

cellar
Internet-Draft
Intended status: Standards Track
Expires: July 7, 2018

S. Lhomme

M. Bunkus

D. Rice
January 3, 2018

Matroska Specifications
draft-lhomme-cellar-matroska-04

Abstract

This document defines the Matroska audiovisual container, including definitions of its structural elements, as well as its terminology, vocabulary, and application.

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 July 7, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	9
2.	Status of this document	10
3.	Security Considerations	10
4.	IANA Considerations	10
5.	Notation and Conventions	10
6.	Basis in EBML	11
6.1.	Added Constraints on EBML	11
6.2.	Matroska Design	11
6.2.1.	Language Codes	11
6.2.2.	Physical Types	12
6.2.3.	Block Structure	12
6.2.4.	Lacing	14
7.	Matroska Structure	18
8.	Matroska Schema	26
8.1.	Matroska Additions to Schema Element Attributes	26
8.2.	Matroska Schema	27
8.2.1.	EBMLMaxIDLength Element	27
8.2.2.	EBMLMaxSizeLength Element	27
8.2.3.	Segment Element	28
8.2.4.	SeekHead Element	28
8.2.5.	Seek Element	28
8.2.6.	SeekID Element	29
8.2.7.	SeekPosition Element	29
8.2.8.	Info Element	30
8.2.9.	SegmentUID Element	30
8.2.10.	SegmentFilename Element	30
8.2.11.	PrevUID Element	31
8.2.12.	PrevFilename Element	31
8.2.13.	NextUID Element	32
8.2.14.	NextFilename Element	32
8.2.15.	SegmentFamily Element	33
8.2.16.	ChapterTranslate Element	33
8.2.17.	ChapterTranslateEditionUID Element	33
8.2.18.	ChapterTranslateCodec Element	34
8.2.19.	ChapterTranslateID Element	34
8.2.20.	TimecodeScale Element	35
8.2.21.	Duration Element	35
8.2.22.	DateUTC Element	35
8.2.23.	Title Element	36
8.2.24.	MuxingApp Element	36
8.2.25.	WritingApp Element	37
8.2.26.	Cluster Element	37
8.2.27.	Timecode Element	37
8.2.28.	SilentTracks Element	38
8.2.29.	SilentTrackNumber Element	38
8.2.30.	Position Element	38

8.2.31	PrevSize Element	39
8.2.32	SimpleBlock Element	39
8.2.33	BlockGroup Element	40
8.2.34	Block Element	40
8.2.35	BlockVirtual Element	40
8.2.36	BlockAdditions Element	41
8.2.37	BlockMore Element	41
8.2.38	BlockAddID Element	41
8.2.39	BlockAdditional Element	42
8.2.40	BlockDuration Element	42
8.2.41	ReferencePriority Element	43
8.2.42	ReferenceBlock Element	43
8.2.43	ReferenceVirtual Element	44
8.2.44	CodecState Element	44
8.2.45	DiscardPadding Element	44
8.2.46	Slices Element	45
8.2.47	TimeSlice Element	45
8.2.48	LaceNumber Element	45
8.2.49	FrameNumber Element	46
8.2.50	BlockAdditionID Element	46
8.2.51	Delay Element	47
8.2.52	SliceDuration Element	47
8.2.53	ReferenceFrame Element	48
8.2.54	ReferenceOffset Element	48
8.2.55	ReferenceTimeCode Element	48
8.2.56	EncryptedBlock Element	49
8.2.57	Tracks Element	49
8.2.58	TrackEntry Element	50
8.2.59	TrackNumber Element	50
8.2.60	TrackUID Element	50
8.2.61	TrackType Element	51
8.2.62	FlagEnabled Element	51
8.2.63	FlagDefault Element	52
8.2.64	FlagForced Element	52
8.2.65	FlagLacing Element	53
8.2.66	MinCache Element	53
8.2.67	MaxCache Element	54
8.2.68	DefaultDuration Element	54
8.2.69	DefaultDecodedFieldDuration Element	54
8.2.70	TrackTimecodeScale Element	55
8.2.71	TrackOffset Element	55
8.2.72	MaxBlockAdditionID Element	56
8.2.73	Name Element	56
8.2.74	Language Element	57
8.2.75	LanguageIETF Element	57
8.2.76	CodecID Element	57
8.2.77	CodecPrivate Element	58
8.2.78	CodecName Element	58

8.2.79	AttachmentLink Element	58
8.2.80	CodecSettings Element	59
8.2.81	CodecInfoURL Element	59
8.2.82	CodecDownloadURL Element	60
8.2.83	CodecDecodeAll Element	60
8.2.84	TrackOverlay Element	60
8.2.85	CodecDelay Element	61
8.2.86	SeekPreRoll Element	61
8.2.87	TrackTranslate Element	62
8.2.88	TrackTranslateEditionUID Element	62
8.2.89	TrackTranslateCodec Element	62
8.2.90	TrackTranslateTrackID Element	63
8.2.91	Video Element	63
8.2.92	FlagInterlaced Element	63
8.2.93	FieldOrder Element	64
8.2.94	StereoMode Element	64
8.2.95	AlphaMode Element	65
8.2.96	OldStereoMode Element	65
8.2.97	PixelWidth Element	66
8.2.98	PixelHeight Element	66
8.2.99	PixelCropBottom Element	66
8.2.100	PixelCropTop Element	67
8.2.101	PixelCropLeft Element	67
8.2.102	PixelCropRight Element	68
8.2.103	DisplayWidth Element	68
8.2.104	DisplayHeight Element	68
8.2.105	DisplayUnit Element	69
8.2.106	AspectRatioType Element	69
8.2.107	ColourSpace Element	70
8.2.108	GammaValue Element	70
8.2.109	FrameRate Element	71
8.2.110	Colour Element	71
8.2.111	MatrixCoefficients Element	71
8.2.112	BitsPerChannel Element	72
8.2.113	ChromaSubsamplingHorz Element	72
8.2.114	ChromaSubsamplingVert Element	73
8.2.115	CbSubsamplingHorz Element	73
8.2.116	CbSubsamplingVert Element	73
8.2.117	ChromaSitingHorz Element	74
8.2.118	ChromaSitingVert Element	74
8.2.119	Range Element	75
8.2.120	TransferCharacteristics Element	75
8.2.121	Primaries Element	75
8.2.122	MaxCLL Element	76
8.2.123	MaxFALL Element	76
8.2.124	MasteringMetadata Element	77
8.2.125	PrimaryRChromaticityX Element	77
8.2.126	PrimaryRChromaticityY Element	77

8.2.127	PrimaryGChromaticityX Element	78
8.2.128	PrimaryGChromaticityY Element	78
8.2.129	PrimaryBChromaticityX Element	79
8.2.130	PrimaryBChromaticityY Element	79
8.2.131	WhitePointChromaticityX Element	79
8.2.132	WhitePointChromaticityY Element	80
8.2.133	LuminanceMax Element	80
8.2.134	LuminanceMin Element	81
8.2.135	Projection Element	81
8.2.136	ProjectionType Element	81
8.2.137	ProjectionPrivate Element	82
8.2.138	ProjectionPoseYaw Element	82
8.2.139	ProjectionPosePitch Element	83
8.2.140	ProjectionPoseRoll Element	84
8.2.141	Audio Element	84
8.2.142	SamplingFrequency Element	84
8.2.143	OutputSamplingFrequency Element	85
8.2.144	Channels Element	85
8.2.145	ChannelPositions Element	86
8.2.146	BitDepth Element	86
8.2.147	TrackOperation Element	87
8.2.148	TrackCombinePlanes Element	87
8.2.149	TrackPlane Element	87
8.2.150	TrackPlaneUID Element	88
8.2.151	TrackPlaneType Element	88
8.2.152	TrackJoinBlocks Element	89
8.2.153	TrackJoinUID Element	89
8.2.154	TrickTrackUID Element	89
8.2.155	TrickTrackSegmentUID Element	90
8.2.156	TrickTrackFlag Element	90
8.2.157	TrickMasterTrackUID Element	91
8.2.158	TrickMasterTrackSegmentUID Element	91
8.2.159	ContentEncodings Element	91
8.2.160	ContentEncoding Element	92
8.2.161	ContentEncodingOrder Element	92
8.2.162	ContentEncodingScope Element	93
8.2.163	ContentEncodingType Element	93
8.2.164	ContentCompression Element	94
8.2.165	ContentCompAlgo Element	94
8.2.166	ContentCompSettings Element	95
8.2.167	ContentEncryption Element	95
8.2.168	ContentEncAlgo Element	95
8.2.169	ContentEncKeyID Element	96
8.2.170	ContentSignature Element	96
8.2.171	ContentSigKeyID Element	97
8.2.172	ContentSigAlgo Element	97
8.2.173	ContentSigHashAlgo Element	97
8.2.174	Cues Element	98

8.2.175	CuePoint Element	98
8.2.176	CueTime Element	99
8.2.177	CueTrackPositions Element	99
8.2.178	CueTrack Element	99
8.2.179	CueClusterPosition Element	100
8.2.180	CueRelativePosition Element	100
8.2.181	CueDuration Element	101
8.2.182	CueBlockNumber Element	101
8.2.183	CueCodecState Element	101
8.2.184	CueReference Element	102
8.2.185	CueRefTime Element	102
8.2.186	CueRefCluster Element	102
8.2.187	CueRefNumber Element	103
8.2.188	CueRefCodecState Element	103
8.2.189	Attachments Element	104
8.2.190	AttachedFile Element	104
8.2.191	FileDescription Element	105
8.2.192	FileName Element	105
8.2.193	FileMimeType Element	105
8.2.194	FileData Element	106
8.2.195	FileUID Element	106
8.2.196	FileReferral Element	106
8.2.197	FileUsedStartTime Element	107
8.2.198	FileUsedEndTime Element	107
8.2.199	Chapters Element	108
8.2.200	EditionEntry Element	108
8.2.201	EditionUID Element	108
8.2.202	EditionFlagHidden Element	109
8.2.203	EditionFlagDefault Element	109
8.2.204	EditionFlagOrdered Element	110
8.2.205	ChapterAtom Element	110
8.2.206	ChapterUID Element	110
8.2.207	ChapterStringUID Element	111
8.2.208	ChapterTimeStart Element	111
8.2.209	ChapterTimeEnd Element	112
8.2.210	ChapterFlagHidden Element	112
8.2.211	ChapterFlagEnabled Element	113
8.2.212	ChapterSegmentUID Element	113
8.2.213	ChapterSegmentEditionUID Element	114
8.2.214	ChapterPhysicalEquiv Element	114
8.2.215	ChapterTrack Element	114
8.2.216	ChapterTrackNumber Element	115
8.2.217	ChapterDisplay Element	115
8.2.218	ChapString Element	116
8.2.219	ChapLanguage Element	116
8.2.220	ChapLanguageIETF Element	116
8.2.221	ChapCountry Element	117
8.2.222	ChapProcess Element	117

8.2.223.	ChapProcessCodecID Element	118
8.2.224.	ChapProcessPrivate Element	118
8.2.225.	ChapProcessCommand Element	118
8.2.226.	ChapProcessTime Element	119
8.2.227.	ChapProcessData Element	119
8.2.228.	Tags Element	120
8.2.229.	Tag Element	120
8.2.230.	Targets Element	120
8.2.231.	TargetTypeValue Element	121
8.2.232.	TargetType Element	121
8.2.233.	TagTrackUID Element	122
8.2.234.	TagEditionUID Element	122
8.2.235.	TagChapterUID Element	122
8.2.236.	TagAttachmentUID Element	123
8.2.237.	SimpleTag Element	123
8.2.238.	TagName Element	123
8.2.239.	TagLanguage Element	124
8.2.240.	TagLanguageIETF Element	124
8.2.241.	TagDefault Element	125
8.2.242.	TagString Element	125
8.2.243.	TagBinary Element	126
9.	Matroska Element Ordering Guidelines	126
9.1.	Top-Level Elements	126
9.2.	CRC-32	127
9.3.	SeekHead	127
9.4.	Cues (index)	127
9.5.	Info	127
9.6.	Chapters	128
9.7.	Attachments	128
9.8.	Tags	128
9.9.	Optimum layout from a muxer	128
9.10.	Optimum layout after editing tags	129
9.11.	Optimum layout with Cues at the front	129
9.12.	Cluster Timecode	129
10.	Chapters	130
10.1.	Edition and Chapter Flags	130
10.1.1.	Chapter Flags	130
10.1.2.	Edition Flags	130
10.2.	Menu features	132
10.2.1.	Matroska Script (0)	132
10.2.2.	DVD menu (1)	133
10.3.	Example 1 : basic chaptering	134
10.4.	Example 2 : nested chapters	136
10.4.1.	The Micronauts "Bleep To Bleep"	136
11.	Attachments	139
11.1.	Introduction	139
11.2.	Cover Art	139
12.	Cues	140

12.1.	Introduction	140
12.2.	Recommendations	140
13.	Matroska Streaming	141
13.1.	File Access	141
13.2.	Livestreaming	142
14.	Menu Specifications	142
14.1.	Introduction	142
14.2.	Requirements	143
14.2.1.	Highlights/Hotspots	143
14.2.2.	Playback features	144
14.2.3.	Player requirements	144
14.3.	Working Graph	144
14.4.	Ideas	145
14.5.	Data Structure	145
15.	Unknown elements	145
16.	Default Values	145
17.	DefaultDecodedFieldDuration	145
18.	Encryption	146
19.	Image cropping	147
20.	Matroska versioning	147
21.	MIME Types	148
22.	Segment Position	148
22.1.	Segment Position Exception	148
22.2.	Example of Segment Position	148
23.	Linked Segments	149
23.1.	Hard Linking	149
23.2.	Medium Linking	150
23.3.	Soft Linking	150
24.	Track Flags	151
24.1.	Default flag	151
24.2.	Forced flag	151
24.3.	Track Operation	151
24.4.	Overlay Track	152
24.5.	Multi-planar and 3D videos	152
25.	Timecodes	153
25.1.	Timecode Types	153
25.2.	Block Timecodes	153
25.3.	Raw Timecode	153
25.4.	TimecodeScale	153
25.5.	TimecodeScale Rounding	154
25.6.	TrackTimecodeScale	156
26.1.	URIs	157
Authors'	Addresses	158

1. Introduction

Matroska aims to become THE standard of multimedia container formats. It was derived from a project called MCF [[1](#)], but differentiates from it significantly because it is based on EBML [[2](#)] (Extensible Binary Meta Language), a binary derivative of XML. EBML enables significant advantages in terms of future format extensibility, without breaking file support in old parsers.

First, it is essential to clarify exactly "What an Audio/Video container is", to avoid any misunderstandings:

- o It is NOT a video or audio compression format (codec)
- o It is an envelope for which there can be many audio, video and subtitles streams, allowing the user to store a complete movie or CD in a single file.

Matroska is designed with the future in mind. It incorporates features like:

- o Fast seeking in the file
- o Chapter entries
- o Full metadata (tags) support
- o Selectable subtitle/audio/video streams
- o Modularly expandable
- o Error resilience (can recover playback even when the stream is damaged)
- o Streamable over the internet and local networks (HTTP, CIFS, FTP, etc)
- o Menus (like DVDs have)

Matroska is an open standards project. This means for personal use it is absolutely free to use and that the technical specifications describing the bitstream are open to everybody, even to companies that would like to support it in their products.

2. Status of this document

This document is a work-in-progress specification defining the Matroska file format as part of the IETF Cellar working group [3]. But since it's quite complete it is used as a reference for the development of libmatroska. Legacy versions of the specification can be found here [4] (PDF doc by Alexander Noe -- outdated).

For a simplified diagram of the layout of a Matroska file, see the Diagram page [5].

A more refined and detailed version of the EBML specifications is being worked on here [6].

The table found below is now generated from the "source" of the Matroska specification. This XML file [7] is also used to generate the semantic data used in libmatroska and libmatroska2. We encourage anyone to use and monitor its changes so your code is spec-proof and always up to date.

Note that versions 1, 2 and 3 have been finalized. Version 4 is currently work in progress. There MAY be further additions to v4.

3. Security Considerations

Matroska inherits security considerations from EBML.

Attacks on a "Matroska Reader" could include:

- o Storage of a arbitrary and potentially executable data within an "Attachment Element". "Matroska Readers" that extract or use data from Matroska Attachments SHOULD check that the data adheres to expectations.
- o A "Matroska Attachment" with an inaccurate mime-type.

4. IANA Considerations

To be determined.

5. Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [8].

This document defines specific terms in order to define the format and application of "Matroska". Specific terms are defined below:

"Matroska": a multimedia container format based on EBML (Extensible Binary Meta Language)

"Matroska Reader": A "Matroska Reader" is a data parser that interprets the semantics of a Matroska document and creates a way for programs to use "Matroska".

"Matroska Player": A "Matroska Player" is a "Matroska Reader" with a primary purpose of playing audiovisual files, including "Matroska" documents.

[6.](#) Basis in EBML

Matroska is a Document Type of EBML (Extensible Binary Meta Language). This specification is dependent on the EBML Specification [\[9\]](#). For an understanding of Matroska's EBML Schema, see in particular the sections of the EBML Specification covering EBML Element Types [\[10\]](#), EBML Schema [\[11\]](#), and EBML Structure [\[12\]](#).

[6.1.](#) Added Constraints on EBML

As an EBML Document Type, Matroska adds the following constraints to the EBML specification.

- o The "docType" of the "EBML Header" MUST be 'matroska'.
- o The "EBMLMaxIDLength" of the "EBML Header" MUST be "4".
- o The "EBMLMaxSizeLength" of the "EBML Header" MUST be between "1" and "8" inclusive.

[6.2.](#) Matroska Design

All top-levels elements (Segment and direct sub-elements) are coded on 4 octets, i.e. class D elements.

[6.2.1.](#) Language Codes

Matroska from version 1 through 3 uses language codes that can be either the 3 letters bibliographic ISO-639-2 [\[13\]](#) form (like "fre" for french), or such a language code followed by a dash and a country code for specialities in languages (like "fre-ca" for Canadian French). The "ISO 639-2 Language Elements" are "Language Element", "TagLanguage Element", and "ChapLanguage Element".

Starting in Matroska version 4, either "ISO 639-2" or [BCP 47](#) [\[14\]](#) MAY be used, although "[BCP 47](#)" is RECOMMENDED. The "[BCP 47](#) Language Elements" are "LanguageIETF Element", "TagLanguageIETF Element", and

"ChapLanguageIETF Element". If a "[BCP 47](#) Language Element" and an "ISO 639-2 Language Element" are used within the same "Parent Element", then the "ISO 639-2 Language Element" MUST be ignored and precedence given to the "[BCP 47](#) Language Element".

Country codes are the same as used for internet domains [[15](#)].

[6.2.2.](#) Physical Types

Each level can have different meanings for audio and video. The ORIGINAL_MEDIUM tag can be used to specify a string for ChapterPhysicalEquiv = 60. Here is the list of possible levels for both audio and video :

ChapterPhysicalEquiv	Audio	Video	Comment
70	SET / PACKAGE	SET / PACKAGE	the collection of different media
60	CD / 12" / 10" / 7" / TAPE / MINIDISC / DAT	DVD / VHS / LASERDISC	the physical medium like a CD or a DVD
50	SIDE	SIDE	when the original medium (LP/DVD) has different sides
40	-	LAYER	another physical level on DVDs
30	SESSION	SESSION	as found on CDs and DVDs
20	TRACK	-	as found on audio CDs
10	INDEX	-	the first logical level of the side/medium

[6.2.3.](#) Block Structure

Size = 1 + (1-8) + 4 + (4 + (4)) octets. So from 6 to 21 octets.

Bit 0 is the most significant bit.

Frames using references SHOULD be stored in "coding order". That means the references first and then the frames referencing them. A consequence is that timecodes MAY NOT be consecutive. But a frame with a past timecode MUST reference a frame already known, otherwise it's considered bad/void.

There can be many Blocks in a BlockGroup provided they all have the same timecode. It is used with different parts of a frame with different priorities.

[6.2.3.1.](#) Block Header

Offset	Player	Description
0x00+	MUST	Track Number (Track Entry). It is coded in EBML like form (1 octet if the value is < 0x80, 2 if < 0x4000, etc) (most significant bits set to increase the range).
0x01+	MUST	Timecode (relative to Cluster timecode, signed int16)

[6.2.3.2.](#) Block Header Flags

Offset	Bit	Player	Description
0x03+	0-3	-	Reserved, set to 0
0x03+	4	-	Invisible, the codec SHOULD decode this frame but not display it
0x03+	5-6	MUST	Lacing
			* 00 : no lacing
			* 01 : Xiph lacing
			* 11 : EBML lacing
			* 10 : fixed-size lacing
0x03+	7	-	not used

[6.2.3.3.](#) Laced Data

When lacing bit is set.

Offset	Player	Description
0x00	MUST	Number of frames in the lace-1 (uint8)
0x01 / 0xXX	MUST*	Lace-coded size of each frame of the lace, except for the last one (multiple uint8). *This is not used with Fixed-size lacing as it is calculated automatically from (total size of lace) / (number of frames in lace).

