, DOI 10.17487/RFC7845, April 2016, <https://www.rfc-editor.org/info/rfc7845>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8794] Lhomme, S., Rice, D., and M. Bunkus, "Extensible Binary Meta Language", RFC 8794, DOI 10.17487/RFC8794, July 2020, <https://www.rfc-editor.org/info/rfc8794>.

[RFC9043] Niedermayer, M., Rice, D., and J. Martinez, "FFV1 Video Coding Format Versions 0, 1, and 3", RFC 9043, DOI 10.17487/RFC9043, August 2021, <https://www.rfc-editor.org/info/rfc9043>.

[RFC9559] Lhomme, S., Bunkus, M., and D. Rice, "Matroska Media Container Format Specification", RFC 9559, DOI 10.17487/RFC9559, October 2024, <https://www.rfc-editor.org/info/rfc9559>.

[RFC9639] van Beurden, M.Q.C. and A. Weaver, "Free Lossless Audio Codec (FLAC)", RFC 9639, DOI 10.17487/RFC9639, December 2024, <https://www.rfc-editor.org/info/rfc9639>.

[SMPTE.RDD36] SMPTE, "Apple ProRes Bitstream Syntax and Decoding Process", 9 September 2022, <https://pub.smpte.org/doc/rdd36/20220909-pub/>.

[SMPTE.RP2025-2007] SMPTE, "VC-1 Bitstream Storage in the ISO Base Media File Format", 4 April 2007, <https://pub.smpte.org/pub/rp2025/rp2025-2007.pdf>.

[SMPTE.ST12-1]
SMPTE, "Time and Control Code", ST 12-1:2014, DOI 10.5594/SMPTE.ST12-1.2014, 20 February 2014, <https://pub.smpte.org/doc/st12-1/20140220-pub/>.

[SSA] "Sub Station Alpha v4.00+ Script Format", <http://www.tcax.org/docs/ass-specs.htm>.

[Theora] Xiph.Org Foundation, "Theora Specification", 3 June 2017, <https://www.theora.org/doc/Theora.pdf>.

[TRUEHD] Dolby, "Dolby TrueHD (MLP) high-level bitstream description", February 2018, <https://web.archive.org/web/20250810135349/https://developer.dolby.com/globalassets/technology/dolby-truehd/dolbytruehdhighlevelbitstreamdescription.pdf>.

[TTA] Tau Software, "TTA", <https://tausoft.org/en/tta-%d0%be%d0%bf%d0%b8%d1%81%d0%b0%d0%bd%d0%b8%d0%b5-%d1%84%d0%be%d1%80%d0%bc%d0%b0%d1%82%d0%b0/>.

[USF] PARIS, C., Vialle, L., and U. Hammer, "Universal Subtitle Format", 28 November 2010, <https://subtitld.org/en/development/usf>.

[VobSub] MultimediaWiki, "VOBSub", 21 March 2007, <https://wiki.multimedia.cx/index.php?title=VOBsub>.

[VORBIS] Xiph.Org Foundation, "Vorbis I specification", 4 July 2020, <https://xiph.org/vorbis/doc/Vorbis_I_spec.pdf>.

[VP9] Grange, A., de Rivaz, P., and J. Hunt, "VP9 Bitstream & Decoding Process Specification - version 0.7", 22 February 2017, <https://storage.googleapis.com/downloads.webmproject.org/docs/vp9/vp9-bitstream-specification-v0.7-20170222-draft.pdf>.

[WAVEFORMATEX]
Microsoft Corporation, "WAVEFORMATEX structure", 4 April 2021, <https://docs.microsoft.com/en-us/windows/win32/api/mmeapi/ns-mmeapi-waveformatex>.

[WAVPACK] Bryant, D., "WavPack 4 & 5 Binary File / Block Format", 12 April 2020, <https://www.wavpack.com/WavPack5FileFormat.pdf>.

[WebVTT] Pieters, S., Pfeiffer, S., Ed., Jaegenstedt, P., and I. Hickson, "WebVTT: The Web Video Text Tracks Format", W3C Candidate Recommendation, April 2019, <https://www.w3.org/TR/2019/CR-webvtt1-20190404/>.

10.2. Informative References

[AtracAES] Sony Corporate Research Laboratories, "ATRAC: Adaptive Transform Acoustic Coding for MiniDisc", 1 October 1992, <https://www.minidisc.wiki/technology/atrac/aes>.

[atracdenc] Cherednik, D., "atracdenc - ATRAC1 and ATRAC3 Decoder/ Encoder", 12 October 2022, <https://github.com/dcherednik/atracdenc>.

[CTA.861-4] Consumer Technology Association, "Updates to Dynamic HDR Metadata Signaling", CTA 861-4, March 2019, <https://shop.cta.tech/products/updates-to-dynamic-hdr-metadata-signaling>.

[DVD-Video] DVD Forum, "DVD-Books: Part 3 DVD-Video Book", November 1995, <http://www.dvdforum.org/>.

[ITU-T.35] ITU-T, "Procedure for the allocation of ITU-T defined codes for non-standard facilities", ITU-T Recommendation T.35, February 2000, <https://www.itu.int/rec/T-REC-T.35/en>.

[RFC5484] Singer, D., "Associating Time-Codes with RTP Streams", RFC 5484, DOI 10.17487/RFC5484, March 2009, <https://www.rfc-editor.org/info/rfc5484>.

[RFC6386] Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., and Y. Xu, "VP8 Data Format and Decoding Guide", RFC 6386, DOI 10.17487/RFC6386, November 2011, <https://www.rfc-editor.org/info/rfc6386>.

[SMPTE.ST2042-1] SMPTE, "VC-2 Video Compression", ST 2042-1:2022, DOI 10.5594/SMPTE.ST2042-1.2022, 8 December 2022, <https://pub.smpte.org/pub/st2042-1/st2042-1-2022.pdf>.

[VP-ISOBMFF]
Galligan, F., Hughes, K., Inskip, T., and D. Ronca, "VP Codec ISO Media File Format Binding", 31 March 2017, <https://www.webmproject.org/vp9/mp4/>.

[WebM-WebVTT]
Heaney, M. and F. Galligan, "WebVTT in WebM", 1 February 2012, <https://wiki.webmproject.org/webm-metadata/temporal-metadata/webvtt-in-webm>.

[WebMContainer]
The WebM Project, "WebM Container Guidelines", 16 October 2023, <https://www.webmproject.org/docs/container/>.

Authors' Addresses

Steve Lhomme
Email: slhomme@matroska.org

Moritz Bunkus
Email: moritz@bunkus.org

Dave Rice
Email: dave@dericed.com