Workgroup: cellar
Internet-Draft: draft-ietf-cellar-matroska-10
Published: 1 May 2022
Intended Status: Standards Track
Expires: 2 November 2022
Authors: S. Lhomme M. Bunkus D. Rice

Matroska Media Container Format Specifications

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 2 November 2022.

Table of Contents

1. Introduction

Matroska aims to become THE standard of multimedia container formats. It was derived from a project called [MCF], but differentiates from it significantly because it is based on EBML (Extensible Binary Meta Language) [RFC8794], 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:

*It is NOT a video or audio compression format (codec)

* 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:

* Fast seeking in the file
* Chapter entries
* Full metadata (tags) support
* Selectable subtitle/audio/video streams
* Modularly expandable
* Error resilience (can recover playback even when the stream is damaged)
* Streamable over the internet and local networks (HTTP, CIFS, FTP, etc)
* 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. But since it's quite complete it is used as a reference for the development of libmatroska.

This document covers Matroska versions 1, 2, 3 and 4. Matroska v4 is the current version. Matroska 1 to 3 are no longer maintained. No new elements are expected in files with theses version numbers. There **MAY** be further additions to Matroska v4.

3. Notation and Conventions

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

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 data parser that interprets the semantics of a Matroska document and creates a way for programs to use Matroska.

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

4. Matroska Overview

4.1. Principles

Matroska is a Document Type of EBML (Extensible Binary Meta Language). This specification is dependent on the EBML Specification [RFC8794]. For an understanding of Matroska's EBML Schema, see in particular the sections of the EBML Specification covering EBML Element Types (Section 7), EBML Schema (Section 11.1), and EBML Structure (Section 3).

4.2. Added EBML Constraints

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

*The docType of the EBML Header **MUST** be "matroska".

*The EBMLMaxIDLength of the EBML Header **MUST** be "4".

*The EBMLMaxSizeLength of the EBML Header **MUST** be between "1" and "8" inclusive.

4.3. Design Rules

The Root Element and all Top-Levels Elements use 4 octets for their EBML Element ID -- i.e. Segment and direct children of Segment.

Legacy EBML/Matroska parsers did not handle Empty Elements properly, elements present in the file but with a length of zero. They always assumed the value was 0 for integers/dates and 0x0p+0 for floats, no matter the default value of the element which should have been used instead. Therefore Matroska writers **MUST NOT** use EBML Empty Elements, if the element has a default value that is not 0 for integers/dates and 0x0p+0 for floats.

When adding new elements to Matroska, these rules **MUST** be followed:

- A non-mandatory integer/date Element **MUST NOT** have a default value other than 0.
- A non-mandatory float Element **MUST NOT** have a default value other than 0x0p+0.
- A non-mandatory string Element **MUST NOT** have a default value, as empty string cannot be defined in the XML Schema.

4.4. Data Layout

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:

- EBML Header
- Segment

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

- EBML Header
- Segment
- EBML Header
- 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.

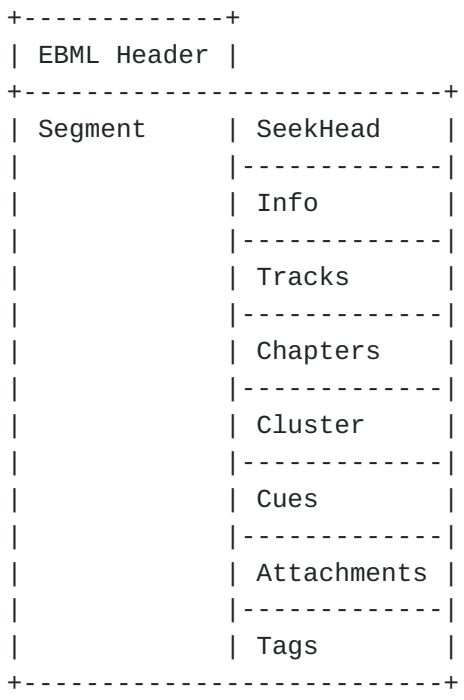

Figure 1: Basic layout of a Matroska file.

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.

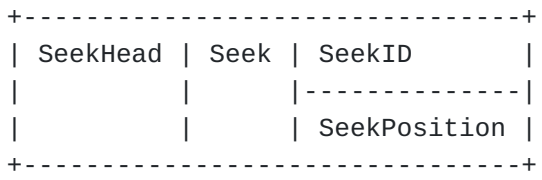

Figure 2: Representation of a SeekHead Element.

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.