For (possibly) Laced Data

Offset	Player	Description
0x00	MUST	Consecutive laced frames

6.2.4. Lacing

Lacing is a mechanism to save space when storing data. It is typically used for small blocks of data (referred to as frames in Matroska). There are 3 types of lacing:

1. Xiph, inspired by what is found in the Ogg container
2. EBML, which is the same with sizes coded differently
3. fixed-size, where the size is not coded

For example, a user wants to store 3 frames of the same track. The first frame is 800 octets long, the second is 500 octets long and the third is 1000 octets long. As these data are small, they can be stored in a lace to save space. They will then be stored in the same block as follows:

6.2.4.1. Xiph lacing

- o Block head (with lacing bits set to 01)
- o Lacing head: Number of frames in the lace -1, i.e. 2 (the 800 and 500 octets one)
- o Lacing sizes: only the 2 first ones will be coded, 800 gives 255;255;255;35, 500 gives 255;245. The size of the last frame is deduced from the total size of the Block.

- o Data in frame 1
- o Data in frame 2
- o Data in frame 3

A frame with a size multiple of 255 is coded with a 0 at the end of the size, for example 765 is coded 255;255;255;0.

[6.2.4.2.](#) EBML lacing

In this case, the size is not coded as blocks of 255 bytes, but as a difference with the previous size and this size is coded as in EBML. The first size in the lace is unsigned as in EBML. The others use a range shifting to get a sign on each value:

Bit Representation	Value
1xxx xxxx	value $-(2^6-1)$ to 2^6-1 (ie 0 to 2^7-2 minus 2^6-1 , half of the range)
01xx xxxx xxxx xxxx	value $-(2^{13}-1)$ to $2^{13}-1$
001x xxxx xxxx xxxx xxxx xxxx	value $-(2^{20}-1)$ to $2^{20}-1$
0001 xxxx xxxx xxxx xxxx xxxx	value $-(2^{27}-1)$ to $2^{27}-1$
xxxx xxxx	
0000 1xxx xxxx xxxx xxxx xxxx	value $-(2^{34}-1)$ to $2^{34}-1$
xxxx xxxx xxxx xxxx	
0000 01xx xxxx xxxx xxxx xxxx	value $-(2^{41}-1)$ to $2^{41}-1$
xxxx xxxx xxxx xxxx xxxx xxxx	
0000 001x xxxx xxxx xxxx xxxx	value $-(2^{48}-1)$ to $2^{48}-1$
xxxx xxxx xxxx xxxx xxxx xxxx	
xxxx xxxx	

- o Block head (with lacing bits set to 11)
- o Lacing head: Number of frames in the lace -1, i.e. 2 (the 800 and 400 octets one)
- o Lacing sizes: only the 2 first ones will be coded, 800 gives 0x320 0x4000 = 0x4320, 500 is coded as -300 : - 0x12C + 0x1FFF + 0x4000 = 0x5ED3. The size of the last frame is deduced from the total size of the Block.
- o Data in frame 1
- o Data in frame 2

- o Data in frame 3

[6.2.4.3.](#) Fixed-size lacing

In this case, only the number of frames in the lace is saved, the size of each frame is deduced from the total size of the Block. For example, for 3 frames of 800 octets each:

- o Block head (with lacing bits set to 10)
- o Lacing head: Number of frames in the lace -1, i.e. 2
- o Data in frame 1
- o Data in frame 2
- o Data in frame 3

[6.2.4.4.](#) SimpleBlock Structure

The "SimpleBlock" is inspired by the [Section 6.2.3](#). The main differences are the added Keyframe flag and Discardable flag. Otherwise everything is the same.

Size = 1 + (1-8) + 4 + (4 + (4)) octets. So from 6 to 21 octets.

Bit 0 is the most significant bit.

Frames using references SHOULD be stored in "coding order". That means the references first and then the frames referencing them. A consequence is that timecodes MAY NOT be consecutive. But a frame with a past timecode MUST reference a frame already known, otherwise it's considered bad/void.

There can be many "Block Elements" in a "BlockGroup" provided they all have the same timecode. It is used with different parts of a frame with different priorities.

[6.2.4.4.1.](#) SimpleBlock Header

Offset	Player	Description
0x00+	MUST	Track Number (Track Entry). It is coded in EBML like form (1 octet if the value is < 0x80, 2 if < 0x4000, etc) (most significant bits set to increase the range).
0x01+	MUST	Timecode (relative to Cluster timecode, signed int16)

6.2.4.4.2. SimpleBlock Header Flags

Offset	Bit	Player	Description
0x03+	0	-	Keyframe, set when the Block contains only keyframes
0x03+	1-3	-	Reserved, set to 0
0x03+	4	-	Invisible, the codec SHOULD decode this frame but not display it
0x03+	5-6	MUST	Lacing * 00 : no lacing * 01 : Xiph lacing * 11 : EBML lacing * 10 : fixed-size lacing
0x03+	7	-	Discardable, the frames of the Block can be discarded during playing if needed

6.2.4.5. Laced Data

When lacing bit is set.

Offset	Player	Description
0x00	MUST	Number of frames in the lace-1 (uint8)
0x01 / 0xFF	MUST*	Lace-coded size of each frame of the lace, except for the last one (multiple uint8). *This is not used with Fixed-size lacing as it is calculated automatically from (total size of lace) / (number of frames in lace).

For (possibly) Laced Data


```

+-----+-----+-----+
| Offset | Player | Description          |
+-----+-----+-----+
| 0x00   | MUST   | Consecutive laced frames |
+-----+-----+-----+

```

7. Matroska Structure

A Matroska file MUST be composed of at least one "EBML Document" using the "Matroska Document Type". Each "EBML Document" MUST start with an "EBML Header" and MUST be followed by the "EBML Root Element", defined as "Segment" in Matroska. Matroska defines several "Top Level Elements" which MAY occur within the "Segment".

As an example, a simple Matroska file consisting of a single "EBML Document" could be represented like this:

- o "EBML Header"
- o "Segment"

A more complex Matroska file consisting of an "EBML Stream" (consisting of two "EBML Documents") could be represented like this:

- o "EBML Header"
- o "Segment"
- o "EBML Header"
- o "Segment"

The following diagram represents a simple Matroska file, comprised of an "EBML Document" with an "EBML Header", a "Segment Element" (the "Root Element"), and all eight Matroska "Top Level Elements". In the following diagrams of this section, horizontal spacing expresses a parent-child relationship between Matroska Elements (e.g. the "Info Element" is contained within the "Segment Element") whereas vertical alignment represents the storage order within the file.

+-----+		
EBML Header		
+-----+		
Segment	SeekHead	

	Info	

	Tracks	

	Chapters	

	Cluster	

	Cues	

	Attachments	

	Tags	
+-----+		

The Matroska "EBML Schema" defines eight "Top Level Elements": "SeekHead", "Info", "Tracks", "Chapters", "Cluster", "Cues", "Attachments", and "Tags".

The "SeekHead Element" (also known as "MetaSeek") contains an index of "Top Level Elements" locations within the "Segment". Use of the "SeekHead Element" is RECOMMENDED. Without a "SeekHead Element", a Matroska parser would have to search the entire file to find all of the other "Top Level Elements". This is due to Matroska's flexible ordering requirements; for instance, it is acceptable for the "Chapters Element" to be stored after the "Cluster Elements".

+-----+			
SeekHead	Seek	SeekID	

		SeekPosition	
+-----+			

Representation of a SeekHead Element.

The "Info Element" contains vital information for identifying the whole "Segment". This includes the title for the "Segment", a randomly generated unique identifier, and the unique identifier(s) of any linked "Segment Elements".

+-----+		
	Info	SegmentUID

		SegmentFilename

		PrevUID

		PrevFilename

		NextUID

		NextFilename

		SegmentFamily

		ChapterTranslate

		TimecodeScale

		Duration

		DateUTC

		Title

		MuxingApp

		WritingApp

Representation of an Info Element and its Child Elements.

The "Tracks Element" defines the technical details for each track and can store the name, number, unique identifier, language and type (audio, video, subtitles, etc.) of each track. For example, the "Tracks Element" MAY store information about the resolution of a video track or sample rate of an audio track.

The "Tracks Element" MUST identify all the data needed by the codec to decode the data of the specified track. However, the data required is contingent on the codec used for the track. For example, a "Track Element" for uncompressed audio only requires the audio bit rate to be present. A codec such as AC-3 would require that the "CodecID Element" be present for all tracks, as it is the primary way to identify which codec to use to decode the track.

+-----+			
Tracks	TrackEntry	TrackNumber	

		TrackUID	

		TrackType	

		Name	

		Language	

		CodecID	

		CodecPrivate	

		CodecName	

		Video	FlagInterlaced

			FieldOrder

			StereoMode

			AlphaMode

			PixelWidth

			PixelHeight

			DisplayWidth

			DisplayHeight

			AspectRatioType

			Color

		Audio	SamplingFrequency

			Channels

			BitDepth

Representation of the Tracks Element and a selection of its
Descendant Elements.

The "Chapters Element" lists all of the chapters. Chapters are a way to set predefined points to jump to in video or audio.

+-----+				
Chapters	Edition	EditionUID		
	Entry	-----		
		EditionFlagHidden		

		EditionFlagDefault		

		EditionFlagOrdered		
		-----+		
		ChapterAtom	ChapterUID	

			ChapterStringUID	

			ChapterTimeStart	

			ChapterTimeEnd	

			ChapterFlagHidden	
			-----+	
			ChapterDisplay	ChapString

				ChapLanguage
+-----+				

Representation of the Chapters Element and a selection of its
Descendant Elements.

"Cluster Elements" contain the content for each track, e.g. video frames. A Matroska file SHOULD contain at least one "Cluster Element". The "Cluster Element" helps to break up "SimpleBlock" or "BlockGroup Elements" and helps with seeking and error protection. It is RECOMMENDED that the size of each individual "Cluster Element" be limited to store no more than 5 seconds or 5 megabytes. Every "Cluster Element" MUST contain a "Timecode Element". This SHOULD be the "Timecode Element" used to play the first "Block" in the "Cluster Element". There SHOULD be one or more "BlockGroup" or "SimpleBlock Element" in each "Cluster Element". A "BlockGroup Element" MAY contain a "Block" of data and any information relating directly to that "Block".

+-----+		
Cluster	Timecode	

	SilentTracks	

	Position	

	PrevSize	

	SimpleBlock	

	BlockGroup	

	EncryptedBlock	
+-----+		

Representation of a Cluster Element and its immediate Child Elements.

+-----+			
Block	Portion of	Data Type	
	a Block	- Bit Flag	
	-----	-----	
	Header	TrackNumber	

		Timecode	

		Flags	
		- Gap	
		- Lacing	
		- Reserved	
	-----	-----	
	Optional	FrameSize	
	-----	-----	
	Data	Frame	
+-----+			

Representation of the Block Element structure.

Each "Cluster" MUST contain exactly one "Timecode Element". The "Timecode Element" value MUST be stored once per "Cluster". The "Timecode Element" in the "Cluster" is relative to the entire "Segment". The "Timecode Element" SHOULD be the first "Element" in the "Cluster".

Additionally, the "Block" contains an offset that, when added to the "Cluster"'s "Timecode Element" value, yields the "Block"'s effective timecode. Therefore, timecode in the "Block" itself is relative to the "Timecode Element" in the "Cluster". For example, if the

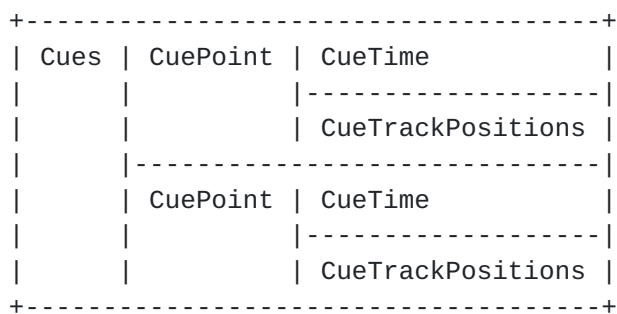
"Timecode Element" in the "Cluster" is set to 10 seconds and a "Block" in that "Cluster" is supposed to be played 12 seconds into the clip, the timecode in the "Block" would be set to 2 seconds.

The "ReferenceBlock" in the "BlockGroup" is used instead of the basic "P-frame"/"B-frame" description. Instead of simply saying that this "Block" depends on the "Block" directly before, or directly afterwards, the "Timecode" of the necessary "Block" is used. Because there can be as many "ReferenceBlock Elements" as necessary for a "Block", it allows for some extremely complex referencing.

The "Cues Element" is used to seek when playing back a file by providing a temporal index for some of the "Tracks". It is similar to the "SeekHead Element", but used for seeking to a specific time when playing back the file. It is possible to seek without this element, but it is much more difficult because a "Matroska Reader" would have to 'hunt and peck' through the file looking for the correct timecode.

The "Cues Element" SHOULD contain at least one "CuePoint Element". Each "CuePoint Element" stores the position of the "Cluster" that contains the "BlockGroup" or "SimpleBlock Element". The timecode is stored in the "CueTime Element" and location is stored in the "CueTrackPositions Element".

The "Cues Element" is flexible. For instance, "Cues Element" can be used to index every single timecode of every "Block" or they can be indexed selectively. For video files, it is RECOMMENDED to index at least the keyframes of the video track.



Representation of a Cues Element and two levels of its Descendant Elements.

The "Attachments Element" is for attaching files to a Matroska file such as pictures, webpages, programs, or even the codec needed to play back the file.

+-----+		
Attachments	AttachedFile	FileDescription

		FileName

		FileMimeType

		FileData

		FileUID

		FileName

		FileReferral

		FileUsedStartTime

		FileUsedEndTime

+-----+		

Representation of a Attachments Element.

The "Tags Element" contains metadata that describes the "Segment" and potentially its "Tracks", "Chapters", and "Attachments". Each "Track" or "Chapter" that those tags applies to has its UID listed in the "Tags". The "Tags" contain all extra information about the file: scriptwriter, singer, actors, directors, titles, edition, price, dates, genre, comments, etc. Tags can contain their values in multiple languages. For example, a movie's "title" "Tag" might contain both the original English title as well as the title it was released as in Germany.

+-----+-----+-----+-----+			
Tags	Tag	Targets	TargetTypeValue

			TargetType

			TagTrackUID

			TagEditionUID

			TagChapterUID

			TagAttachmentUID

		SimpleTag	TagName

			TagLanguage

			TagDefault

			TagString

			TagBinary

			SimpleTag
+-----+-----+-----+-----+			

Representation of a Tags Element and three levels of its Children Elements.

8. Matroska Schema

This specification includes an "EBML Schema" which defines the Elements and structure of Matroska as an EBML Document Type. The EBML Schema defines every valid Matroska element in a manner defined by the EBML specification.

8.1. Matroska Additions to Schema Element Attributes

In addition to the EBML Schema definition provided by the EBML Specification, Matroska adds the following additional attributes:

attribute	required	definition
name		
webm	No	A boolean to express if the Matroska Element is also supported within version 2 of the "webm" specification. Please consider the webm specification [16] as the authoritative on "webm".

8.2. Matroska Schema

Here the definition of each Matroska Element is provided.

% concatenate with Matroska EBML Schema converted to markdown %

8.2.1. EBMLMaxIDLength Element

name: "EBMLMaxIDLength"

path: "1*1(\EBML\EBMLMaxIDLength)"

id: "0x42F2"

minOccurs: "1"

maxOccurs: "1"

range: "4"

default: "4"

type: "uinteger"

8.2.2. EBMLMaxSizeLength Element

name: "EBMLMaxSizeLength"

path: "1*1(\EBML\EBMLMaxSizeLength)"

id: "0x42F3"

minOccurs: "1"

maxOccurs: "1"

range: "1-8"

default: "8"

type: "uinteger"

8.2.3. Segment Element

name: "Segment"

path: "1*1(\Segment)"

id: "0x18538067"

minOccurs: "1"

maxOccurs: "1"

type: "master"

unknownsizeallowed: "1"

minver: "1"

documentation: The Root Element that contains all other Top-Level Elements (Elements defined only at Level 1). A Matroska file is composed of 1 Segment.

8.2.4. SeekHead Element

name: "SeekHead"

path: "0*2(\Segment\SeekHead)"

id: "0x114D9B74"

maxOccurs: "2"

type: "master"

minver: "1"

documentation: Contains the Segment Position of other Top-Level Elements.

8.2.5. Seek Element

name: "Seek"

path: "1*(\Segment\SeekHead\Seek)"

id: "0x4DBB"

minOccurs: "1"

type: "master"

minver: "1"

documentation: Contains a single seek entry to an EBML Element.

8.2.6. SeekID Element

name: "SeekID"

path: "1*1(\Segment\SeekHead\Seek\SeekID)"

id: "0x53AB"

minOccurs: "1"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: The binary ID corresponding to the Element name.

8.2.7. SeekPosition Element

name: "SeekPosition"

path: "1*1(\Segment\SeekHead\Seek\SeekPosition)"

id: "0x53AC"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: The Segment Position of the Element.

8.2.8. Info Element

name: "Info"

path: "1*(\Segment\Info)"

id: "0x1549A966"

minOccurs: "1"

type: "master"

minver: "1"

definition: Contains general information about the Segment.

8.2.9. SegmentUID Element

name: "SegmentUID"

path: "0*1(\Segment\Info\SegmentUID)"

id: "0x73A4"

maxOccurs: "1"

range: "not 0"

size: "16"

type: "binary"

minver: "1"

definition: A randomly generated unique ID to identify the Segment amongst many others (128 bits).

usage notes: If the Segment is a part of a Linked Segment then this Element is REQUIRED.

8.2.10. SegmentFilename Element

name: "SegmentFilename"

path: "0*1(\Segment\Info\SegmentFilename)"

id: "0x7384"

maxOccurs: "1"

type: "utf-8"

minver: "1"

definition: A filename corresponding to this Segment.

8.2.11. PrevUID Element

name: "PrevUID"

path: "0*1(\Segment\Info\PrevUID)"

id: "0x3CB923"

maxOccurs: "1"

size: "16"

type: "binary"

minver: "1"

definition: A unique ID to identify the previous Segment of a Linked Segment (128 bits).

usage notes: If the Segment is a part of a Linked Segment that uses Hard Linking then either the PrevUID or the NextUID Element is REQUIRED. If a Segment contains a PrevUID but not a NextUID then it MAY be considered as the last Segment of the Linked Segment. The PrevUID MUST NOT be equal to the SegmentUID.

8.2.12. PrevFilename Element

name: "PrevFilename"

path: "0*1(\Segment\Info\PrevFilename)"

id: "0x3C83AB"

maxOccurs: "1"

type: "utf-8"

minver: "1"

definition: A filename corresponding to the file of the previous Linked Segment.

usage notes: Provision of the previous filename is for display convenience, but PrevUID SHOULD be considered authoritative for identifying the previous Segment in a Linked Segment.

8.2.13. NextUID Element

name: "NextUID"

path: "0*1(\Segment\Info\NextUID)"

id: "0x3EB923"

maxOccurs: "1"

size: "16"

type: "binary"

minver: "1"

definition: A unique ID to identify the next Segment of a Linked Segment (128 bits).

usage notes: If the Segment is a part of a Linked Segment that uses Hard Linking then either the PrevUID or the NextUID Element is REQUIRED. If a Segment contains a NextUID but not a PrevUID then it MAY be considered as the first Segment of the Linked Segment. The NextUID MUST NOT be equal to the SegmentUID.

8.2.14. NextFilename Element

name: "NextFilename"

path: "0*1(\Segment\Info\NextFilename)"

id: "0x3E83BB"

maxOccurs: "1"

type: "utf-8"

minver: "1"

definition: A filename corresponding to the file of the next Linked Segment.

usage notes: Provision of the next filename is for display convenience, but NextUID SHOULD be considered authoritative for identifying the Next Segment.

8.2.15. SegmentFamily Element

name: "SegmentFamily"

path: "0*(\\Segment\\Info\\SegmentFamily)"

id: "0x4444"

size: "16"

type: "binary"

minver: "1"

definition: A randomly generated unique ID that all Segments of a Linked Segment MUST share (128 bits).

usage notes: If the Segment is a part of a Linked Segment that uses Soft Linking then this Element is REQUIRED.

8.2.16. ChapterTranslate Element

name: "ChapterTranslate"

path: "0*(\\Segment\\Info\\ChapterTranslate)"

id: "0x6924"

type: "master"

minver: "1"

documentation: A tuple of corresponding ID used by chapter codecs to represent this Segment.

8.2.17. ChapterTranslateEditionUID Element

name: "ChapterTranslateEditionUID"

path: "0*(\\Segment\\Info\\ChapterTranslate\\ChapterTranslateEditionUID)"

id: "0x69FC"

type: "uinteger"

minver: "1"

documentation: Specify an edition UID on which this correspondance applies. When not specified, it means for all editions found in the Segment.

8.2.18. ChapterTranslateCodec Element

name: "ChapterTranslateCodec"

path: "1*1(\Segment\Info\ChapterTranslate\ChapterTranslateCodec)"

id: "0x69BF"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: The chapter codec

8.2.19. ChapterTranslateID Element

name: "ChapterTranslateID"

path: "1*1(\Segment\Info\ChapterTranslate\ChapterTranslateID)"

id: "0x69A5"

minOccurs: "1"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: The binary value used to represent this Segment in the chapter codec data. The format depends on the ChapProcessCodecID used.

8.2.20. TimecodeScale Element

name: "TimecodeScale"
path: "1*1(\Segment\Info\TimecodeScale)"
id: "0x2AD7B1"
minOccurs: "1"
maxOccurs: "1"
range: "not 0"
default: "1000000"
type: "uinteger"
minver: "1"

documentation: Timestamp scale in nanoseconds (1.000.000 means all timestamps in the Segment are expressed in milliseconds).

8.2.21. Duration Element

name: "Duration"
path: "0*1(\Segment\Info\Duration)"
id: "0x4489"
maxOccurs: "1"
range: "> 0x0p+0"
type: "float"
minver: "1"

definition: Duration of the Segment in nanoseconds based on TimecodeScale.

8.2.22. DateUTC Element

name: "DateUTC"
path: "0*1(\Segment\Info\DateUTC)"

id: "0x4461"

maxOccurs: "1"

type: "date"

minver: "1"

documentation: The date and time that the Segment was created by the muxing application or library.

8.2.23. Title Element

name: "Title"

path: "0*1(\Segment\Info\Title)"

id: "0x7BA9"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: General name of the Segment.

8.2.24. MuxingApp Element

name: "MuxingApp"

path: "1*1(\Segment\Info\MuxingApp)"

id: "0x4D80"

minOccurs: "1"

maxOccurs: "1"

type: "utf-8"

minver: "1"

definition: Muxing application or library (example: "libmatroska-0.4.3").

usage notes: Include the full name of the application or library followed by the version number.

[8.2.25.](#) **WritingApp Element**

name: "WritingApp"

path: "1*1(\Segment\Info\WritingApp)"

id: "0x5741"

minOccurs: "1"

maxOccurs: "1"

type: "utf-8"

minver: "1"

definition: Writing application (example: "mkvmerge-0.3.3").

usage notes: Include the full name of the application followed by the version number.

[8.2.26.](#) **Cluster Element**

name: "Cluster"

path: "0*(\Segment\Cluster)"

id: "0x1F43B675"

type: "master"

unknownsizeallowed: "1"

minver: "1"

documentation: The Top-Level Element containing the (monolithic) Block structure.

[8.2.27.](#) **Timecode Element**

name: "Timecode"

path: "1*1(\Segment\Cluster\Timecode)"

id: "0xE7"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: Absolute timestamp of the cluster (based on TimecodeScale).

8.2.28. SilentTracks Element

name: "SilentTracks"

path: "0*1(\Segment\Cluster\SilentTracks)"

id: "0x5854"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: The list of tracks that are not used in that part of the stream. It is useful when using overlay tracks on seeking or to decide what track to use.

8.2.29. SilentTrackNumber Element

name: "SilentTrackNumber"

path: "0*(\Segment\Cluster\SilentTracks\SilentTrackNumber)"

id: "0x58D7"

type: "uinteger"

minver: "1"

documentation: One of the track number that are not used from now on in the stream. It could change later if not specified as silent in a further Cluster.

8.2.30. Position Element

name: "Position"

path: "0*1(\Segment\Cluster\Position)"

id: "0xA7"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: The Segment Position of the Cluster in the Segment (0 in live broadcast streams). It might help to resynchronise offset on damaged streams.

8.2.31. PrevSize Element

name: "PrevSize"

path: "0*1(\Segment\Cluster\PrevSize)"

id: "0xAB"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: Size of the previous Cluster, in octets. Can be useful for backward playing.

8.2.32. SimpleBlock Element

name: "SimpleBlock"

path: "0*(\Segment\Cluster\SimpleBlock)"

id: "0xA3"

type: "binary"

minver: "2"

documentation: Similar to Block but without all the extra information, mostly used to reduced overhead when no extra feature is needed. (see SimpleBlock Structure)

8.2.33. BlockGroup Element

name: "BlockGroup"

path: "0*(\Segment\Cluster\BlockGroup)"

id: "0xA0"

type: "master"

minver: "1"

documentation: Basic container of information containing a single Block and information specific to that Block.

8.2.34. Block Element

name: "Block"

path: "1*1(\Segment\Cluster\BlockGroup\Block)"

id: "0xA1"

minOccurs: "1"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: Block containing the actual data to be rendered and a timestamp relative to the Cluster Timecode. (see Block Structure)

8.2.35. BlockVirtual Element

name: "BlockVirtual"

path: "0*1(\Segment\Cluster\BlockGroup\BlockVirtual)"

id: "0xA2"

maxOccurs: "1"

type: "binary"

minver: "0"

maxver: "0"

documentation: A Block with no data. It MUST be stored in the stream at the place the real Block would be in display order. (see Block Virtual)

8.2.36. BlockAdditions Element

name: "BlockAdditions"

path: "0*1(\Segment\Cluster\BlockGroup\BlockAdditions)"

id: "0x75A1"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Contain additional blocks to complete the main one. An EBML parser that has no knowledge of the Block structure could still see and use/skip these data.

8.2.37. BlockMore Element

name: "BlockMore"

path: "1*(\Segment\Cluster\BlockGroup\BlockAdditions\BlockMore)"

id: "0xA6"

minOccurs: "1"

type: "master"

minver: "1"

documentation: Contain the BlockAdditional and some parameters.

8.2.38. BlockAddID Element

name: "BlockAddID"

path: "1*1(\Segment\Cluster\BlockGroup\BlockAdditions\BlockMore\BlockAddID)"

id: "0xEE"

minOccurs: "1"
maxOccurs: "1"
range: "not 0"
default: "1"
type: "uinteger"
minver: "1"
documentation: An ID to identify the BlockAdditional level.

8.2.39. BlockAdditional Element

name: "BlockAdditional"
path: "1*1(\Segment\Cluster\BlockGroup\BlockAdditions\BlockMore\BlockAdditional)"
id: "0xA5"
minOccurs: "1"
maxOccurs: "1"
type: "binary"
minver: "1"
documentation: Interpreted by the codec as it wishes (using the BlockAddID).

8.2.40. BlockDuration Element

name: "BlockDuration"
path: "0*1(\Segment\Cluster\BlockGroup\BlockDuration)"
id: "0x9B"
maxOccurs: "1"
default: "DefaultDuration"
type: "uinteger"

minver: "1"

documentation: The duration of the Block (based on TimecodeScale). This Element is mandatory when DefaultDuration is set for the track (but can be omitted as other default values). When not written and with no DefaultDuration, the value is assumed to be the difference between the timestamp of this Block and the timestamp of the next Block in "display" order (not coding order). This Element can be useful at the end of a Track (as there is not other Block available), or when there is a break in a track like for subtitle tracks.

8.2.41. ReferencePriority Element

name: "ReferencePriority"

path: "1*1(\Segment\Cluster\BlockGroup\ReferencePriority)"

id: "0xFA"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: This frame is referenced and has the specified cache priority. In cache only a frame of the same or higher priority can replace this frame. A value of 0 means the frame is not referenced.

8.2.42. ReferenceBlock Element

name: "ReferenceBlock"

path: "0*(\Segment\Cluster\BlockGroup\ReferenceBlock)"

id: "0xFB"

type: "integer"

minver: "1"

documentation: Timestamp of another frame used as a reference (ie: B or P frame). The timestamp is relative to the block it's attached to.

8.2.43. ReferenceVirtual Element

name: "ReferenceVirtual"

path: "0*1(\Segment\Cluster\BlockGroup\ReferenceVirtual)"

id: "0xFD"

maxOccurs: "1"

type: "integer"

minver: "0"

maxver: "0"

documentation: The Segment Position of the data that would otherwise be in position of the virtual block.

8.2.44. CodecState Element

name: "CodecState"

path: "0*1(\Segment\Cluster\BlockGroup\CodecState)"

id: "0xA4"

maxOccurs: "1"

type: "binary"

minver: "2"

documentation: The new codec state to use. Data interpretation is private to the codec. This information SHOULD always be referenced by a seek entry.

8.2.45. DiscardPadding Element

name: "DiscardPadding"

path: "0*1(\Segment\Cluster\BlockGroup\DiscardPadding)"

id: "0x75A2"

maxOccurs: "1"

type: "integer"

minver: "4"

documentation: Duration in nanoseconds of the silent data added to the Block (padding at the end of the Block for positive value, at the beginning of the Block for negative value). The duration of DiscardPadding is not calculated in the duration of the TrackEntry and SHOULD be discarded during playback.

8.2.46. Slices Element

name: "Slices"

path: "0*1(\Segment\Cluster\BlockGroup\Slices)"

id: "0x8E"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Contains slices description.

8.2.47. TimeSlice Element

name: "TimeSlice"

path: "0*(\Segment\Cluster\BlockGroup\Slices\TimeSlice)"

id: "0xE8"

type: "master"

minver: "1"

maxver: "1"

documentation: Contains extra time information about the data contained in the Block. Being able to interpret this Element is not REQUIRED for playback.

8.2.48. LaceNumber Element

name: "LaceNumber"

path: "0*1(\Segment\Cluster\BlockGroup\Slices\TimeSlice\LaceNumber)"

id: "0xCC"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

maxver: "1"

documentation: The reverse number of the frame in the lace (0 is the last frame, 1 is the next to last, etc). Being able to interpret this Element is not REQUIRED for playback.

8.2.49. FrameNumber Element

name: "FrameNumber"

path: "0*1(\Segment\Cluster\BlockGroup\Slices\TimeSlice\FrameNumber)"

id: "0xCD"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: The number of the frame to generate from this lace with this delay (allow you to generate many frames from the same Block/Frame).

8.2.50. BlockAdditionID Element

name: "BlockAdditionID"

path:

"0*1(\Segment\Cluster\BlockGroup\Slices\TimeSlice\BlockAdditionID)"

id: "0xCB"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: The ID of the BlockAdditional Element (0 is the main Block).

8.2.51. Delay Element

name: "Delay"

path: "0*1(\Segment\Cluster\BlockGroup\Slices\TimeSlice\Delay)"

id: "0xCE"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: The (scaled) delay to apply to the Element.

8.2.52. SliceDuration Element

name: "SliceDuration"

path:
"0*1(\Segment\Cluster\BlockGroup\Slices\TimeSlice\SliceDuration)"

id: "0xCF"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: The (scaled) duration to apply to the Element.

8.2.53. ReferenceFrame Element

name: "ReferenceFrame"

path: "0*1(\Segment\Cluster\BlockGroup\ReferenceFrame)"

id: "0xC8"

maxOccurs: "1"

type: "master"

minver: "0"

maxver: "0"

documentation: DivX trick track extensions

8.2.54. ReferenceOffset Element

name: "ReferenceOffset"

path:
"1*1(\Segment\Cluster\BlockGroup\ReferenceFrame\ReferenceOffset)"

id: "0xC9"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: DivX trick track extensions

8.2.55. ReferenceTimeCode Element

name: "ReferenceTimeCode"


```
path:
  "1*1(\Segment\Cluster\BlockGroup\ReferenceFrame\ReferenceTimeCode)"

id: "0xCA"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: DivX trick track extensions
```

8.2.56. EncryptedBlock Element

```
name: "EncryptedBlock"

path: "0*(\Segment\Cluster\EncryptedBlock)"

id: "0xAF"

type: "binary"

minver: "0"

maxver: "0"

documentation: Similar to SimpleBlock but the data inside the Block
are Transformed (encrypt and/or signed). (see EncryptedBlock
Structure)
```

8.2.57. Tracks Element

```
name: "Tracks"

path: "0*(\Segment\Tracks)"

id: "0x1654AE6B"

type: "master"

minver: "1"
```


documentation: A Top-Level Element of information with many tracks described.

8.2.58. TrackEntry Element

name: "TrackEntry"

path: "1*(\Segment\Tracks\TrackEntry)"

id: "0xAE"

minOccurs: "1"

type: "master"

minver: "1"

documentation: Describes a track with all Elements.

8.2.59. TrackNumber Element

name: "TrackNumber"

path: "1*1(\Segment\Tracks\TrackEntry\TrackNumber)"

id: "0xD7"

minOccurs: "1"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: The track number as used in the Block Header (using more than 127 tracks is not encouraged, though the design allows an unlimited number).

8.2.60. TrackUID Element

name: "TrackUID"

path: "1*1(\Segment\Tracks\TrackEntry\TrackUID)"

id: "0x73C5"

minOccurs: "1"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: A unique ID to identify the Track. This SHOULD be kept the same when making a direct stream copy of the Track to another file.

8.2.61. TrackType Element

name: "TrackType"

path: "1*1(\Segment\Tracks\TrackEntry\TrackType)"

id: "0x83"

minOccurs: "1"

maxOccurs: "1"

range: "1-254"

type: "uinteger"

minver: "1"

documentation: A set of track types coded on 8 bits.

8.2.62. FlagEnabled Element

name: "FlagEnabled"

path: "1*1(\Segment\Tracks\TrackEntry\FlagEnabled)"

id: "0xB9"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "1"
type: "uinteger"
minver: "2"
documentation: Set if the track is usable. (1 bit)

8.2.63. FlagDefault Element

name: "FlagDefault"
path: "1*1(\Segment\Tracks\TrackEntry\FlagDefault)"
id: "0x88"
minOccurs: "1"
maxOccurs: "1"
range: "0-1"
default: "1"
type: "uinteger"
minver: "1"
documentation: Set if that track (audio, video or subs) SHOULD be active if no language found matches the user preference. (1 bit)

8.2.64. FlagForced Element

name: "FlagForced"
path: "1*1(\Segment\Tracks\TrackEntry\FlagForced)"
id: "0x55AA"
minOccurs: "1"
maxOccurs: "1"
range: "0-1"
default: "0"
type: "uinteger"

minver: "1"

documentation: Set if that track MUST be active during playback. There can be many forced track for a kind (audio, video or subs), the player SHOULD select the one which language matches the user preference or the default + forced track. Overlay MAY happen between a forced and non-forced track of the same kind. (1 bit)

8.2.65. FlagLacing Element

name: "FlagLacing"

path: "1*1(\Segment\Tracks\TrackEntry\FlagLacing)"

id: "0x9C"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "1"

type: "uinteger"

minver: "1"

documentation: Set if the track MAY contain blocks using lacing. (1 bit)

8.2.66. MinCache Element

name: "MinCache"

path: "1*1(\Segment\Tracks\TrackEntry\MinCache)"

id: "0x6DE7"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The minimum number of frames a player SHOULD be able to cache during playback. If set to 0, the reference pseudo-cache system is not used.

8.2.67. MaxCache Element

name: "MaxCache"

path: "0*1(\Segment\Tracks\TrackEntry\MaxCache)"

id: "0x6DF8"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: The maximum cache size necessary to store referenced frames in and the current frame. 0 means no cache is needed.

8.2.68. DefaultDuration Element

name: "DefaultDuration"

path: "0*1(\Segment\Tracks\TrackEntry\DefaultDuration)"

id: "0x23E383"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: Number of nanoseconds (not scaled via TimecodeScale) per frame ('frame' in the Matroska sense -- one Element put into a (Simple)Block).

8.2.69. DefaultDecodedFieldDuration Element

name: "DefaultDecodedFieldDuration"

path: "0*1(\Segment\Tracks\TrackEntry\DefaultDecodedFieldDuration)"

id: "0x234E7A"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "4"

documentation: The period in nanoseconds (not scaled by TimecodeScale) between two successive fields at the output of the decoding process (see the notes)

8.2.70. TrackTimecodeScale Element

name: "TrackTimecodeScale"

path: "1*1(\Segment\Tracks\TrackEntry\TrackTimecodeScale)"

id: "0x23314F"

minOccurs: "1"

maxOccurs: "1"

range: "> 0x0p+0"

default: "0x1p+0"

type: "float"

minver: "1"

maxver: "3"

documentation: DEPRECATED, DO NOT USE. The scale to apply on this track to work at normal speed in relation with other tracks (mostly used to adjust video speed when the audio length differs).

8.2.71. TrackOffset Element

name: "TrackOffset"

path: "0*1(\Segment\Tracks\TrackEntry\TrackOffset)"

id: "0x537F"

maxOccurs: "1"

default: "0"

type: "integer"

minver: "0"

maxver: "0"

documentation: A value to add to the Block's Timestamp. This can be used to adjust the playback offset of a track.

8.2.72. MaxBlockAdditionID Element

name: "MaxBlockAdditionID"

path: "1*1(\Segment\Tracks\TrackEntry\MaxBlockAdditionID)"

id: "0x55EE"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The maximum value of BlockAddID. A value 0 means there is no BlockAdditions for this track.

8.2.73. Name Element

name: "Name"

path: "0*1(\Segment\Tracks\TrackEntry\Name)"

id: "0x536E"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: A human-readable track name.

8.2.74. Language Element

name: "Language"

path: "0*1(\Segment\Tracks\TrackEntry\Language)"

id: "0x22B59C"

maxOccurs: "1"

default: "eng"

type: "string"

minver: "1"

documentation: Specifies the language of the track in the Matroska languages form. This Element MUST be ignored if the LanguageIETF Element is used in the same TrackEntry.

8.2.75. LanguageIETF Element

name: "LanguageIETF"

path: "0*1(\Segment\Tracks\TrackEntry\LanguageIETF)"

id: "0x22B59D"

maxOccurs: "1"

type: "string"

minver: "4"

documentation: Specifies the language of the track according to [BCP 47](#) and using the IANA Language Subtag Registry. If this Element is used, then any Language Elements used in the same TrackEntry MUST be ignored.

8.2.76. CodecID Element

name: "CodecID"

path: "1*1(\Segment\Tracks\TrackEntry\CodecID)"

id: "0x86"

minOccurs: "1"

maxOccurs: "1"

type: "string"

minver: "1"

documentation: An ID corresponding to the codec, see the codec page for more info.

8.2.77. CodecPrivate Element

name: "CodecPrivate"

path: "0*1(\\Segment\\Tracks\\TrackEntry\\CodecPrivate)"

id: "0x63A2"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: Private data only known to the codec.

8.2.78. CodecName Element

name: "CodecName"

path: "0*1(\\Segment\\Tracks\\TrackEntry\\CodecName)"

id: "0x258688"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: A human-readable string specifying the codec.

8.2.79. AttachmentLink Element

name: "AttachmentLink"

path: "0*1(\\Segment\\Tracks\\TrackEntry\\AttachmentLink)"

id: "0x7446"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

maxver: "3"

documentation: The UID of an attachment that is used by this codec.

8.2.80. CodecSettings Element

name: "CodecSettings"

path: "0*1(\\Segment\\Tracks\\TrackEntry\\CodecSettings)"

id: "0x3A9697"

maxOccurs: "1"

type: "utf-8"

minver: "0"

maxver: "0"

documentation: A string describing the encoding setting used.

8.2.81. CodecInfoURL Element

name: "CodecInfoURL"

path: "0*(\\Segment\\Tracks\\TrackEntry\\CodecInfoURL)"

id: "0x3B4040"

type: "string"

minver: "0"

maxver: "0"

documentation: A URL to find information about the codec used.

[8.2.82.](#) **CodecDownloadURL Element**

name: "CodecDownloadURL"

path: "0*(\Segment\Tracks\TrackEntry\CodecDownloadURL)"

id: "0x26B240"

type: "string"

minver: "0"

maxver: "0"

documentation: A URL to download about the codec used.

[8.2.83.](#) **CodecDecodeAll Element**

name: "CodecDecodeAll"

path: "1*1(\Segment\Tracks\TrackEntry\CodecDecodeAll)"

id: "0xAA"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "1"

type: "uinteger"

minver: "2"

documentation: The codec can decode potentially damaged data (1 bit).

[8.2.84.](#) **TrackOverlay Element**

name: "TrackOverlay"

path: "0*(\Segment\Tracks\TrackEntry\TrackOverlay)"

id: "0x6FAB"

type: "uinteger"

minver: "1"

documentation: Specify that this track is an overlay track for the Track specified (in the u-integer). That means when this track has a gap (see SilentTracks) the overlay track SHOULD be used instead. The order of multiple TrackOverlay matters, the first one is the one that SHOULD be used. If not found it SHOULD be the second, etc.

8.2.85. CodecDelay Element

name: "CodecDelay"

path: "0*1(\Segment\Tracks\TrackEntry\CodecDelay)"

id: "0x56AA"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "4"

documentation: CodecDelay is The codec-built-in delay in nanoseconds. This value MUST be subtracted from each block timestamp in order to get the actual timestamp. The value SHOULD be small so the muxing of tracks with the same actual timestamp are in the same Cluster.

8.2.86. SeekPreRoll Element

name: "SeekPreRoll"

path: "1*1(\Segment\Tracks\TrackEntry\SeekPreRoll)"

id: "0x56BB"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "4"

documentation: After a discontinuity, SeekPreRoll is the duration in nanoseconds of the data the decoder MUST decode before the decoded data is valid.

8.2.87. TrackTranslate Element

name: "TrackTranslate"

path: "0*(\Segment\Tracks\TrackEntry\TrackTranslate)"

id: "0x6624"

type: "master"

minver: "1"

documentation: The track identification for the given Chapter Codec.