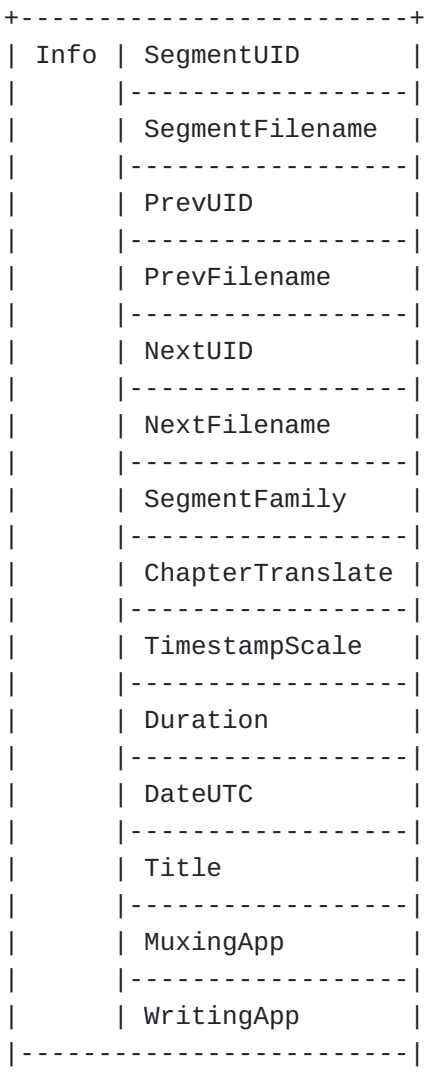

Figure 3: Representation of an Info Element and its Child Elements.

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.