8.2.88. TrackTranslateEditionUID Element

name: "TrackTranslateEditionUID"

path: "0*(\Segment\Tracks\TrackEntry\TrackTranslate\TrackTranslateEditionUID)"

id: "0x66FC"

type: "uinteger"

minver: "1"

documentation: Specify an edition UID on which this translation applies. When not specified, it means for all editions found in the Segment.

8.2.89. TrackTranslateCodec Element

name: "TrackTranslateCodec"

path:
"1*1(\Segment\Tracks\TrackEntry\TrackTranslate\TrackTranslateCodec)"

id: "0x66BF"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: The chapter codec.

8.2.90. TrackTranslateTrackID Element

name: "TrackTranslateTrackID"

path: "1*1(\Segment\Tracks\TrackEntry\TrackTranslate\TrackTranslateTrackID)"

id: "0x66A5"

minOccurs: "1"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: The binary value used to represent this track in the chapter codec data. The format depends on the ChapProcessCodecID used.

8.2.91. Video Element

name: "Video"

path: "0*1(\Segment\Tracks\TrackEntry\Video)"

id: "0xE0"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Video settings.

8.2.92. FlagInterlaced Element

name: "FlagInterlaced"

path: "1*1(\Segment\Tracks\TrackEntry\Video\FlagInterlaced)"

id: "0x9A"

minOccurs: "1"

maxOccurs: "1"

range: "0-2"

default: "0"

type: "uinteger"

minver: "2"

documentation: A flag to declare is the video is known to be progressive or interlaced and if applicable to declare details about the interlacement.

8.2.93. FieldOrder Element

name: "FieldOrder"

path: "1*1(\Segment\Tracks\TrackEntry\Video\FieldOrder)"

id: "0x9D"

minOccurs: "1"

maxOccurs: "1"

range: "0-14"

default: "2"

type: "uinteger"

minver: "4"

documentation: Declare the field ordering of the video. If FlagInterlaced is not set to 1, this Element MUST be ignored.

8.2.94. StereoMode Element

name: "StereoMode"

path: "0*1(\Segment\Tracks\TrackEntry\Video\StereoMode)"

id: "0x53B8"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "3"

documentation: Stereo-3D video mode. There are some more details on 3D support in the Specification Notes.

8.2.95. AlphaMode Element

name: "AlphaMode"

path: "0*1(\Segment\Tracks\TrackEntry\Video\AlphaMode)"

id: "0x53C0"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "3"

documentation: Alpha Video Mode. Presence of this Element indicates that the BlockAdditional Element could contain Alpha data.

8.2.96. OldStereoMode Element

name: "OldStereoMode"

path: "0*1(\Segment\Tracks\TrackEntry\Video\OldStereoMode)"

id: "0x53B9"

maxOccurs: "1"

type: "uinteger"

maxver: "0"

documentation: DEPRECATED, DO NOT USE. Bogus StereoMode value used in old versions of libmatroska.

8.2.97. PixelWidth Element

name: "PixelWidth"
path: "1*1(\Segment\Tracks\TrackEntry\Video\PixelWidth)"
id: "0xB0"
minOccurs: "1"
maxOccurs: "1"
range: "not 0"
type: "uinteger"
minver: "1"
documentation: Width of the encoded video frames in pixels.

8.2.98. PixelHeight Element

name: "PixelHeight"
path: "1*1(\Segment\Tracks\TrackEntry\Video\PixelHeight)"
id: "0xBA"
minOccurs: "1"
maxOccurs: "1"
range: "not 0"
type: "uinteger"
minver: "1"
documentation: Height of the encoded video frames in pixels.

8.2.99. PixelCropBottom Element

name: "PixelCropBottom"
path: "0*1(\Segment\Tracks\TrackEntry\Video\PixelCropBottom)"
id: "0x54AA"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The number of video pixels to remove at the bottom of the image (for HDTV content).

8.2.100. PixelCropTop Element

name: "PixelCropTop"

path: "0*1(\Segment\Tracks\TrackEntry\Video\PixelCropTop)"

id: "0x54BB"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The number of video pixels to remove at the top of the image.

8.2.101. PixelCropLeft Element

name: "PixelCropLeft"

path: "0*1(\Segment\Tracks\TrackEntry\Video\PixelCropLeft)"

id: "0x54CC"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The number of video pixels to remove on the left of the image.

8.2.102. PixelCropRight Element

name: "PixelCropRight"

path: "0*1(\Segment\Tracks\TrackEntry\Video\PixelCropRight)"

id: "0x54DD"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The number of video pixels to remove on the right of the image.

8.2.103. DisplayWidth Element

name: "DisplayWidth"

path: "0*1(\Segment\Tracks\TrackEntry\Video\DisplayWidth)"

id: "0x54B0"

maxOccurs: "1"

range: "not 0"

default: "PixelWidth - PixelCropLeft - PixelCropRight"

type: "uinteger"

minver: "1"

documentation: Width of the video frames to display. Applies to the video frame after cropping (PixelCrop* Elements). The default value is only valid when DisplayUnit is 0.

8.2.104. DisplayHeight Element

name: "DisplayHeight"

path: "0*1(\Segment\Tracks\TrackEntry\Video\DisplayHeight)"

id: "0x54BA"

maxOccurs: "1"

range: "not 0"

default: "PixelHeight - PixelCropTop - PixelCropBottom"

type: "uinteger"

minver: "1"

documentation: Height of the video frames to display. Applies to the video frame after cropping (PixelCrop* Elements). The default value is only valid when DisplayUnit is 0.

8.2.105. DisplayUnit Element

name: "DisplayUnit"

path: "0*1(\Segment\Tracks\TrackEntry\Video\DisplayUnit)"

id: "0x54B2"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: How DisplayWidth & DisplayHeight are interpreted.

8.2.106. AspectRatioType Element

name: "AspectRatioType"

path: "0*1(\Segment\Tracks\TrackEntry\Video\AspectRatioType)"

id: "0x54B3"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: Specify the possible modifications to the aspect ratio.

8.2.107. ColourSpace Element

name: "ColourSpace"

path: "0*1(\Segment\Tracks\TrackEntry\Video\ColourSpace)"

id: "0x2EB524"

maxOccurs: "1"

size: "4"

type: "binary"

minver: "1"

documentation: Specify the pixel format used for the Track's data as a FourCC. This value is similar in scope to the biCompression value of AVI's BITMAPINFOHEADER. This Element is MANDATORY in TrackEntry when the CodecID Element of the TrackEntry is set to "V_UNCOMPRESSED".

8.2.108. GammaValue Element

name: "GammaValue"

path: "0*1(\Segment\Tracks\TrackEntry\Video\GammaValue)"

id: "0x2FB523"

maxOccurs: "1"

range: "> 0x0p+0"

type: "float"

minver: "0"

maxver: "0"

documentation: Gamma Value.

8.2.109. FrameRate Element

name: "FrameRate"

path: "0*1(\Segment\Tracks\TrackEntry\Video\FrameRate)"

id: "0x2383E3"

maxOccurs: "1"

range: "> 0x0p+0"

type: "float"

minver: "0"

maxver: "0"

documentation: Number of frames per second. Informational only.

8.2.110. Colour Element

name: "Colour"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour)"

id: "0x55B0"

maxOccurs: "1"

type: "master"

minver: "4"

documentation: Settings describing the colour format.

8.2.111. MatrixCoefficients Element

name: "MatrixCoefficients"

path:
"0*1(\Segment\Tracks\TrackEntry\Video\Colour\MatrixCoefficients)"

id: "0x55B1"

maxOccurs: "1"

default: "2"

type: "uinteger"

minver: "4"

documentation: The Matrix Coefficients of the video used to derive luma and chroma values from red, green, and blue color primaries. For clarity, the value and meanings for MatrixCoefficients are adopted from Table 4 of ISO/IEC 23001-8:2013/DCOR1.

8.2.112. BitsPerChannel Element

name: "BitsPerChannel"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\BitsPerChannel)"

id: "0x55B2"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "4"

documentation: Number of decoded bits per channel. A value of 0 indicates that the BitsPerChannel is unspecified.

8.2.113. ChromaSubsamplingHorz Element

name: "ChromaSubsamplingHorz"

path:
"0*1(\Segment\Tracks\TrackEntry\Video\Colour\ChromaSubsamplingHorz)"

id: "0x55B3"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: The amount of pixels to remove in the Cr and Cb channels for every pixel not removed horizontally. Example: For video with 4:2:0 chroma subsampling, the ChromaSubsamplingHorz SHOULD be set to 1.

8.2.114. ChromaSubsamplingVert Element

name: "ChromaSubsamplingVert"

path:
"0*1(\Segment\Tracks\TrackEntry\Video\Colour\ChromaSubsamplingVert)"

id: "0x55B4"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: The amount of pixels to remove in the Cr and Cb channels for every pixel not removed vertically. Example: For video with 4:2:0 chroma subsampling, the ChromaSubsamplingVert SHOULD be set to 1.

8.2.115. CbSubsamplingHorz Element

name: "CbSubsamplingHorz"

path:
"0*1(\Segment\Tracks\TrackEntry\Video\Colour\CbSubsamplingHorz)"

id: "0x55B5"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: The amount of pixels to remove in the Cb channel for every pixel not removed horizontally. This is additive with ChromaSubsamplingHorz. Example: For video with 4:2:1 chroma subsampling, the ChromaSubsamplingHorz SHOULD be set to 1 and CbSubsamplingHorz SHOULD be set to 1.

8.2.116. CbSubsamplingVert Element

name: "CbSubsamplingVert"

path:
"0*1(\Segment\Tracks\TrackEntry\Video\Colour\CbSubsamplingVert)"

id: "0x55B6"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: The amount of pixels to remove in the Cb channel for every pixel not removed vertically. This is additive with ChromaSubsamplingVert.

8.2.117. ChromaSitingHorz Element

name: "ChromaSitingHorz"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\ChromaSitingHorz)"

id: "0x55B7"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "4"

documentation: How chroma is subsampled horizontally.

8.2.118. ChromaSitingVert Element

name: "ChromaSitingVert"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\ChromaSitingVert)"

id: "0x55B8"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "4"

documentation: How chroma is subsampled vertically.

8.2.119. Range Element

name: "Range"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\Range)"

id: "0x55B9"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "4"

documentation: Clipping of the color ranges.

8.2.120. TransferCharacteristics Element

name: "TransferCharacteristics"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\TransferCharacteristics)"

id: "0x55BA"

maxOccurs: "1"

default: "2"

type: "uinteger"

minver: "4"

documentation: The transfer characteristics of the video. For clarity, the value and meanings for TransferCharacteristics 1-15 are adopted from Table 3 of ISO/IEC 23001-8:2013/DCOR1. TransferCharacteristics 16-18 are proposed values.

8.2.121. Primaries Element

name: "Primaries"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\Primaries)"

id: "0x55BB"

maxOccurs: "1"

default: "2"

type: "uinteger"

minver: "4"

documentation: The colour primaries of the video. For clarity, the value and meanings for Primaries are adopted from Table 2 of ISO/IEC 23001-8:2013/DCOR1.

8.2.122. MaxCLL Element

name: "MaxCLL"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MaxCLL)"

id: "0x55BC"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: Maximum brightness of a single pixel (Maximum Content Light Level) in candelas per square meter (cd/m²).

8.2.123. MaxFALL Element

name: "MaxFALL"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MaxFALL)"

id: "0x55BD"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: Maximum brightness of a single full frame (Maximum Frame-Average Light Level) in candelas per square meter (cd/m²).

8.2.124. MasteringMetadata Element

name: "MasteringMetadata"

path:

"0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata)"

id: "0x55D0"

maxOccurs: "1"

type: "master"

minver: "4"

documentation: SMPTE 2086 mastering data.

8.2.125. PrimaryRChromaticityX Element

name: "PrimaryRChromaticityX"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\
PrimaryRChromaticityX)"

id: "0x55D1"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: Red X chromaticity coordinate as defined by CIE 1931.

8.2.126. PrimaryRChromaticityY Element

name: "PrimaryRChromaticityY"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\
PrimaryRChromaticityY)"

id: "0x55D2"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: Red Y chromaticity coordinate as defined by CIE 1931.

8.2.127. PrimaryGChromaticityX Element

name: "PrimaryGChromaticityX"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\
PrimaryGChromaticityX)"

id: "0x55D3"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: Green X chromaticity coordinate as defined by CIE
1931.

8.2.128. PrimaryGChromaticityY Element

name: "PrimaryGChromaticityY"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\
PrimaryGChromaticityY)"

id: "0x55D4"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: Green Y chromaticity coordinate as defined by CIE
1931.

8.2.129. PrimaryBChromaticityX Element

name: "PrimaryBChromaticityX"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\
PrimaryBChromaticityX)"

id: "0x55D5"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: Blue X chromaticity coordinate as defined by CIE 1931.

8.2.130. PrimaryBChromaticityY Element

name: "PrimaryBChromaticityY"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\
PrimaryBChromaticityY)"

id: "0x55D6"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: Blue Y chromaticity coordinate as defined by CIE 1931.

8.2.131. WhitePointChromaticityX Element

name: "WhitePointChromaticityX"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\
WhitePointChromaticityX)"

id: "0x55D7"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: White X chromaticity coordinate as defined by CIE 1931.

8.2.132. WhitePointChromaticityY Element

name: "WhitePointChromaticityY"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\WhitePointChromaticityY)"

id: "0x55D8"

maxOccurs: "1"

range: "0-1"

type: "float"

minver: "4"

documentation: White Y chromaticity coordinate as defined by CIE 1931.

8.2.133. LuminanceMax Element

name: "LuminanceMax"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\LuminanceMax)"

id: "0x55D9"

maxOccurs: "1"

range: ">= 0x0p+0"

type: "float"

minver: "4"

documentation: Maximum luminance. Represented in candelas per square meter (cd/m²).

8.2.134. LuminanceMin Element

name: "LuminanceMin"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\LuminanceMin)"

id: "0x55DA"

maxOccurs: "1"

range: ">= 0x0p+0"

type: "float"

minver: "4"

documentation: Minimum luminance. Represented in candelas per square meter (cd/m²).

8.2.135. Projection Element

name: "Projection"

path: "0*1(\Segment\Tracks\TrackEntry\Video\Projection)"

id: "0x7670"

maxOccurs: "1"

type: "master"

minver: "4"

documentation: Describes the video projection details. Used to render spherical and VR videos.

8.2.136. ProjectionType Element

name: "ProjectionType"

path:
"1*1(\Segment\Tracks\TrackEntry\Video\Projection\ProjectionType)"

id: "0x7671"

minOccurs: "1"

maxOccurs: "1"

range: "0-3"

default: "0"

type: "uinteger"

minver: "4"

documentation: Describes the projection used for this video track.

8.2.137. ProjectionPrivate Element

name: "ProjectionPrivate"

path:

"0*1(\Segment\Tracks\TrackEntry\Video\Projection\ProjectionPrivate)"

id: "0x7672"

maxOccurs: "1"

type: "binary"

minver: "4"

documentation: Private data that only applies to a specific projection. Semantics: If ProjectionType equals 0 (Rectangular), then this element must not be present. If ProjectionType equals 1 (Equirectangular), then this element must be present and contain the same binary data that would be stored inside an ISO BMFF Equirectangular Projection Box ('equi'). If ProjectionType equals 2 (Cubemap), then this element must be present and contain the same binary data that would be stored inside an ISO BMFF Cubemap Projection Box ('cbmp'). If ProjectionType equals 3 (Mesh), then this element must be present and contain the same binary data that would be stored inside an ISO BMFF Mesh Projection Box ('mshp'). Note: ISO BMFF box size and fourcc fields are not included in the binary data, but the FullBox version and flag fields are. This is to avoid redundant framing information while preserving versioning and semantics between the two container formats.

8.2.138. ProjectionPoseYaw Element

name: "ProjectionPoseYaw"

path:
"1*1(\Segment\Tracks\TrackEntry\Video\Projection\ProjectionPoseYaw)"

id: "0x7673"

minOccurs: "1"

maxOccurs: "1"

default: "0x0p+0"

type: "float"

minver: "4"

documentation: Specifies a yaw rotation to the
projection.SemanticsValue represents a clockwise rotation, in
degrees, around the up vector. This rotation must be applied before
any ProjectionPosePitch or ProjectionPoseRoll rotations. The value
of this field should be in the -180 to 180 degree range.

8.2.139. ProjectionPosePitch Element

name: "ProjectionPosePitch"

path: "1*1(\Segment\Tracks\TrackEntry\Video\Projection\ProjectionPose
Pitch)"

id: "0x7674"

minOccurs: "1"

maxOccurs: "1"

default: "0x0p+0"

type: "float"

minver: "4"

documentation: Specifies a pitch rotation to the
projection.SemanticsValue represents a counter-clockwise rotation, in
degrees, around the right vector. This rotation must be applied
after the ProjectionPoseYaw rotation and before the
ProjectionPoseRoll rotation. The value of this field should be in
the -90 to 90 degree range.

8.2.140. ProjectionPoseRoll Element

name: "ProjectionPoseRoll"

path:
"1*1(\Segment\Tracks\TrackEntry\Video\Projection\ProjectionPoseRoll)"

id: "0x7675"

minOccurs: "1"

maxOccurs: "1"

default: "0x0p+0"

type: "float"

minver: "4"

documentation: Specifies a roll rotation to the projection.SemanticsValue represents a counter-clockwise rotation, in degrees, around the forward vector. This rotation must be applied after the ProjectionPoseYaw and ProjectionPosePitch rotations. The value of this field should be in the -180 to 180 degree range.

8.2.141. Audio Element

name: "Audio"

path: "0*1(\Segment\Tracks\TrackEntry\Audio)"

id: "0xE1"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Audio settings.

8.2.142. SamplingFrequency Element

name: "SamplingFrequency"

path: "1*1(\Segment\Tracks\TrackEntry\Audio\SamplingFrequency)"

id: "0xB5"

minOccurs: "1"
maxOccurs: "1"
range: "> 0x0p+0"
default: "0x1.f4p+12"
type: "float"
minver: "1"
documentation: Sampling frequency in Hz.

8.2.143. OutputSamplingFrequency Element

name: "OutputSamplingFrequency"
path: "0*1(\Segment\Tracks\TrackEntry\Audio\OutputSamplingFrequency)"
id: "0x78B5"
maxOccurs: "1"
range: "> 0x0p+0"
default: "SamplingFrequency"
type: "float"
minver: "1"
documentation: Real output sampling frequency in Hz (used for SBR techniques).

8.2.144. Channels Element

name: "Channels"
path: "1*1(\Segment\Tracks\TrackEntry\Audio\Channels)"
id: "0x9F"
minOccurs: "1"
maxOccurs: "1"
range: "not 0"

default: "1"
type: "uinteger"
minver: "1"
documentation: Numbers of channels in the track.

8.2.145. ChannelPositions Element

name: "ChannelPositions"
path: "0*1(\Segment\Tracks\TrackEntry\Audio\ChannelPositions)"
id: "0x7D7B"
maxOccurs: "1"
type: "binary"
minver: "0"
maxver: "0"
documentation: Table of horizontal angles for each successive channel, see appendix.

8.2.146. BitDepth Element

name: "BitDepth"
path: "0*1(\Segment\Tracks\TrackEntry\Audio\BitDepth)"
id: "0x6264"
maxOccurs: "1"
range: "not 0"
type: "uinteger"
minver: "1"
documentation: Bits per sample, mostly used for PCM.

8.2.147. TrackOperation Element

name: "TrackOperation"

path: "0*1(\Segment\Tracks\TrackEntry\TrackOperation)"

id: "0xE2"

maxOccurs: "1"

type: "master"

minver: "3"

documentation: Operation that needs to be applied on tracks to create this virtual track. For more details look at the Specification Notes on the subject.

8.2.148. TrackCombinePlanes Element

name: "TrackCombinePlanes"

path:
"0*1(\Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes)"

id: "0xE3"

maxOccurs: "1"

type: "master"

minver: "3"

documentation: Contains the list of all video plane tracks that need to be combined to create this 3D track

8.2.149. TrackPlane Element

name: "TrackPlane"

path: "1*(\Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes\TrackPlane)"

id: "0xE4"

minOccurs: "1"

type: "master"

minver: "3"

documentation: Contains a video plane track that need to be combined to create this 3D track

8.2.150. TrackPlaneUID Element

name: "TrackPlaneUID"

path: "1*1(\Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes\TrackPlane\TrackPlaneUID)"

id: "0xE5"

minOccurs: "1"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "3"

documentation: The trackUID number of the track representing the plane.

8.2.151. TrackPlaneType Element

name: "TrackPlaneType"

path: "1*1(\Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes\TrackPlane\TrackPlaneType)"

id: "0xE6"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "3"

documentation: The kind of plane this track corresponds to.

[8.2.152.](#) TrackJoinBlocks Element

name: "TrackJoinBlocks"

path:

"0*1(\Segment\Tracks\TrackEntry\TrackOperation\TrackJoinBlocks)"

id: "0xE9"

maxOccurs: "1"

type: "master"

minver: "3"

documentation: Contains the list of all tracks whose Blocks need to be combined to create this virtual track

[8.2.153.](#) TrackJoinUID Element

name: "TrackJoinUID"

path: "1*(\Segment\Tracks\TrackEntry\TrackOperation\TrackJoinBlocks\TrackJoinUID)"

id: "0xED"

minOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "3"

documentation: The trackUID number of a track whose blocks are used to create this virtual track.

[8.2.154.](#) TrickTrackUID Element

name: "TrickTrackUID"

path: "0*1(\Segment\Tracks\TrackEntry\TrickTrackUID)"

id: "0xC0"

maxOccurs: "1"

type: "uinteger"
minver: "0"
maxver: "0"
documentation: DivX trick track extensions

8.2.155. TrickTrackSegmentUID Element

name: "TrickTrackSegmentUID"
path: "0*1(\Segment\Tracks\TrackEntry\TrickTrackSegmentUID)"
id: "0xC1"
maxOccurs: "1"
size: "16"
type: "binary"
minver: "0"
maxver: "0"
documentation: DivX trick track extensions

8.2.156. TrickTrackFlag Element

name: "TrickTrackFlag"
path: "0*1(\Segment\Tracks\TrackEntry\TrickTrackFlag)"
id: "0xC6"
maxOccurs: "1"
default: "0"
type: "uinteger"
minver: "0"
maxver: "0"
documentation: DivX trick track extensions

8.2.157. TrickMasterTrackUID Element

name: "TrickMasterTrackUID"
path: "0*1(\Segment\Tracks\TrackEntry\TrickMasterTrackUID)"
id: "0xC7"
maxOccurs: "1"
type: "uinteger"
minver: "0"
maxver: "0"
documentation: DivX trick track extensions

8.2.158. TrickMasterTrackSegmentUID Element

name: "TrickMasterTrackSegmentUID"
path: "0*1(\Segment\Tracks\TrackEntry\TrickMasterTrackSegmentUID)"
id: "0xC4"
maxOccurs: "1"
size: "16"
type: "binary"
minver: "0"
maxver: "0"
documentation: DivX trick track extensions

8.2.159. ContentEncodings Element

name: "ContentEncodings"
path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings)"
id: "0x6D80"
maxOccurs: "1"

type: "master"

minver: "1"

documentation: Settings for several content encoding mechanisms like compression or encryption.

8.2.160. ContentEncoding Element

name: "ContentEncoding"

path:

"1*(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding)"

id: "0x6240"

minOccurs: "1"

type: "master"

minver: "1"

documentation: Settings for one content encoding like compression or encryption.

8.2.161. ContentEncodingOrder Element

name: "ContentEncodingOrder"

path: "1*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncodingOrder)"

id: "0x5031"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: Tells when this modification was used during encoding/muxing starting with 0 and counting upwards. The decoder/demuxer has to start with the highest order number it finds and work its way

down. This value has to be unique over all ContentEncodingOrder Elements in the Segment.

8.2.162. ContentEncodingScope Element

name: "ContentEncodingScope"

path: "1*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncodingScope)"

id: "0x5032"

minOccurs: "1"

maxOccurs: "1"

range: "not 0"

default: "1"

type: "uinteger"

minver: "1"

documentation: A bit field that describes which Elements have been modified in this way. Values (big endian) can be OR'ed. Possible values: 1 - all frame contents, 2 - the track's private data, 4 - the next ContentEncoding (next ContentEncodingOrder. Either the data inside ContentCompression and/or ContentEncryption)

8.2.163. ContentEncodingType Element

name: "ContentEncodingType"

path: "1*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncodingType)"

id: "0x5033"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: A value describing what kind of transformation has been done. Possible values: 0 - compression, 1 - encryption

8.2.164. ContentCompression Element

name: "ContentCompression"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentCompression)"

id: "0x5034"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Settings describing the compression used. This Element MUST be present if the value of ContentEncodingType is 0 and absent otherwise. Each block MUST be decompressable even if no previous block is available in order not to prevent seeking.

8.2.165. ContentCompAlgo Element

name: "ContentCompAlgo"

path: "1*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentCompression\ContentCompAlgo)"

id: "0x4254"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The compression algorithm used. Algorithms that have been specified so far are: 0 - zlib, 1 - bzip, 2 - lzo1x 3 - Header Stripping

8.2.166. ContentCompSettings Element

name: "ContentCompSettings"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentCompression\ContentCompSettings)"

id: "0x4255"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: Settings that might be needed by the decompressor. For Header Stripping (ContentCompAlgo=3), the bytes that were removed from the beginning of each frames of the track.

8.2.167. ContentEncryption Element

name: "ContentEncryption"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption)"

id: "0x5035"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Settings describing the encryption used. This Element MUST be present if the value of ContentEncodingType is 1 and absent otherwise.

8.2.168. ContentEncAlgo Element

name: "ContentEncAlgo"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentEncAlgo)"

id: "0x47E1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The encryption algorithm used. The value '0' means that the contents have not been encrypted but only signed.

Predefined values: 1 - DES, 2 - 3DES, 3 - Twofish, 4 - Blowfish, 5 - AES

8.2.169. ContentEncKeyID Element

name: "ContentEncKeyID"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentEncKeyID)"

id: "0x47E2"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: For public key algorithms this is the ID of the public key the the data was encrypted with.

8.2.170. ContentSignature Element

name: "ContentSignature"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentSignature)"

id: "0x47E3"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: A cryptographic signature of the contents.

8.2.171. ContentSigKeyID Element

name: "ContentSigKeyID"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentSigKeyID)"

id: "0x47E4"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: This is the ID of the private key the data was signed with.

8.2.172. ContentSigAlgo Element

name: "ContentSigAlgo"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentSigAlgo)"

id: "0x47E5"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The algorithm used for the signature. A value of '0' means that the contents have not been signed but only encrypted.

Predefined values: 1 - RSA

8.2.173. ContentSigHashAlgo Element

name: "ContentSigHashAlgo"

path: "0*1(\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentSigHashAlgo)"

id: "0x47E6"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: The hash algorithm used for the signature. A value of '0' means that the contents have not been signed but only encrypted.
Predefined values: 1 - SHA1-160 2 - MD5

8.2.174. Cues Element

name: "Cues"

path: "0*1(\Segment\Cues)"

id: "0x1C53BB6B"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: A Top-Level Element to speed seeking access. All entries are local to the Segment. This Element SHOULD be mandatory for non "live" streams.

8.2.175. CuePoint Element

name: "CuePoint"

path: "1*(\Segment\Cues\CuePoint)"

id: "0xBB"

minOccurs: "1"

type: "master"

minver: "1"

documentation: Contains all information relative to a seek point in the Segment.

8.2.176. CueTime Element

name: "CueTime"

path: "1*1(\Segment\Cues\CuePoint\CueTime)"

id: "0xB3"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: Absolute timestamp according to the Segment time base.

8.2.177. CueTrackPositions Element

name: "CueTrackPositions"

path: "1*(\Segment\Cues\CuePoint\CueTrackPositions)"

id: "0xB7"

minOccurs: "1"

type: "master"

minver: "1"

documentation: Contain positions for different tracks corresponding to the timestamp.

8.2.178. CueTrack Element

name: "CueTrack"

path: "1*1(\Segment\Cues\CuePoint\CueTrackPositions\CueTrack)"

id: "0xF7"

minOccurs: "1"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: The track for which a position is given.

8.2.179. CueClusterPosition Element

name: "CueClusterPosition"

path:

"1*1(\Segment\Cues\CuePoint\CueTrackPositions\CueClusterPosition)"

id: "0xF1"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: The Segment Position of the Cluster containing the associated Block.

8.2.180. CueRelativePosition Element

name: "CueRelativePosition"

path:

"0*1(\Segment\Cues\CuePoint\CueTrackPositions\CueRelativePosition)"

id: "0xF0"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: The relative position of the referenced block inside the cluster with 0 being the first possible position for an Element inside that cluster.

8.2.181. CueDuration Element

name: "CueDuration"

path: "0*1(\Segment\Cues\CuePoint\CueTrackPositions\CueDuration)"

id: "0xB2"

maxOccurs: "1"

type: "uinteger"

minver: "4"

documentation: The duration of the block according to the Segment time base. If missing the track's DefaultDuration does not apply and no duration information is available in terms of the cues.

8.2.182. CueBlockNumber Element

name: "CueBlockNumber"

path: "0*1(\Segment\Cues\CuePoint\CueTrackPositions\CueBlockNumber)"

id: "0x5378"

maxOccurs: "1"

range: "not 0"

default: "1"

type: "uinteger"

minver: "1"

documentation: Number of the Block in the specified Cluster.

8.2.183. CueCodecState Element

name: "CueCodecState"

path: "0*1(\Segment\Cues\CuePoint\CueTrackPositions\CueCodecState)"

id: "0xEA"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "2"

documentation: The Segment Position of the Codec State corresponding to this Cue Element. 0 means that the data is taken from the initial Track Entry.

8.2.184. CueReference Element

name: "CueReference"

path: "0*(\Segment\Cues\CuePoint\CueTrackPositions\CueReference)"

id: "0xDB"

type: "master"

minver: "2"

documentation: The Clusters containing the referenced Blocks.

8.2.185. CueRefTime Element

name: "CueRefTime"

path: "1*1(\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefTime)"

id: "0x96"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "2"

documentation: Timestamp of the referenced Block.

8.2.186. CueRefCluster Element

name: "CueRefCluster"

path: "1*1(\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefCluster)"

id: "0x97"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: The Segment Position of the Cluster containing the referenced Block.

8.2.187. CueRefNumber Element

name: "CueRefNumber"

path: "0*1(\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefNumber)"

id: "0x535F"

maxOccurs: "1"

range: "not 0"

default: "1"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: Number of the referenced Block of Track X in the specified Cluster.

8.2.188. CueRefCodecState Element

name: "CueRefCodecState"

path: "0*1(\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefCodecState)"

id: "0xEB"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: The Segment Position of the Codec State corresponding to this referenced Element. 0 means that the data is taken from the initial Track Entry.

8.2.189. Attachments Element

name: "Attachments"

path: "0*1(\Segment\Attachments)"

id: "0x1941A469"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Contain attached files.

8.2.190. AttachedFile Element

name: "AttachedFile"

path: "1*(\Segment\Attachments\AttachedFile)"

id: "0x61A7"

minOccurs: "1"

type: "master"

minver: "1"

documentation: An attached file.

8.2.191. FileDescription Element

name: "FileDescription"

path: "0*1(\Segment\Attachments\AttachedFile\FileDescription)"

id: "0x467E"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: A human-friendly name for the attached file.

8.2.192. FileName Element

name: "FileName"

path: "1*1(\Segment\Attachments\AttachedFile\FileName)"

id: "0x466E"

minOccurs: "1"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: Filename of the attached file.

8.2.193. FileMimeType Element

name: "FileMimeType"

path: "1*1(\Segment\Attachments\AttachedFile\FileMimeType)"

id: "0x4660"

minOccurs: "1"

maxOccurs: "1"

type: "string"

minver: "1"

documentation: MIME type of the file.

8.2.194. FileData Element

name: "FileData"

path: "1*1(\Segment\Attachments\AttachedFile\FileData)"

id: "0x465C"

minOccurs: "1"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: The data of the file.

8.2.195. FileUID Element

name: "FileUID"

path: "1*1(\Segment\Attachments\AttachedFile\FileUID)"

id: "0x46AE"

minOccurs: "1"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: Unique ID representing the file, as random as possible.

8.2.196. FileReferral Element

name: "FileReferral"

path: "0*1(\Segment\Attachments\AttachedFile\FileReferral)"

id: "0x4675"

maxOccurs: "1"

type: "binary"

minver: "0"

maxver: "0"

documentation: A binary value that a track/codec can refer to when the attachment is needed.

8.2.197. FileUsedStartTime Element

name: "FileUsedStartTime"

path: "0*1(\Segment\Attachments\AttachedFile\FileUsedStartTime)"

id: "0x4661"

maxOccurs: "1"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: DivX font extension

8.2.198. FileUsedEndTime Element

name: "FileUsedEndTime"

path: "0*1(\Segment\Attachments\AttachedFile\FileUsedEndTime)"

id: "0x4662"

maxOccurs: "1"

type: "uinteger"

minver: "0"

maxver: "0"

documentation: DivX font extension

8.2.199. Chapters Element

name: "Chapters"

path: "0*1(\Segment\Chapters)"

id: "0x1043A770"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: A system to define basic menus and partition data.
For more detailed information, look at the Chapters Explanation.

8.2.200. EditionEntry Element

name: "EditionEntry"

path: "1*(\Segment\Chapters\EditionEntry)"

id: "0x45B9"

minOccurs: "1"

type: "master"

minver: "1"

documentation: Contains all information about a Segment edition.

8.2.201. EditionUID Element

name: "EditionUID"

path: "0*1(\Segment\Chapters\EditionEntry\EditionUID)"

id: "0x45BC"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: A unique ID to identify the edition. It's useful for tagging an edition.

8.2.202. EditionFlagHidden Element

name: "EditionFlagHidden"

path: "1*1(\Segment\Chapters\EditionEntry\EditionFlagHidden)"

id: "0x45BD"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "0"

type: "uinteger"

minver: "1"

documentation: If an edition is hidden (1), it SHOULD NOT be available to the user interface (but still to Control Tracks; see flag notes). (1 bit)

8.2.203. EditionFlagDefault Element

name: "EditionFlagDefault"

path: "1*1(\Segment\Chapters\EditionEntry\EditionFlagDefault)"

id: "0x45DB"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "0"

type: "uinteger"

minver: "1"

documentation: If a flag is set (1) the edition SHOULD be used as the default one. (1 bit)

8.2.204. EditionFlagOrdered Element

name: "EditionFlagOrdered"

path: "0*1(\Segment\Chapters\EditionEntry\EditionFlagOrdered)"

id: "0x45DD"

maxOccurs: "1"

range: "0-1"

default: "0"

type: "uinteger"

minver: "1"

documentation: Specify if the chapters can be defined multiple times and the order to play them is enforced. (1 bit)

8.2.205. ChapterAtom Element

name: "ChapterAtom"

path: "1*(\Segment\Chapters\EditionEntry(1*(\ChapterAtom)))"

id: "0xB6"

minOccurs: "1"

type: "master"

recursive: "1"

minver: "1"

documentation: Contains the atom information to use as the chapter atom (apply to all tracks).

8.2.206. ChapterUID Element

name: "ChapterUID"

path: "1*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterUID)"

id: "0x73C4"
minOccurs: "1"
maxOccurs: "1"
range: "not 0"
type: "uinteger"
minver: "1"
documentation: A unique ID to identify the Chapter.

8.2.207. ChapterStringUID Element

name: "ChapterStringUID"
path:
"0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterStringUID)"
id: "0x5654"
maxOccurs: "1"
type: "utf-8"
minver: "3"
documentation: A unique string ID to identify the Chapter. Use for WebVTT cue identifier storage.

8.2.208. ChapterTimeStart Element

name: "ChapterTimeStart"
path:
"1*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterTimeStart)"
id: "0x91"
minOccurs: "1"
maxOccurs: "1"
type: "uinteger"
minver: "1"

documentation: Timestamp of the start of Chapter (not scaled).

8.2.209. ChapterTimeEnd Element

name: "ChapterTimeEnd"

path:

"0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterTimeEnd)"

id: "0x92"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: Timestamp of the end of Chapter (timestamp excluded, not scaled).

8.2.210. ChapterFlagHidden Element

name: "ChapterFlagHidden"

path:

"1*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterFlagHidden)"

id: "0x98"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "0"

type: "uinteger"

minver: "1"

documentation: If a chapter is hidden (1), it SHOULD NOT be available to the user interface (but still to Control Tracks; see flag notes).
(1 bit)

8.2.211. ChapterFlagEnabled Element

name: "ChapterFlagEnabled"

path:

"1*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterFlagEnabled)"

id: "0x4598"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "1"

type: "uinteger"

minver: "1"

documentation: Specify whether the chapter is enabled. It can be enabled/disabled by a Control Track. When disabled, the movie SHOULD skip all the content between the TimeStart and TimeEnd of this chapter (see flag notes). (1 bit)

8.2.212. ChapterSegmentUID Element

name: "ChapterSegmentUID"

path:

"0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterSegmentUID)"

id: "0x6E67"

maxOccurs: "1"

range: ">0"

size: "16"

type: "binary"

minver: "1"

documentation: The SegmentUID of another Segment to play during this chapter.

usage notes: ChapterSegmentUID is mandatory if ChapterSegmentEditionUID is used.

8.2.213. ChapterSegmentEditionUID Element

name: "ChapterSegmentEditionUID"

path: "0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterSegmentEditionUID)"

id: "0x6EBC"

maxOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: The EditionUID to play from the Segment linked in ChapterSegmentUID. If ChapterSegmentEditionUID is undeclared then no Edition of the linked Segment is used.

8.2.214. ChapterPhysicalEquiv Element

name: "ChapterPhysicalEquiv"

path: "0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterPhysicalEquiv)"

id: "0x63C3"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: Specify the physical equivalent of this ChapterAtom like "DVD" (60) or "SIDE" (50), see complete list of values.

8.2.215. ChapterTrack Element

name: "ChapterTrack"

path: "0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterTrack)"

id: "0x8F"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: List of tracks on which the chapter applies. If this Element is not present, all tracks apply

8.2.216. ChapterTrackNumber Element

name: "ChapterTrackNumber"

path: "1*(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterTrack\ChapterTrackNumber)"

id: "0x89"

minOccurs: "1"

range: "not 0"

type: "uinteger"

minver: "1"

documentation: UID of the Track to apply this chapter too. In the absence of a control track, choosing this chapter will select the listed Tracks and deselect unlisted tracks. Absence of this Element indicates that the Chapter SHOULD be applied to any currently used Tracks.

8.2.217. ChapterDisplay Element

name: "ChapterDisplay"

path: "0*(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterDisplay)"

id: "0x80"

type: "master"

minver: "1"

documentation: Contains all possible strings to use for the chapter display.

8.2.218. ChapString Element

name: "ChapString"

path: "1*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterDisplay\ChapString)"

id: "0x85"

minOccurs: "1"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: Contains the string to use as the chapter atom.

8.2.219. ChapLanguage Element

name: "ChapLanguage"

path: "1*(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterDisplay\ChapLanguage)"

id: "0x437C"

minOccurs: "1"

default: "eng"

type: "string"

minver: "1"

documentation: The languages corresponding to the string, in the bibliographic ISO-639-2 form. This Element MUST be ignored if the ChapLanguageIETF Element is used within the same ChapterDisplay Element.

8.2.220. ChapLanguageIETF Element

name: "ChapLanguageIETF"

path: "0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterDisplay\ChapLanguageIETF)"

id: "0x437D"

maxOccurs: "1"

type: "string"

minver: "4"

documentation: Specifies the language used in the ChapString according to [BCP 47](#) and using the IANA Language Subtag Registry. If this Element is used, then any ChapLanguage Elements used in the same ChapterDisplay MUST be ignored.

[8.2.221](#). ChapCountry Element

name: "ChapCountry"

path: "0*(\Segment\Chapters\EditionEntry\ChapterAtom\ChapterDisplay\ChapCountry)"

id: "0x437E"

type: "string"

minver: "1"

documentation: The countries corresponding to the string, same 2 octets as in Internet domains. This Element MUST be ignored if the ChapLanguageIETF Element is used within the same ChapterDisplay Element.

[8.2.222](#). ChapProcess Element

name: "ChapProcess"

path: "0*(\Segment\Chapters\EditionEntry\ChapterAtom\ChapProcess)"

id: "0x6944"

type: "master"

minver: "1"

documentation: Contains all the commands associated to the Atom.

8.2.223. ChapProcessCodecID Element

name: "ChapProcessCodecID"

path: "1*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapProcess\ChapProcessCodecID)"

id: "0x6955"

minOccurs: "1"

maxOccurs: "1"

default: "0"

type: "uinteger"

minver: "1"

documentation: Contains the type of the codec used for the processing. A value of 0 means native Matroska processing (to be defined), a value of 1 means the DVD command set is used. More codec IDs can be added later.

8.2.224. ChapProcessPrivate Element

name: "ChapProcessPrivate"

path: "0*1(\Segment\Chapters\EditionEntry\ChapterAtom\ChapProcess\ChapProcessPrivate)"

id: "0x450D"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: Some optional data attached to the ChapProcessCodecID information. For ChapProcessCodecID = 1, it is the "DVD level" equivalent.

8.2.225. ChapProcessCommand Element

name: "ChapProcessCommand"

path: "0*(\\Segment\\Chapters\\EditionEntry\\ChapterAtom\\ChapProcess\\ChapProcessCommand)"

id: "0x6911"

type: "master"

minver: "1"

documentation: Contains all the commands associated to the Atom.

8.2.226. ChapProcessTime Element

name: "ChapProcessTime"

path: "1*1(\\Segment\\Chapters\\EditionEntry\\ChapterAtom\\ChapProcess\\ChapProcessCommand\\ChapProcessTime)"

id: "0x6922"

minOccurs: "1"

maxOccurs: "1"

type: "uinteger"

minver: "1"

documentation: Defines when the process command SHOULD be handled

8.2.227. ChapProcessData Element

name: "ChapProcessData"

path: "1*1(\\Segment\\Chapters\\EditionEntry\\ChapterAtom\\ChapProcess\\ChapProcessCommand\\ChapProcessData)"

id: "0x6933"

minOccurs: "1"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: Contains the command information. The data SHOULD be interpreted depending on the ChapProcessCodecID value. For ChapProcessCodecID = 1, the data correspond to the binary DVD cell pre/post commands.