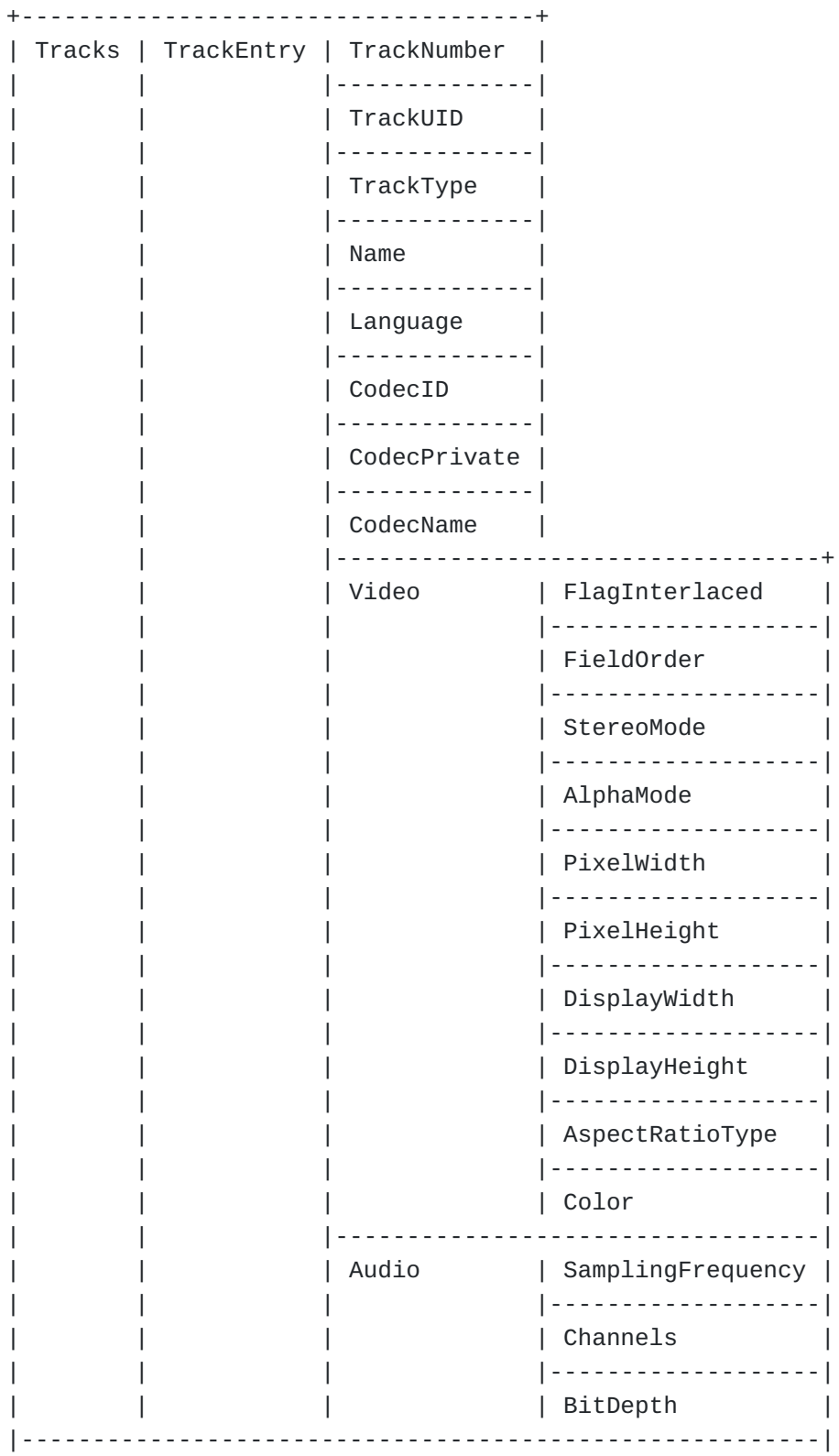

Figure 4: Representation of the Tracks Element and a selection of its Descendant Elements.

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

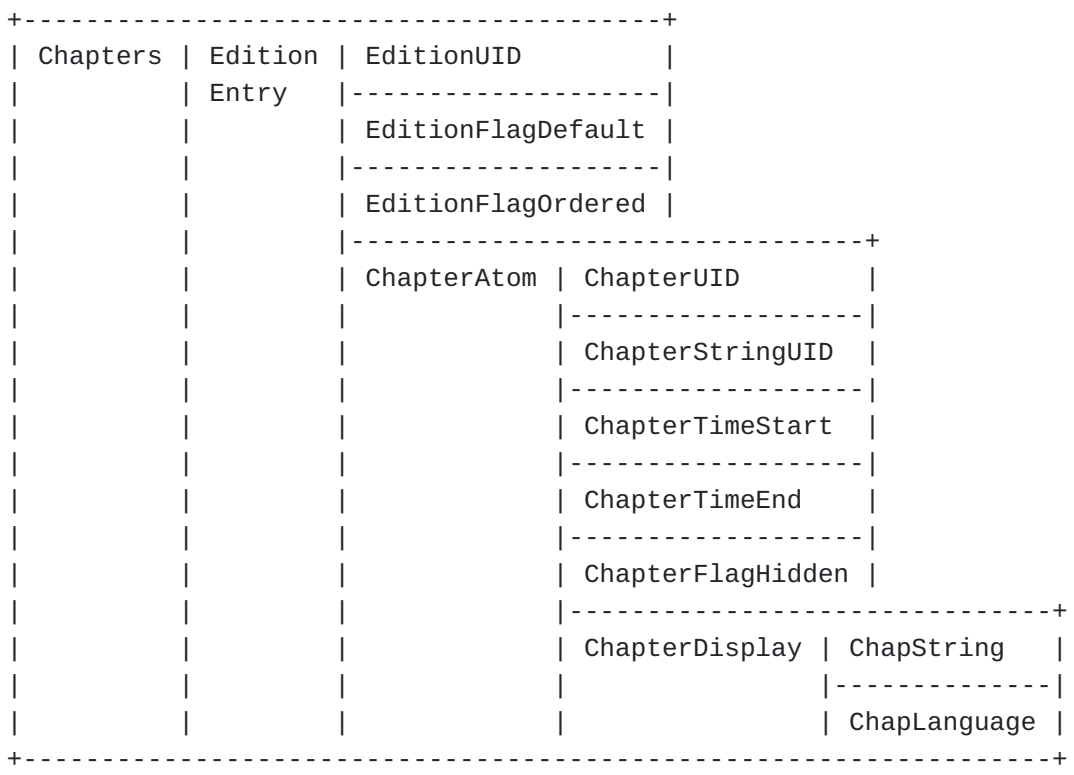

Figure 5:
Representation of the Chapters Element and a selection of its Descendant Elements.

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 Timestamp Element. This **SHOULD** be the Timestamp 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.