8.2.228. Tags Element

name: "Tags"

path: "0*(\Segment\Tags)"

id: "0x1254C367"

type: "master"

minver: "1"

documentation: Element containing metadata describing Tracks, Editions, Chapters, Attachments, or the Segment as a whole. A list of valid tags can be found here.

8.2.229. Tag Element

name: "Tag"

path: "1*(\Segment\Tags\Tag)"

id: "0x7373"

minOccurs: "1"

type: "master"

minver: "1"

documentation: A single metadata descriptor.

8.2.230. Targets Element

name: "Targets"

path: "1*1(\Segment\Tags\Tag\Targets)"

id: "0x63C0"

minOccurs: "1"

maxOccurs: "1"

type: "master"

minver: "1"

documentation: Specifies which other elements the metadata represented by the Tag applies to. If empty or not present, then the Tag describes everything in the Segment.

8.2.231. TargetTypeValue Element

name: "TargetTypeValue"

path: "0*1(\\Segment\\Tags\\Tag\\Targets\\TargetTypeValue)"

id: "0x68CA"

maxOccurs: "1"

default: "50"

type: "uinteger"

minver: "1"

documentation: A number to indicate the logical level of the target.

8.2.232. TargetType Element

name: "TargetType"

path: "0*1(\\Segment\\Tags\\Tag\\Targets\\TargetType)"

id: "0x63CA"

maxOccurs: "1"

type: "string"

minver: "1"

documentation: An informational string that can be used to display the logical level of the target like "ALBUM", "TRACK", "MOVIE", "CHAPTER", etc (see TargetType).

8.2.233. TagTrackUID Element

name: "TagTrackUID"

path: "0*(\Segment\Tags\Tag\Targets\TagTrackUID)"

id: "0x63C5"

default: "0"

type: "uinteger"

minver: "1"

documentation: A unique ID to identify the Track(s) the tags belong to. If the value is 0 at this level, the tags apply to all tracks in the Segment.

8.2.234. TagEditionUID Element

name: "TagEditionUID"

path: "0*(\Segment\Tags\Tag\Targets\TagEditionUID)"

id: "0x63C9"

default: "0"

type: "uinteger"

minver: "1"

documentation: A unique ID to identify the EditionEntry(s) the tags belong to. If the value is 0 at this level, the tags apply to all editions in the Segment.

8.2.235. TagChapterUID Element

name: "TagChapterUID"

path: "0*(\Segment\Tags\Tag\Targets\TagChapterUID)"

id: "0x63C4"

default: "0"

type: "uinteger"

minver: "1"

documentation: A unique ID to identify the Chapter(s) the tags belong to. If the value is 0 at this level, the tags apply to all chapters in the Segment.

8.2.236. TagAttachmentUID Element

name: "TagAttachmentUID"

path: "0*(\Segment\Tags\Tag\Targets\TagAttachmentUID)"

id: "0x63C6"

default: "0"

type: "uinteger"

minver: "1"

documentation: A unique ID to identify the Attachment(s) the tags belong to. If the value is 0 at this level, the tags apply to all the attachments in the Segment.

8.2.237. SimpleTag Element

name: "SimpleTag"

path: "1*(\Segment\Tags\Tag(1*(\SimpleTag)))"

id: "0x67C8"

minOccurs: "1"

type: "master"

recursive: "1"

minver: "1"

documentation: Contains general information about the target.

8.2.238. TagName Element

name: "TagName"

path: "1*1(\Segment\Tags\Tag\SimpleTag\TagName)"

id: "0x45A3"

minOccurs: "1"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: The name of the Tag that is going to be stored.

8.2.239. TagLanguage Element

name: "TagLanguage"

path: "1*1(\Segment\Tags\Tag\SimpleTag\TagLanguage)"

id: "0x447A"

minOccurs: "1"

maxOccurs: "1"

default: "und"

type: "string"

minver: "1"

documentation: Specifies the language of the tag specified, in the Matroska languages form. This Element MUST be ignored if the TagLanguageIETF Element is used within the same SimpleTag Element.

8.2.240. TagLanguageIETF Element

name: "TagLanguageIETF"

path: "0*1(\Segment\Tags\Tag\SimpleTag\TagLanguageIETF)"

id: "0x447B"

maxOccurs: "1"

type: "string"

minver: "4"

documentation: Specifies the language used in the TagString according to [BCP 47](#) and using the IANA Language Subtag Registry. If this Element is used, then any TagLanguage Elements used in the same SimpleTag MUST be ignored.

[8.2.241.](#) TagDefault Element

name: "TagDefault"

path: "1*1(\Segment\Tags\Tag\SimpleTag\TagDefault)"

id: "0x4484"

minOccurs: "1"

maxOccurs: "1"

range: "0-1"

default: "1"

type: "uinteger"

minver: "1"

documentation: A boolean value to indicate if this is the default/original language to use for the given tag.

[8.2.242.](#) TagString Element

name: "TagString"

path: "0*1(\Segment\Tags\Tag\SimpleTag\TagString)"

id: "0x4487"

maxOccurs: "1"

type: "utf-8"

minver: "1"

documentation: The value of the Tag.

8.2.243. TagBinary Element

name: "TagBinary"

path: "0*1(\Segment\Tags\Tag\SimpleTag\TagBinary)"

id: "0x4485"

maxOccurs: "1"

type: "binary"

minver: "1"

documentation: The values of the Tag if it is binary. Note that this cannot be used in the same SimpleTag as TagString.

9. Matroska Element Ordering Guidelines

Except for the "EBML Header" and the "CRC-32 Element", the EBML specification does not require any particular storage order for "Elements". The Matroska specification however defines mandates and recommendations for ordering certain "Elements" in order to facilitate better playback, seeking, and editing efficiency. This section describes and offers rationale for ordering requirements and recommendations for Matroska.

9.1. Top-Level Elements

The "Info Element" is the only REQUIRED "Top-Level Element" in a Matroska file. To be playable, Matroska MUST also contain at least one "Tracks Element" and "Cluster Element". The first "Info Element" and the first "Tracks Element" MUST either be stored before the first "Cluster Element" or both SHALL be referenced by a "SeekHead Element" occurring before the first "Cluster Element".

It is possible to edit a Matroska file after it has been created. For example, chapters, tags or attachments can be added. When new "Top-Level Elements" are added to a Matroska file, the "SeekHead" Element(s) MUST be updated so that the "SeekHead" Element(s) itemize the identity and position of all "Top-Level Elements". Editing, removing, or adding "Elements" to a Matroska file often requires that some existing "Elements" be voided or extended; therefore, it is RECOMMENDED to use "Void Elements" as padding in between "Top-Level Elements".

9.2. CRC-32

As noted by the EBML specification, if a "CRC-32 Element" is used then the "CRC-32 Element" MUST be the first ordered "Element" within its "Parent Element". The Matroska specification recommends that "CRC-32 Elements" SHOULD NOT be used as an immediate "Child Element" of the "Segment Element"; however all "Top-Level Elements" of an "EBML Document" SHOULD include a "CRC-32 Element" as a "Child Element".

9.3. SeekHead

If used, the first "SeekHead Element" SHOULD be the first non-"CRC-32 Child Element" of the "Segment Element". If a second "SeekHead Element" is used, then the first "SeekHead Element" MUST reference the identity and position of the second "SeekHead". Additionally, the second "SeekHead Element" MUST only reference "Cluster" Elements and not any other "Top-Level Element" already contained within the first "SeekHead Element". The second "SeekHead Element" MAY be stored in any order relative to the other "Top-Level Elements." Whether one or two "SeekHead Element(s)" are used, the "SeekHead Element(s)" MUST collectively reference the identity and position of all "Top-Level Elements" except for the first "SeekHead Element".

It is RECOMMENDED that the first "SeekHead Element" be followed by a "Void Element" to allow for the "SeekHead Element" to be expanded to cover new "Top-Level Elements" that could be added to the Matroska file, such as "Tags", "Chapters" and "Attachments Elements".

9.4. Cues (index)

The "Cues Element" is RECOMMENDED to optimize seeking access in Matroska. It is programmatically simpler to add the "Cues Element" after all "Cluster Elements" have been written because this does not require a prediction of how much space to reserve before writing the "Cluster Elements". However, storing the "Cues Element" before the "Cluster Elements" can provide some seeking advantages. If the "Cues Element" is present, then it SHOULD either be stored before the first "Cluster Element" or be referenced by a "SeekHead Element".

9.5. Info

The first "Info Element" SHOULD occur before the first "Tracks Element" and first "Cluster Element" except when referenced by a "SeekHead Element".

9.6. Chapters

The "Chapters Element" SHOULD be placed before the "Cluster Element(s)". The "Chapters Element" can be used during playback even if the user does not need to seek. It immediately gives the user information about what section is being read and what other sections are available. In the case of Ordered Chapters it RECOMMENDED to evaluate the logical linking even before playing. The "Chapters Element" SHOULD be placed before the first "Tracks Element" and after the first "Info Element".

9.7. Attachments

The "Attachments Element" is not intended to be used by default when playing the file, but could contain information relevant to the content, such as cover art or fonts. Cover art is useful even before the file is played and fonts could be needed before playback starts for initialization of subtitles. The "Attachments Element" MAY be placed before the first "Cluster Element"; however if the "Attachments Element" is likely to be edited, then it SHOULD be placed after the last "Cluster Element".

9.8. Tags

The "Tags Element" is most subject to changes after the file was originally created. For easier editing, the "Tags Element" SHOULD be placed at the end of the "Segment Element", even after the "Attachments Element". On the other hand, it is inconvenient to have to seek in the "Segment" for tags, especially for network streams. So it's better if the "Tags Element" is found early in the stream. When editing the "Tags Element", the original "Tags Element" at the beginning can be overwritten with a "Void Element" and a new "Tags Element" written at the end of the "Segment Element". The file size will only marginally change.

9.9. Optimum layout from a muxer

- o SeekHead
- o Info
- o Tracks
- o Chapters
- o Attachments
- o Tags

- o Clusters
- o Cues

9.10. Optimum layout after editing tags

- o SeekHead
- o Info
- o Tracks
- o Chapters
- o Attachments
- o Void
- o Clusters
- o Cues
- o Tags

9.11. Optimum layout with Cues at the front

- o SeekHead
- o Info
- o Tracks
- o Chapters
- o Attachments
- o Tags
- o Cues
- o Clusters

9.12. Cluster Timecode

The "Timecode Element" MUST occur as in storage order before any "SimpleBlock", "BlockGroup", or "EncryptedBlock" within the "Cluster Element".

10. Chapters

10.1. Edition and Chapter Flags

10.1.1. Chapter Flags

Two "Chapter Flags" are defined to describe the behavior of the "ChapterAtom Element": "ChapterFlagHidden" and "ChapterFlagEnabled".

If a "ChapterAtom Element" is the "Child Element" of another "ChapterAtom Element" with a "Chapter Flag" set to "true", then the "Child ChapterAtom Element" MUST be interpreted as having its same "Chapter Flag" set to "true". If a "ChapterAtom Element" is the "Child Element" of another "ChapterAtom Element" with a "Chapter Flag" set to "false" or if the "ChapterAtom Element" does not have a "ChapterAtom Element" as its "Parent Element", then it MUST be interpreted according to its own "Chapter Flag".

As an example, consider a "Parent ChapterAtom Element" that has its "ChapterFlagHidden" set to "true" and also contains two child "ChapterAtoms", the first with "ChapterFlagHidden" set to "true" and the second with "ChapterFlagHidden" either set to "false" or not present at all (in which case the default value of the Element applies, which is "false"). Since the parent "ChapterAtom" has its "ChapterFlagHidden" set to "true" then all of its children "ChapterAtoms" MUST also be interpreted as if their "ChapterFlagHidden" is also set to "true". However, if a "Control Track" toggles the parent's "ChapterFlagHidden" flag to "false", then only the parent "ChapterAtom" and its second child "ChapterAtom" MUST be interpreted as if "ChapterFlagHidden" is set to "false". The first child "ChapterAtom" which has the "ChapterFlagHidden" flag set to "true" retains its value until its value is toggled to "false" by a "Control Track".

10.1.2. Edition Flags

Three "Edition Flags" are defined to describe the behavior of the "EditionEntry Element": "EditionFlagHidden", "EditionFlagDefault" and "EditionFlagOrdered".

10.1.2.1. EditionFlagHidden

The "EditionFlagHidden Flag" behaves similarly to the "ChapterFlagHidden Flag": if "EditionFlagHidden" is set to "true", its "Child ChapterAtoms Elements" MUST also be interpreted as if their "ChapterFlagHidden" is also set to "true", regardless of their own "ChapterFlagHidden Flags". If "EditionFlagHidden" is toggled by a "Control Track" to "false", the "ChapterFlagHidden Flags" of the

"Child ChapterAtoms Elements" SHALL determine whether the "ChapterAtom" is hidden or not.

10.1.2.2. EditionFlagDefault

It is RECOMMENDED that no more than one "Edition" have an "EditionFlagDefault Flag" set to "true". The first "Edition" with both the "EditionFlagDefault Flag" set to "true" and the "EditionFlagHidden Flag" set to "false" is the "Default Edition". When all "EditionFlagDefault Flags" are set to "false", then the first "Edition" is the "Default Edition".

10.1.2.3. EditionFlagOrdered

The "EditionFlagOrdered Flag" is a significant feature as it enables an "Edition" of "Ordered Chapters" which defines and arranges a virtual timeline rather than simply labeling points within the timeline. For example, with "Editions" of "Ordered Chapters" a single "Matroska file" can present multiple edits of a film without duplicating content. Alternatively if a videotape is digitized in full, one "Ordered Edition" could present the full content (including colorbars, countdown, slate, a feature presentation, and black frames), while another "Edition" of "Ordered Chapters" can use "Chapters" that only mark the intended presentation with the colorbars and other ancillary visual information excluded. If an "Edition" of "Ordered Chapters" is enabled then the "Matroska Player" MUST play those Chapters in their stored order from the timecode marked in the "ChapterTimeStart Element" to the timecode marked in to "ChapterTimeEnd Element".

If the "EditionFlagOrdered Flag" is set to "false", "Simple Chapters" are used and only the "ChapterTimeStart" of a "Chapter" is used as chapter mark to jump to the predefined point in the timeline. With "Simple Chapters", a "Matroska Player" MUST ignore certain "Chapter Elements". All these elements are now informational only.

The following list shows the different usage of "Chapter Elements" between an ordered and non-ordered "Edition".

Chapter elements / ordered Edition		False		True	ChapterUID		X		X
ChapterStringUID		X		X	ChapterTimeStart		X		X
ChapterTimeEnd		X		X	ChapterFlagHidden		X		X
ChapterFlagEnabled		X		X	ChapterSegmentUID		-		X
ChapterSegmentEditionUID		-		X	ChapterPhysicalEquiv		X		X
ChapterTrack		-		X	ChapterDisplay		X		X
ChapProcess		-		X					

Furthermore there are other EBML "Elements" which could be used if the "EditionFlagOrdered Flag" is set to "true".

Other elements / ordered Edition | False | True Info/SegmentFamily |
- | X Info/ChapterTranslate | - | X Track/TrackTranslate | - | X

These other "Elements" belong to the Matroska DVD menu system and are only used when the "ChapProcessCodecID Element" is set to 1.

10.1.2.3.1. Ordered-Edition and Matroska Segment-Linking

- o Hard Linking: "Ordered-Chapters" supersedes the "Hard Linking".
- o Soft Linking: In this complex system "Ordered Chapters" are REQUIRED and a "Chapter CODEC" MUST interpret the "ChapProcess" of all chapters.
- o Medium Linking: "Ordered Chapters" are used in a normal way and can be combined with the "ChapterSegmentUID" element which establishes a link to another Matroska file/Segment.

See [Section 23](#)) for more information about "Hard Linking", "Soft Linking" and "Medium Linking".

10.2. Menu features

The menu features are handled like a `_chapter codec_`. That means each codec has a type, some private data and some data in the chapters.

The type of the menu system is defined by the "ChapProcessCodecID" parameter. For now only 2 values are supported : 0 matroska script, 1 menu borrowed from the DVD. The private data depend on the type of menu system (stored in ChapProcessPrivate), idem for the data in the chapters (stored in ChapProcessData).

10.2.1. Matroska Script (0)

This is the case when "ChapProcessCodecID" = 0. This is a script language build for Matroska purposes. The inspiration comes from ActionScript, javascript and other similar scripting languages. The commands are stored as text commands, in UTF-8. The syntax is C like, with commands spanned on many lines, each terminating with a ";". You can also include comments at the end of lines with "///" or comment many lines using "/* */". The scripts are stored in ChapProcessData. For the moment ChapProcessPrivate is not used.

The one and only command existing for the moment is "GotoAndPlay(ChapterUID);". As the same suggests, it means that when this command is encountered, the "Matroska Player" SHOULD jump to the "Chapter" specified by the UID and play it.

[10.2.2.](#) DVD menu (1)

This is the case when "ChapProcessCodecID" = 1. Each level of a chapter corresponds to a logical level in the DVD system that is stored in the first octet of the ChapProcessPrivate. This DVD hierarchy is as follows:

```
ChapProcessPrivate | DVD Name | Hierarchy | Commands Possible |
Comment 0x30 | SS | DVD domain | - | First Play, Video Manager, Video
Title 0x2A | LU | Language Unit | - | Contains only PGCs 0x28 | TT |
Title | - | Contains only PGCs 0x20 | PGC | Program Group Chain
(PGC) | * | 0x18 | PG | Program 1 / Program 2 / Program 3 | - |
0x10 | PTT | Part Of Title 1 / Part Of Title 2 | - | Equivalent to
the chapters on the sleeve. 0x08 | CN | Cell 1 / Cell 2 / Cell 3 /
Cell 4 / Cell 5 / Cell 6 | - |
```

You can also recover whether a Segment is a Video Manager (VMG), Video Title Set (VTS) or Video Title Set Menu (VTSM) from the ChapterTranslateID element found in the Segment Info. This field uses 2 octets as follows:

1. Domain Type: 0 for VMG, the domain number for VTS and VTSM
2. Domain Value: 0 for VMG and VTSM, 1 for the VTS source.

For instance, the menu part from VTS_01_0.VOB would be coded [[1](#),0] and the content part from VTS_02_3.VOB would be [[2](#),[1](#)]. The VMG is always [0,0]

The following octets of ChapProcessPrivate are as follows:

```
Octet 1 | DVD Name | Following Octets 0x30 | SS | Domain name code
(1: 0x00= First play, 0xC0= VMG, 0x40= VTSM, 0x80= VTS) + VTS(M)
number (2) 0x2A | LU | Language code (2) + Language extension (1)
0x28 | TT | global Title number (2) + corresponding TTN of the VTS
(1) 0x20 | PGC | PGC number (2) + Playback Type (1) + Disabled User
Operations (4) 0x18 | PG | Program number (2) 0x10 | PTT | PTT-
chapter number (1) 0x08 | CN | Cell number [VOB ID(2)][Cell
ID(1)][Angle Num(1)]
```

If the level specified in ChapProcessPrivate is a PGC (0x20), there is an octet called the Playback Type, specifying the kind of PGC defined:

- o 0x00: entry only/basic PGC
- o 0x82: Title+Entry Menu (only found in the Video Manager domain)

- o 0x83: Root Menu (only found in the VTSM domain)
- o 0x84: Subpicture Menu (only found in the VTSM domain)
- o 0x85: Audio Menu (only found in the VTSM domain)
- o 0x86: Angle Menu (only found in the VTSM domain)
- o 0x87: Chapter Menu (only found in the VTSM domain)

The next 4 following octets correspond to the User Operation flags [\[17\]](#) in the standard PGC. When a bit is set, the command SHOULD be disabled.

ChapProcessData contains the pre/post/cell commands in binary format as there are stored on a DVD. There is just an octet preceding these data to specify the number of commands in the element. As follows:
[# of commands(1)][command 1 (8)][command 2 (8)][command 3 (8)].

More information on the DVD commands and format on DVD-replica [\[18\]](#), where we got most of the info about it. You can also get information on DVD from the DVDinfo project [\[19\]](#).

[10.3](#). Example 1 : basic chaptering

In this example a movie is split in different chapters. It could also just be an audio file (album) on which each track corresponds to a chapter.

- o 00000ms - 05000ms : Intro
- o 05000ms - 25000ms : Before the crime
- o 25000ms - 27500ms : The crime
- o 27500ms - 38000ms : The killer arrested
- o 38000ms - 43000ms : Credits

This would translate in the following matroska form :

```
<Chapters>
  <EditionEntry>
    <EditionUID>16603393396715046047</EditionUID>
    <ChapterAtom>
      <ChapterUID>1193046</ChapterUID>
      <ChapterTimeStart>0</ChapterTimeStart>
      <ChapterTimeEnd>5000000000</ChapterTimeEnd>
```



```
<ChapterDisplay>
  <ChapString>Intro</ChapString>
  <ChapLanguage>eng</ChapLanguage>
</ChapterDisplay>
<ChapterFlagHidden>0</ChapterFlagHidden>
<ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>2311527</ChapterUID>
  <ChapterTimeStart>5000000000</ChapterTimeStart>
  <ChapterTimeEnd>25000000000</ChapterTimeEnd>
  <ChapterDisplay>
    <ChapString>Before the crime</ChapString>
    <ChapLanguage>eng</ChapLanguage>
  </ChapterDisplay>
  <ChapterDisplay>
    <ChapString>Avant le crime</ChapString>
    <ChapLanguage>fra</ChapLanguage>
  </ChapterDisplay>
  <ChapterFlagHidden>0</ChapterFlagHidden>
  <ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>3430008</ChapterUID>
  <ChapterTimeStart>25000000000</ChapterTimeStart>
  <ChapterTimeEnd>27500000000</ChapterTimeEnd>
  <ChapterDisplay>
    <ChapString>The crime</ChapString>
    <ChapLanguage>eng</ChapLanguage>
  </ChapterDisplay>
  <ChapterDisplay>
    <ChapString>Le crime</ChapString>
    <ChapLanguage>fra</ChapLanguage>
  </ChapterDisplay>
  <ChapterFlagHidden>0</ChapterFlagHidden>
  <ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>4548489</ChapterUID>
  <ChapterTimeStart>27500000000</ChapterTimeStart>
  <ChapterTimeEnd>38000000000</ChapterTimeEnd>
  <ChapterDisplay>
    <ChapString>After the crime</ChapString>
    <ChapLanguage>eng</ChapLanguage>
  </ChapterDisplay>
  <ChapterDisplay>
    <ChapString>Apres le crime</ChapString>
    <ChapLanguage>fra</ChapLanguage>
```



```

    </ChapterDisplay>
    <ChapterFlagHidden>0</ChapterFlagHidden>
    <ChapterFlagEnabled>1</ChapterFlagEnabled>
  </ChapterAtom>
  <ChapterAtom>
    <ChapterUID>5666960</ChapterUID>
    <ChapterTimeStart>38000000000</ChapterTimeStart>
    <ChapterTimeEnd>43000000000</ChapterTimeEnd>
    <ChapterDisplay>
      <ChapString>Credits</ChapString>
      <ChapLanguage>eng</ChapLanguage>
    </ChapterDisplay>
    <ChapterDisplay>
      <ChapString>Generique</ChapString>
      <ChapLanguage>fra</ChapLanguage>
    </ChapterDisplay>
    <ChapterFlagHidden>0</ChapterFlagHidden>
    <ChapterFlagEnabled>1</ChapterFlagEnabled>
  </ChapterAtom>
  <EditionFlagDefault>0</EditionFlagDefault>
  <EditionFlagHidden>0</EditionFlagHidden>
</EditionEntry>
</Chapters>

```

10.4. Example 2 : nested chapters

In this example an (existing) album is split into different chapters, and one of them contain another splitting.

10.4.1. The Micronauts "Bleep To Bleep"

- o 00:00 - 12:28 : Baby Wants To Bleep/Rock
 - * 00:00 - 04:38 : Baby wants to bleep (pt.1)
 - * 04:38 - 07:12 : Baby wants to rock
 - * 07:12 - 10:33 : Baby wants to bleep (pt.2)
 - * 10:33 - 12:28 : Baby wants to bleep (pt.3)
- o 12:30 - 19:38 : Bleeper_0+2
- o 19:40 - 22:20 : Baby wants to bleep (pt.4)
- o 22:22 - 25:18 : Bleep to bleep
- o 25:20 - 33:35 : Baby wants to bleep (k)

- o 33:37 - 44:28 : Bleeper

```
<Chapters>
  <EditionEntry>
    <EditionUID>1281690858003401414</EditionUID>
    <ChapterAtom>
      <ChapterUID>1</ChapterUID>
      <ChapterTimeStart>0</ChapterTimeStart>
      <ChapterTimeEnd>748000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Baby wants to Bleep/Rock</ChapString>
        <ChapLanguage>eng</ChapLanguage>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>2</ChapterUID>
      <ChapterTimeStart>0</ChapterTimeStart>
      <ChapterTimeEnd>278000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Baby wants to bleep (pt.1)</ChapString>
        <ChapLanguage>eng</ChapLanguage>
      </ChapterDisplay>
      <ChapterFlagHidden>0</ChapterFlagHidden>
      <ChapterFlagEnabled>1</ChapterFlagEnabled>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>3</ChapterUID>
      <ChapterTimeStart>278000000</ChapterTimeStart>
      <ChapterTimeEnd>432000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Baby wants to rock</ChapString>
        <ChapLanguage>eng</ChapLanguage>
      </ChapterDisplay>
      <ChapterFlagHidden>0</ChapterFlagHidden>
      <ChapterFlagEnabled>1</ChapterFlagEnabled>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>4</ChapterUID>
      <ChapterTimeStart>432000000</ChapterTimeStart>
      <ChapterTimeEnd>633000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Baby wants to bleep (pt.2)</ChapString>
        <ChapLanguage>eng</ChapLanguage>
      </ChapterDisplay>
      <ChapterFlagHidden>0</ChapterFlagHidden>
      <ChapterFlagEnabled>1</ChapterFlagEnabled>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>5</ChapterUID>
```



```
<ChapterTimeStart>633000000</ChapterTimeStart>
<ChapterTimeEnd>748000000</ChapterTimeEnd>
<ChapterDisplay>
  <ChapString>Baby wants to bleep (pt.3)</ChapString>
  <ChapLanguage>eng</ChapLanguage>
</ChapterDisplay>
<ChapterFlagHidden>0</ChapterFlagHidden>
<ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterFlagHidden>0</ChapterFlagHidden>
<ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>6</ChapterUID>
  <ChapterTimeStart>750000000</ChapterTimeStart>
  <ChapterTimeEnd>1178500000</ChapterTimeEnd>
  <ChapterDisplay>
    <ChapString>Bleeper_0+2</ChapString>
    <ChapLanguage>eng</ChapLanguage>
  </ChapterDisplay>
  <ChapterFlagHidden>0</ChapterFlagHidden>
  <ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>7</ChapterUID>
  <ChapterTimeStart>1180500000</ChapterTimeStart>
  <ChapterTimeEnd>1340000000</ChapterTimeEnd>
  <ChapterDisplay>
    <ChapString>Baby wants to bleep (pt.4)</ChapString>
    <ChapLanguage>eng</ChapLanguage>
  </ChapterDisplay>
  <ChapterFlagHidden>0</ChapterFlagHidden>
  <ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>8</ChapterUID>
  <ChapterTimeStart>1342000000</ChapterTimeStart>
  <ChapterTimeEnd>1518000000</ChapterTimeEnd>
  <ChapterDisplay>
    <ChapString>Bleep to bleep</ChapString>
    <ChapLanguage>eng</ChapLanguage>
  </ChapterDisplay>
  <ChapterFlagHidden>0</ChapterFlagHidden>
  <ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>9</ChapterUID>
  <ChapterTimeStart>1520000000</ChapterTimeStart>
```



```
<ChapterTimeEnd>2015000000</ChapterTimeEnd>
<ChapterDisplay>
  <ChapString>Baby wants to bleep (k)</ChapString>
  <ChapLanguage>eng</ChapLanguage>
</ChapterDisplay>
<ChapterFlagHidden>0</ChapterFlagHidden>
<ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<ChapterAtom>
  <ChapterUID>10</ChapterUID>
  <ChapterTimeStart>2017000000</ChapterTimeStart>
  <ChapterTimeEnd>2668000000</ChapterTimeEnd>
  <ChapterDisplay>
    <ChapString>Bleeper</ChapString>
    <ChapLanguage>eng</ChapLanguage>
  </ChapterDisplay>
  <ChapterFlagHidden>0</ChapterFlagHidden>
  <ChapterFlagEnabled>1</ChapterFlagEnabled>
</ChapterAtom>
<EditionFlagDefault>0</EditionFlagDefault>
<EditionFlagHidden>0</EditionFlagHidden>
</EditionEntry>
</Chapters>
```

11. Attachments

11.1. Introduction

Matroska supports storage of related files and data in the "Attachments Element" (a "Top-Level Element"). "Attachment Elements" can be used to store related cover art, font files, transcripts, reports, error recovery files, picture or text-based annotations, copies of specifications, or other ancillary files related to the "Segment".

"Matroska Readers" MUST NOT execute files stored as "Attachment Elements".

11.2. Cover Art

This section defines a set of guidelines for the storage of cover art in Matroska files. A "Matroska Reader" MAY use embedded cover art to display a representational still-image depiction of the multimedia contents of the Matroska file.

Only JPEG and PNG image formats SHOULD be used for cover art pictures.

There can be two different covers for a movie/album: a portrait style (e.g., a DVD case) and a landscape style (e.g., a wide banner ad).

There can be two versions of the same cover, the "normal cover" and the "small cover". The dimension of the "normal cover" SHOULD be 600 pixels on the smallest side (for example, 960x600 for landscape, 600x800 for portrait, or 600x600 for square). The dimension of the "small cover" SHOULD be 120 pixels on the smallest side (for example, 192x120 or 120x160).

Versions of cover art can be differentiated by the filename, which is stored in the "FileName Element". The default filename of the "normal cover" in square or portrait mode is "cover.(jpg|png)". When stored, the "normal cover" SHOULD be the first Attachment in storage order. The "small cover" SHOULD be prefixed with "small_", such as "small_cover.(jpg|png)". The landscape variant SHOULD be suffixed with "_land", such as "cover_land.(jpg|png)". The filenames are case sensitive.

The following table provides examples of file names for cover art in Attachments.

FileName	Image Orientation	Pixel Length of Smallest Side
cover.jpg	Portrait or square	600
small_cover.png	Portrait or square	120
cover_land.png	Landscape	600
small_cover_land.jpg	Landscape	120

12. Cues

12.1. Introduction

The "Cues Element" provides an index of certain "Cluster Elements" to allow for optimized seeking to absolute timestamps within the "Segment". The "Cues Element" contains one or many "CuePoint Elements" which each MUST reference an absolute timestamp (via the "CueTime Element"), a "Track" (via the "CueTrack Element"), and a "Segment Position" (via the "CueClusterPosition Element"). Additional non-mandated Elements are part of the "CuePoint Element" such as "CueDuration", "CueRelativePosition", "CueCodecState" and others which provide any "Matroska Reader" with additional information to use in the optimization of seeking performance.

12.2. Recommendations

The following recommendations are provided to optimize Matroska performance.

- o Unless Matroska is used as a live stream, it SHOULD contain a "Cues Element".
- o For each video track, each keyframe SHOULD be referenced by a "CuePoint Element".
- o It is RECOMMENDED to not reference non-keyframes of video tracks in "Cues" unless it references a "Cluster Element" which contains a "CodecState Element" but no keyframes.
- o For each subtitle track present, each subtitle frame SHOULD be referenced by a "CuePoint Element" with a "CueDuration Element".
- o References to audio tracks MAY be skipped in "CuePoint Elements" if a video track is present. When included the "CuePoint Elements" SHOULD reference audio keyframes at most once every 500 milliseconds.
- o If the referenced frame is not stored within the first "SimpleBlock" or first "BlockGroup" within its "Cluster Element", then the "CueRelativePosition Element" SHOULD be written to reference where in the "Cluster" the reference frame is stored.
- o If a "CuePoint Element" references "Cluster Element" that includes a "CodecState Element", then that "CuePoint Element" MUST use a "CueCodecState Element".
- o "CuePoint Elements" SHOULD be numerically sorted in storage order by the value of the "CueTime Element".

13. Matroska Streaming

In Matroska, there are two kinds of streaming: file access and livestreaming.

13.1. File Access

File access can simply be reading a file located on your computer, but also includes accessing a file from an HTTP (web) server or CIFS (Windows share) server. These protocols are usually safe from reading errors and seeking in the stream is possible. However, when a file is stored far away or on a slow server, seeking can be an expensive operation and SHOULD be avoided. The following guidelines, when followed, help reduce the number of seeking operations for regular playback and also have the playback start quickly without a lot of data needed to read first (like a "Cues Element", "Attachment Element" or "SeekHead Element").

Matroska, having a small overhead, is well suited for storing music/videos on file servers without a big impact on the bandwidth used. Matroska does not require the index to be loaded before playing, which allows playback to start very quickly. The index can be loaded only when seeking is requested the first time.

13.2. Livestreaming

Livestreaming is the equivalent of television broadcasting on the internet. There are 2 families of servers for livestreaming: RTP/RTSP and HTTP. Matroska is not meant to be used over RTP. RTP already has timing and channel mechanisms that would be wasted if doubled in Matroska. Additionally, having the same information at the RTP and Matroska level would be a source of confusion if they do not match. Livestreaming of Matroska over HTTP (or any other plain protocol based on TCP) is possible.

A live Matroska stream is different from a file because it usually has no known end (only ending when the client disconnects). For this, all bits of the "size" portion of the "Segment Element" MUST be set to 1. Another option is to concatenate "Segment Elements" with known sizes, one after the other. This solution allows a change of codec/resolution between each segment. For example, this allows for a switch between 4:3 and 16:9 in a television program.

When "Segment Elements" are continuous, certain "Elements", like "MetaSeek", "Cues", "Chapters", and "Attachments", MUST NOT be used.

It is possible for a "Matroska Player" to detect that a stream is not seekable. If the stream has neither a "MetaSeek" list or a "Cues" list at the beginning of the stream, it SHOULD be considered non-seekable. Even though it is possible to seek blindly forward in the stream, it is NOT RECOMMENDED.

In the context of live radio or web TV, it is possible to "tag" the content while it is playing. The "Tags Element" can be placed between "Clusters" each time it is necessary. In that case, the new "Tags Element" MUST reset the previously encountered "Tags Elements" and use the new values instead.

14. Menu Specifications

14.1. Introduction

This document is a _draft of the Menu system_ that will be the default one in "Matroska". As it will just be composed of a Control Track, it will be seen as a "codec" and could be replaced later by something else if needed.

A menu is like what you see on DVDs, when you have some screens to select the audio format, subtitles or scene selection.

14.2. Requirements

What we'll try to have is a system that can do almost everything done on a DVD, or more, or better, or drop the unused features if necessary.

As the name suggests, a Control Track is a track that can control the playback of the file and/or all the playback features. To make it as simple as possible for "Matroska Players", the Control Track will just give orders to the "Matroska Player" and get the actions associated with the highlights/hotspots.

14.2.1. Highlights/Hotspots

A highlight is basically a rectangle/key associated with an action UID. When that rectangle/key is activated, the "Matroska Player" send the UID of the action to the Control Track handler (codec). The fact that it can also be a key means that even for audio only files, a keyboard shortcut or button panel could be used for menus. But in that case, the hotspot will have to be associated with a name to display.

This highlight is sent from the Control Track to the "Matroska Player". Then the "Matroska Player" has to handle that highlight until it's deactivated (see [Section 14.2.2](#)).

The highlight contains a UID of the action, a displayable name (UTF-8), an associated key (list of keys to be defined, probably up/down/left/right/select), a screen position/range and an image to display. The image will be displayed either when the user place the mouse over the rectangle (or any other shape), or when an option of the screen is selected (not activated). There could be a second image used when the option is activated. And there could be a third image that can serve as background. This way you could have a still image (like in some DVDs) for the menu and behind that image blank video (small bitrate).

When a highlight is activated by the user, the "Matroska Player" has to send the UID of the action to the Control Track. Then the Control Track codec will handle the action and possibly give new orders to the "Matroska Player".

The format used for storing images SHOULD be extensible. For the moment we'll use PNG and BMP, both with alpha channel.

14.2.2. Playback features

All the following features will be sent from the Control Track to the "Matroska Player" :

- o Jump to chapter (UID, prev, next, number)
- o Disable all tracks of a kind (audio, video, subtitle)
- o Enable track UID (the kind doesn't matter)
- o Define/Disable a highlight
- o Enable/Disable jumping
- o Enable/Disable track selection of a kind
- o Select Edition ID (see chapters)
- o Pause playback
- o Stop playback
- o Enable/Disable a Chapter UID
- o Hide/Unhide a Chapter UID

All the actions will be written in a normal Matroska track, with a timecode. A "Menu Frame" SHOULD be able to contain more that one action/highlight for a given timecode. (to be determined, EBML format structure)

14.2.3. Player requirements

Some "Matroska Players" might not support the control track. That mean they will play the active/looped parts as part of the data. So I suggest putting the active/looped parts of a movie at the end of a movie. When a Menu-aware "Matroska Player" encounter the default Control Track of a "Matroska" file, the first order SHOULD be to jump at the start of the active/looped part of the movie.

14.3. Working Graph

Matroska Source file -> Control Track <-> Player.
-> other tracks -> rendered

14.4. Ideas

!!!! KNOW Where the main/audio/subs menu starts wherever we are (use chapters) !!!!

!!!! Keep in mind the state of the selected tracks of each kind (more than 1 for each possible) !!!!

!!!! Order of blending !!!!

!!!! What if a command is not supported by the player ? !!!!

!!!! Track selection issue, only applies when 'quitting' the menu (but still possible to change live too) !!!!

!!!! Allow to hide (not render) some parts of a movie for certain editions !!!!

!!!! Get the parental level of the player (can be changed live) !!!!

14.5. Data Structure

As a Matroska side project, the obvious choice for storing binary data is EBML.

15. Unknown elements

Matroska is based upon the principle that a reading application does not have to support 100% of the specifications in order to be able to play the file. A Matroska file therefore contains version indicators that tell a reading application what to expect.

It is possible and valid to have the version fields indicate that the file contains Matroska "Elements" from a higher specification version number while signaling that a reading application MUST only support a lower version number properly in order to play it back (possibly with a reduced feature set). For example, a reading application supporting at least Matroska version "V" reading a file whose "DocTypeReadVersion" field is equal to or lower than "V" MUST skip Matroska/EBML "Elements" it encounters but does not know about if that unknown element fits into the size constraints set by the current "Parent Element".

16. Default Values

The default value of an "Element" is assumed when not present in the data stream. It is assumed only in the scope of its "Parent Element". For example, the "Language Element" is in the scope of the "Track Element". If the "Parent Element" is not present or assumed, then the "Child Element" cannot be assumed.

17. DefaultDecodedFieldDuration

The "DefaultDecodedFieldDuration Element" can signal to the displaying application how often fields of a video sequence will be

available for displaying. It can be used for both interlaced and progressive content. If the video sequence is signaled as

interlaced, then the period between two successive fields at the output of the decoding process equals "DefaultDecodedFieldDuration".

For video sequences signaled as progressive, it is twice the value of "DefaultDecodedFieldDuration".

These values are valid at the end of the decoding process before post-processing (such as deinterlacing or inverse telecine) is applied.

Examples:

- o Blu-ray movie: $1000000000\text{ns}/(48/1.001) = 20854167\text{ns}$
- o PAL broadcast/DVD: $1000000000\text{ns}/(50/1.000) = 20000000\text{ns}$
- o N/ATSC broadcast: $1000000000\text{ns}/(60/1.001) = 16683333\text{ns}$
- o hard-telecined DVD: $1000000000\text{ns}/(60/1.001) = 16683333\text{ns}$ (60 encoded interlaced fields per second)
- o soft-telecined DVD: $1000000000\text{ns}/(60/1.001) = 16683333\text{ns}$ (48 encoded interlaced fields per second, with "repeat_first_field = 1")

18. Encryption

Encryption in Matroska is designed in a very generic style to allow people to implement whatever form of encryption is best for them. It is possible to use the encryption framework in Matroska as a type of DRM (Digital Rights Management).

Because encryption occurs within the "Block Element", it is possible to manipulate encrypted streams without decrypting them. The streams could potentially be copied, deleted, cut, appended, or any number of other possible editing techniques without decryption. The data can be used without having to expose it or go through the decrypting process.

Encryption can also be layered within Matroska. This means that two completely different types of encryption can be used, requiring two separate keys to be able to decrypt a stream.

Encryption information is stored in the "ContentEncodings Element" under the "ContentEncryption Element".

19. Image cropping

The "PixelCrop Elements" ("PixelCropTop", "PixelCropBottom", "PixelCropRight" and "PixelCropLeft") indicate when and by how much encoded videos frames SHOULD be cropped for display. These Elements allow edges of the frame that are not intended for display, such as the sprockets of a full-frame film scan or the VANC area of a digitized analog videotape, to be stored but hidden. "PixelCropTop" and "PixelCropBottom" store an integer of how many rows of pixels SHOULD be cropped from the top and bottom of the image (respectively). "PixelCropLeft" and "PixelCropRight" store an integer of how many columns of pixels SHOULD be cropped from the left and right of the image (respectively). For example, a pillar-boxed video that stores a 1440x1080 visual image within the center of a padded 1920x1080 encoded image MAY set both "PixelCropLeft" and "PixelCropRight" to "240", so that a "Matroska Player" SHOULD crop off 240 columns of pixels from the left and right of the encoded image to present the image with the pillar-boxes hidden.

20. Matroska versioning

The "EBML Header" of each Matroska document informs the reading application on what version of Matroska to expect. The "Elements" within "EBML Header" with jurisdiction over this information are "DocTypeVersion" and "DocTypeReadVersion".

"DocTypeVersion" MUST be equal to or greater than the highest Matroska version number of any "Element" present in the Matroska file. For example, a file using the "SimpleBlock Element" MUST have a "DocTypeVersion" equal to or greater than 2. A file containing "CueRelativePosition" Elements MUST have a "DocTypeVersion" equal to or greater than 4.

The "DocTypeReadVersion" MUST contain the minimum version number that a reading application can minimally support in order to play the file back -- optionally with a reduced feature set. For example, if a file contains only "Elements" of version 2 or lower except for "CueRelativePosition" (which is a version 4 Matroska "Element"), then "DocTypeReadVersion" SHOULD still be set to 2 and not 4 because evaluating "CueRelativePosition" is not necessary for standard playback -- it makes seeking more precise if used.

"DocTypeVersion" MUST always be equal to or greater than "DocTypeReadVersion".

A reading application supporting Matroska version "V" MUST NOT refuse to read an application with "DocReadTypeVersion" equal to or lower

than "V" even if "DocTypeVersion" is greater than "V". See also the note about [Section 15](#).

21. MIME Types

There is no IETF endorsed MIME type for Matroska files. These definitions can be used:

- o .mka : Matroska audio "audio/x-matroska"
- o .mkv : Matroska video "video/x-matroska"
- o .mk3d : Matroska 3D video "video/x-matroska-3d"

22. Segment Position

The "Segment Position" of an "Element" refers to the position of the first octet of the "Element ID" of that "Element", measured in octets, from the beginning of the "Element Data" section of the containing "Segment Element". In other words, the "Segment Position" of an "Element" is the distance in octets from the beginning of its containing "Segment Element" minus the size of the "Element ID" and "Element Data Size" of that "Segment Element". The "Segment Position" of the first "Child Element" of the "Segment Element" is 0. An "Element" which is not stored within a "Segment Element", such as the "Elements" of the "EBML Header", do not have a "Segment Position".

22.1. Segment Position Exception

"Elements" that are defined to store a "Segment Position" MAY define reserved values to indicate a special meaning.

22.2. Example of Segment Position

This table presents an example of "Segment Position" by showing a hexadecimal representation of a very small Matroska file with labels to show the offsets in octets. The file contains a "Segment Element" with an "Element ID" of "0x18538067" and a "MuxingApp Element" with an "Element ID" of "0x4D80".

0	1																2			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
+---																				
0	1A 45 DF A3 8B 42 82 88 6D 61 74 72 6F 73 6B 61 18 53 80 67																			
20	93 15 49 A9 66 8E 4D 80 84 69 65 74 66 57 41 84 69 65 74 66																			

In the above example, the "Element ID" of the "Segment Element" is stored at offset 16, the "Element Data Size" of the "Segment Element" is stored at offset 20, and the "Element Data" of the "Segment Element" is stored at offset 21.

The "MuxingApp Element" is stored at offset 26. Since the "Segment Position" of an "Element" is calculated by subtracting the position of the "Element Data" of the containing "Segment Element" from the position of that "Element", the "Segment Position" of "MuxingApp Element" in the above example is "26 - 21" or "5".

23. Linked Segments

Matroska provides several methods to link two or many "Segment Elements" together to create a "Linked Segment". A "Linked Segment" is a set of multiple "Segments" related together into a single presentation by using Hard Linking, Medium Linking, or Soft Linking. All "Segments" within a "Linked Segment" MUST utilize the same track numbers and timescale. All "Segments" within a "Linked Segment" MUST be stored within the same directory. All "Segments" within a "Linked Segment" MUST store a "SegmentUID".

23.1. Hard Linking

Hard Linking (also called splitting) is the process of creating a "Linked Segment" by relating multiple "Segment Elements" using the "PrevUID" and "NextUID Elements". Within a "Linked Segment", the timestamps of each "Segment" MUST follow consecutively in linking order. With Hard Linking, the chapters of any "Segment" within the "Linked Segment" MUST only reference the current "Segment". With Hard Linking, the "NextUID" and "PrevUID" MUST reference the respective "SegmentUID" values of the next and previous "Segments". The first "Segment" of a "Linked Segment" MUST have a "NextUID Element" and MUST NOT have a "PrevUID Element". The last "Segment" of a "Linked Segment" MUST have a "PrevUID Element" and MUST NOT have a "NextUID Element". The middle "Segments" of a "Linked Segment" MUST have both a "NextUID Element" and a "PrevUID Element".

As an example, four "Segments" can be Hard Linked as a "Linked Segment" through cross-referencing each other with "SegmentUID", "PrevUID", and "NextUID", as in this table.

file name	SegmentUID	PrevUID	NextUID
"start .mkv"	"71000c23cd31099 853fbc94dd984a5d d"	n/a	"a77b3598941cb803 eac0fcdafe44fac9"
"middl e.mkv"	"a77b3598941cb80 3eac0fcdafe44fac 9"	"71000c23cd310998 53fbc94dd984a5dd"	"6c92285fa6d3e827 b198d120ea3ac674"
"end.m kv"	"6c92285fa6d3e82 7b198d120ea3ac67 4"	"a77b3598941cb803 eac0fcdafe44fac9"	n/a

23.2. Medium Linking

Medium Linking creates relationships between "Segments" using Ordered Chapters and the "ChapterSegmentUID Element". A "Segment Edition" with Ordered Chapters MAY contain "Chapter Elements" that reference timestamp ranges from other "Segments". The "Segment" referenced by the Ordered Chapter via the "ChapterSegmentUID Element" SHOULD be played as part of a Linked Segment. The timestamps of Segment content referenced by Ordered Chapters MUST be adjusted according to the cumulative duration of the the previous Ordered Chapters.

As an example a file named "intro.mkv" could have a "SegmentUID" of "0xb16a58609fc7e60653a60c984fc11ead". Another file called "program.mkv" could use a Chapter Edition that contains two Ordered Chapters. The first chapter references the "Segment" of "intro.mkv" with the use of a "ChapterSegmentUID", "ChapterSegmentEditionUID", "ChapterTimeStart" and optionally a "ChapterTimeEnd" element. The second chapter references content within the "Segment" of "program.mkv". A "Matroska Player" SHOULD recognize the "Linked Segment" created by the use of "ChapterSegmentUID" in an enabled "Edition" and present the reference content of the two "Segments" together.

23.3. Soft Linking

Soft Linking is used by codec chapters. They can reference another "Segment" and jump to that "Segment". The way the "Segments" are described are internal to the chapter codec and unknown to the Matroska level. But there are "Elements" within the "Info Element" (such as "ChapterTranslate") that can translate a value representing a "Segment" in the chapter codec and to the current "SegmentUID". All "Segments" that could be used in a "Linked Segment" in this way SHOULD be marked as members of the same family via the "SegmentFamily

Element", so that the "Matroska Player" can quickly switch from one to the other.

24. Track Flags

24.1. Default flag

The "default track" flag is a hint for a "Matroska Player" and SHOULD always be changeable by the user. If the user wants to see or hear a track of a certain kind (audio, video, subtitles) and hasn't chosen a specific track, the "Matroska Player" SHOULD use the first track of that kind whose "default track" flag is set to "1". If no such track is found then the first track of this kind SHOULD be chosen.

Only one track of a kind MAY have its "default track" flag set in a segment. If a track entry does not contain the "default track" flag element then its default value "1" is to be used.

24.2. Forced flag

The "forced" flag tells the "Matroska Player" that it MUST display/play this track or another track of the same kind that also has its "forced" flag set. When there are multiple "forced" tracks, the "Matroska Player" SHOULD determine the track based upon the language of the forced flag or use the default flag if no track matches the use languages. Another track of the same kind without the "forced" flag may be use simultaneously with the "forced" track (like DVD subtitles for example).

24.3. Track Operation

"TrackOperation" allows combining multiple tracks to make a virtual one. It uses two separate system to combine tracks. One to create a 3D "composition" (left/right/background planes) and one to simplify join two tracks together to make a single track.

A track created with "TrackOperation" is a proper track with a UID and all its flags. However the codec ID is meaningless because each "sub" track needs to be decoded by its own decoder before the "operation" is applied. The "Cues Elements" corresponding to such a virtual track SHOULD be the sum of the "Cues Elements" for each of the tracks it's composed of (when the "Cues" are defined per track).

In the case of "TrackJoinBlocks", the "Block Elements" (from "BlockGroup" and "SimpleBlock") of all the tracks SHOULD be used as if they were defined for this new virtual "Track". When two "Block Elements" have overlapping start or end timecodes, it's up to the underlying system to either drop some of these frames or render them

the way they overlap. This situation SHOULD be avoided when creating such tracks as you can never be sure of the end result on different platforms.

[24.4.](#) Overlay Track

Overlay tracks SHOULD be rendered in the same 'channel' as the track its linked to. When content is found in such a track, it SHOULD be played on the rendering channel instead of the original track.

[24.5.](#) Multi-planar and 3D videos

There are two different ways to compress 3D videos: have each 'eye' track in a separate track and have one track have both 'eyes' combined inside (which is more efficient, compression-wise). Matroska supports both ways.

For the single track variant, there is the "StereoMode Element" which defines how planes are assembled in the track (mono or left-right combined). Odd values of StereoMode means the left plane comes first for more convenient reading. The pixel count of the track ("PixelWidth"/"PixelHeight") is the raw amount of pixels (for example 3840x1080 for full HD side by side) and the "DisplayWidth"/"DisplayHeight" in pixels is the amount of pixels for one plane (1920x1080 for that full HD stream). Old stereo 3D were displayed using anaglyph (cyan and red colours separated). For compatibility with such movies, there is a value of the StereoMode that corresponds to AnaGlyph.

There is also a "packed" mode (values 13 and 14) which consists of packing two frames together in a "Block" using lacing. The first frame is the left eye and the other frame is the right eye (or vice versa). The frames SHOULD be decoded in that order and are possibly dependent on each other (P and B frames).

For separate tracks, Matroska needs to define exactly which track does what. "TrackOperation" with "TrackCombinePlanes" do that. For more details look at [Section 24.3](#).

The 3D support is still in infancy and may evolve to support more features.

The StereoMode used to be part of Matroska v2 but it didn't meet the requirement for multiple tracks. There was also a bug in libmatroska prior to 0.9.0 that would save/read it as 0x53B9 instead of 0x53B8. "Matroska Readers" may support these legacy files by checking Matroska v2 or 0x53B9. The older values [20] were 0: mono, 1: right eye, 2: left eye, 3: both eyes.

[25.](#) Timecodes

[25.1.](#) Timecode Types

- o Absolute Timecode = Block+Cluster
- o Relative Timecode = Block
- o Scaled Timecode = Block+Cluster
- o Raw Timecode = (Block+Cluster)*TimecodeScale*TrackTimecodeScale

[25.2.](#) Block Timecodes

The "Block Element"'s timecode MUST be a signed integer that represents the "Raw Timecode" relative to the "Cluster"'s "Timecode Element", multiplied by the "TimecodeScale Element". See [Section 25.4](#) for more information.

The "Block Element"'s timecode MUST be represented by a 16bit signed integer (sint16). The "Block"'s timecode has a range of -32768 to +32767 units. When using the default value of the "TimecodeScale Element", each integer represents 1ms. The maximum time span of "Block Elements" in a "Cluster" using the default "TimecodeScale Element" of 1ms is 65536ms.

If a "Cluster"'s "Timecode Element" is set to zero, it is possible to have "Block Elements" with a negative "Raw Timecode". "Block Elements" with a negative "Raw Timecode" are not valid.

[25.3.](#) Raw Timecode

The exact time of an object SHOULD be represented in nanoseconds. To find out a "Block"'s "Raw Timecode", you need the "Block"'s "Timecode Element", the "Cluster"'s "Timecode Element", and the "TimecodeScale Element".

[25.4.](#) TimecodeScale

The "TimecodeScale Element" is used to calculate the "Raw Timecode" of a "Block". The timecode is obtained by adding the "Block"'s timecode to the "Cluster"'s "Timecode Element", and then multiplying that result by the "TimecodeScale". The result will be the "Block"'s "Raw Timecode" in nanoseconds. The formula for this would look like:

$$(a + b) * c$$

```
a = `Block`'s Timecode  
b = `Cluster`'s Timecode  
c = `TimeCodeScale`
```

For example, assume a "Cluster"'s "Timecode" has a value of 564264, the "Block" has a "Timecode" of 1233, and the "TimecodeScale Element" is the default of 1000000.

$$(1233 + 564264) * 1000000 = 565497000000$$

So, the "Block" in this example has a specific time of 565497000000 in nanoseconds. In milliseconds this would be 565497ms.

[25.5.](#) TimecodeScale Rounding

Because the default value of "TimecodeScale" is 1000000, which makes each integer in the "Cluster" and "Block" "Timecode Elements" equal 1ms, this is the most commonly used. When dealing with audio, this causes inaccuracy when seeking. When the audio is combined with video, this is not an issue. For most cases, the the synch of audio to video does not need to be more than 1ms accurate. This becomes obvious when one considers that sound will take 2-3ms to travel a single meter, so distance from your speakers will have a greater effect on audio/visual synch than this.

However, when dealing with audio-only files, seeking accuracy can become critical. For instance, when storing a whole CD in a single track, a user will want to be able to seek to the exact sample that a song begins at. If seeking a few sample ahead or behind, a 'crack' or 'pop' may result as a few odd samples are rendered. Also, when performing precise editing, it may be very useful to have the audio accuracy down to a single sample.

When storing timecodes for an audio stream, the "TimecodeScale Element" SHOULD have an accuracy of at least that of the audio sample rate, otherwise there are rounding errors that prevent users from knowing the precise location of a sample. Here's how a program has to round each timecode in order to be able to recreate the sample number accurately.

Let's assume that the application has an audio track with a sample rate of 44100. As written above the "TimecodeScale" MUST have at least the accuracy of the sample rate itself: $1000000000 / 44100 = 22675.7369614512$. This value MUST always be truncated. Otherwise the accuracy will not suffice. So in this example the application will use 22675 for the "TimecodeScale". The application could even

use some lower value like 22674 which would allow it to be a little bit imprecise about the original timecodes. But more about that in a minute.

Next the application wants to write sample number 52340 and calculates the timecode. This is easy. In order to calculate the "Raw Timecode" in ns all it has to do is calculate "Raw Timecode = round(1000000000 * sample_number / sample_rate)". Rounding at this stage is very important! The application might skip it if it choses a slightly smaller value for the "TimecodeScale" factor instead of the truncated one like shown above. Otherwise it has to round or the results won't be reversible. For our example we get "Raw Timecode = round(1000000000 * 52340 / 44100) = round(1186848072.56236) = 1186848073".

The next step is to calculate the "Absolute Timecode" - that is the timecode that will be stored in the Matroska file. Here the application has to divide the "Raw Timecode" from the previous paragraph by the "TimecodeScale" factor and round the result: "Absolute Timecode = round(Raw Timecode / TimecodeScale_factor)" which will result in the following for our example: "Absolute Timecode = round(1186848073 / 22675) = round(52341.7011245866) = 52342". This number is the one the application has to write to the file.

Now our file is complete, and we want to play it back with another application. Its task is to find out which sample the first application wrote into the file. So it starts reading the Matroska file and finds the "TimecodeScale" factor 22675 and the audio sample rate 44100. Later it finds a data block with the "Absolute Timecode" of 52342. But how does it get the sample number from these numbers?

First it has to calculate the "Raw Timecode" of the block it has just read. Here's no rounding involved, just an integer multiplication: "Raw Timecode = Absolute Timecode * TimecodeScale_factor". In our example: "Raw Timecode = 52342 * 22675 = 1186854850".

The conversion from the "Raw Timecode" to the sample number again requires rounding: "sample_number = round(Raw Timecode * sample_rate / 1000000000)". In our example: "sample_number = round(1186854850 * 44100 / 1000000000) = round(52340.298885) = 52340". This is exactly the sample number that the previous program started with.

Some general notes for a program:

1. Always calculate the timestamps / sample numbers with floating point numbers of at least 64bit precision (called 'double' in

most modern programming languages). If you're calculating with integers then make sure they're 64bit long, too.

2. Always round if you divide. Always! If you don't you'll end up with situations in which you have a timecode in the Matroska file that does not correspond to the sample number that it started with. Using a slightly lower timecode scale factor can help here in that it removes the need for proper rounding in the conversion from sample number to "Raw Timecode".

[25.6.](#) **TrackTimecodeScale**

The "TrackTimecodeScale Element" is used align tracks that would otherwise be played at different speeds. An example of this would be if you have a film that was originally recorded at 24fps video. When playing this back through a PAL broadcasting system, it is standard to speed up the film to 25fps to match the 25fps display speed of the PAL broadcasting standard. However, when broadcasting the video through NTSC, it is typical to leave the film at its original speed. If you wanted to make a single file where there was one video stream, and an audio stream used from the PAL broadcast, as well as an audio stream used from the NTSC broadcast, you would have the problem that the PAL audio stream would be 1/24th faster than the NTSC audio stream, quickly leading to problems. It is possible to stretch out the PAL audio track and re-encode it at a slower speed, however when dealing with lossy audio codecs, this often results in a loss of audio quality and/or larger file sizes.

This is the type of problem that "TrackTimecodeScale" was designed to fix. Using it, the video can be played back at a speed that will synch with either the NTSC or the PAL audio stream, depending on which is being used for playback. To continue the above example:

```
Track 1: Video
Track 2: NTSC Audio
Track 3: PAL Audio
```

Because the NTSC track is at the original speed, it will be used as the default value of 1.0 for its "TrackTimecodeScale". The video will also be aligned to the NTSC track with the default value of 1.0.

The "TrackTimecodeScale" value to use for the PAL track would be calculated by determining how much faster the PAL track is than the NTSC track. In this case, because we know the video for the NTSC audio is being played back at 24fps and the video for the PAL audio is being played back at 25fps, the calculation would be:

25/24 ≈ 1.0416666666666667

When writing a file that uses a non-default "TrackTimecodeScale", the values of the "Block"'s timecode are whatever they would be when normally storing the track with a default value for the "TrackTimecodeScale". However, the data is interleaved a little differently. Data SHOULD be interleaved by its [Section 25.3](#) in the order handed back from the encoder. The "Raw Timecode" of a "Block" from a track using "TrackTimecodeScale" is calculated using:

$$\text{"(Block's Timecode + Cluster's Timecode) * TimecodeScale * TrackTimecodeScale"}$$

So, a Block from the PAL track above that had a [Section 25.1](#) of 100 seconds would have a "Raw Timecode" of 104.66666667 seconds, and so would be stored in that part of the file.

When playing back a track using the "TrackTimecodeScale", if the track is being played by itself, there is no need to scale it. From the above example, when playing the Video with the NTSC Audio, neither are scaled. However, when playing back the Video with the PAL Audio, the timecodes from the PAL Audio track are scaled using the "TrackTimecodeScale", resulting in the video playing back in synch with the audio.

It would be possible for a "Matroska Player" to also adjust the audio's samplerate at the same time as adjusting the timecodes if you wanted to play the two audio streams synchronously. It would also be possible to adjust the video to match the audio's speed. However, for playback, the selected track(s) timecodes SHOULD be adjusted if they need to be scaled.

While the above example deals specifically with audio tracks, this element can be used to align video, audio, subtitles, or any other type of track contained in a Matroska file.

[26. References](#)

[26.1. URIs](#)

- [1] <http://mukoli.free.fr/mcf/mcf.html>
- [2] <https://github.com/Matroska-Org/ebml-specification/blob/master/specification.markdown>
- [3] <https://datatracker.ietf.org/wg/cellar/charter/>
- [4] <https://matroska.org/files/matroska.pdf>
- [5] diagram.md

- [6] <https://github.com/Matroska-Org/ebml-specification/blob/master/specification.markdown>
- [7] <https://github.com/Matroska-Org/foundation-source/blob/master/spectool/specdata.xml>
- [8] <https://tools.ietf.org/html/rfc2119>
- [9] <https://github.com/Matroska-Org/ebml-specification/blob/master/specification.markdown>
- [10] <https://github.com/Matroska-Org/ebml-specification/blob/master/specification.markdown#ebml-element-types>
- [11] <https://github.com/Matroska-Org/ebml-specification/blob/master/specification.markdown#ebml-schema>
- [12] <https://github.com/Matroska-Org/ebml-specification/blob/master/specification.markdown#structure>
- [13] https://www.loc.gov/standards/iso639-2/php/English_list.php
- [14] <https://tools.ietf.org/html/bcp47>
- [15] <https://www.iana.org/domains/root/db>
- [16] <http://www.webmproject.org/docs/container/>
- [17] <http://dvd.sourceforge.net/dvdinfo/uops.html>
- [18] <http://www.dvd-replica.com/DVD/>
- [19] <http://dvd.sourceforge.net/dvdinfo/>
- [20] <http://www.matroska.org/node/1/revisions/74/view#StereoMode>

Authors' Addresses

Steve Lhomme

Email: slhomme@matroska.org

Moritz Bunkus

Email: moritz@bunkus.org

Dave Rice

Email: dave@dericed.com