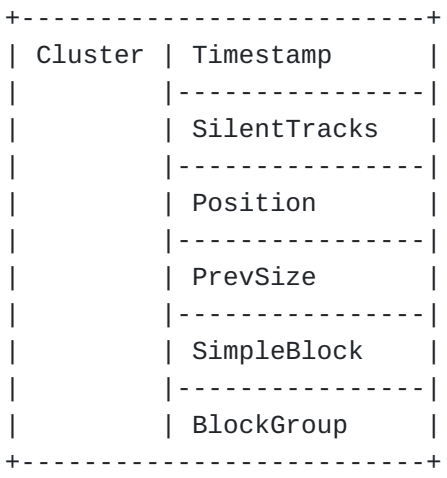

Figure 6: Representation of a Cluster Element and its immediate Child Elements.

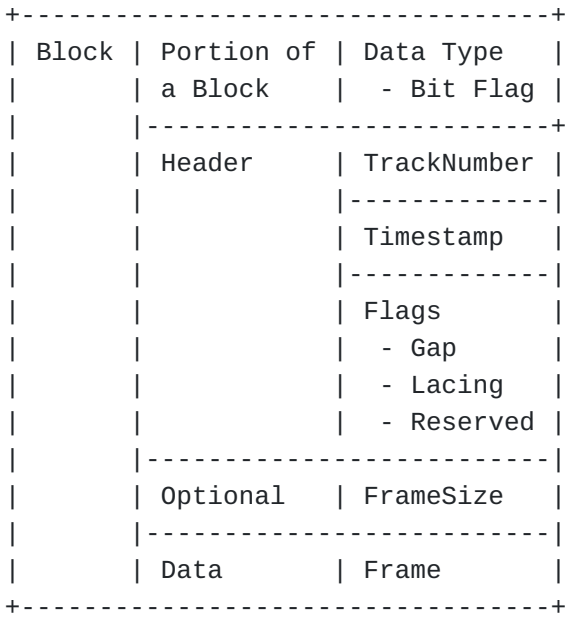

Figure 7: Representation of the Block Element structure.

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

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

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 timestamp 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 timestamp 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.

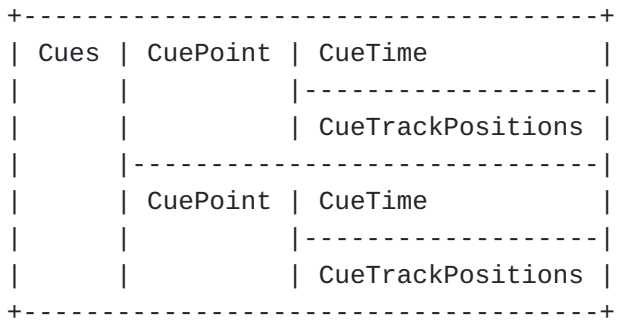

Figure 8: Representation of a Cues Element and two levels of its Descendant Elements.

The Attachments Element is for attaching files to a Matroska file such as pictures, fonts, webpages, etc.

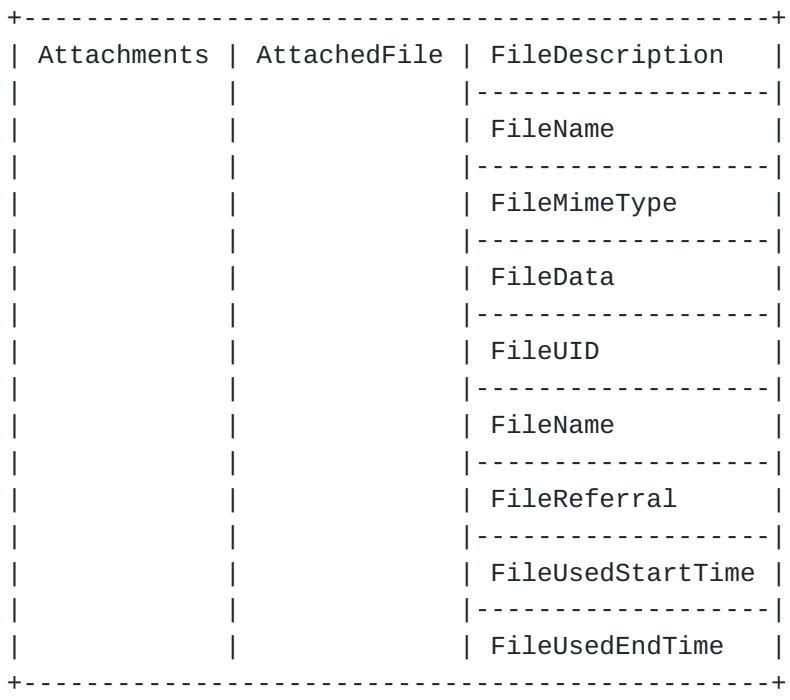

Figure 9: Representation of a Attachments Element.

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.

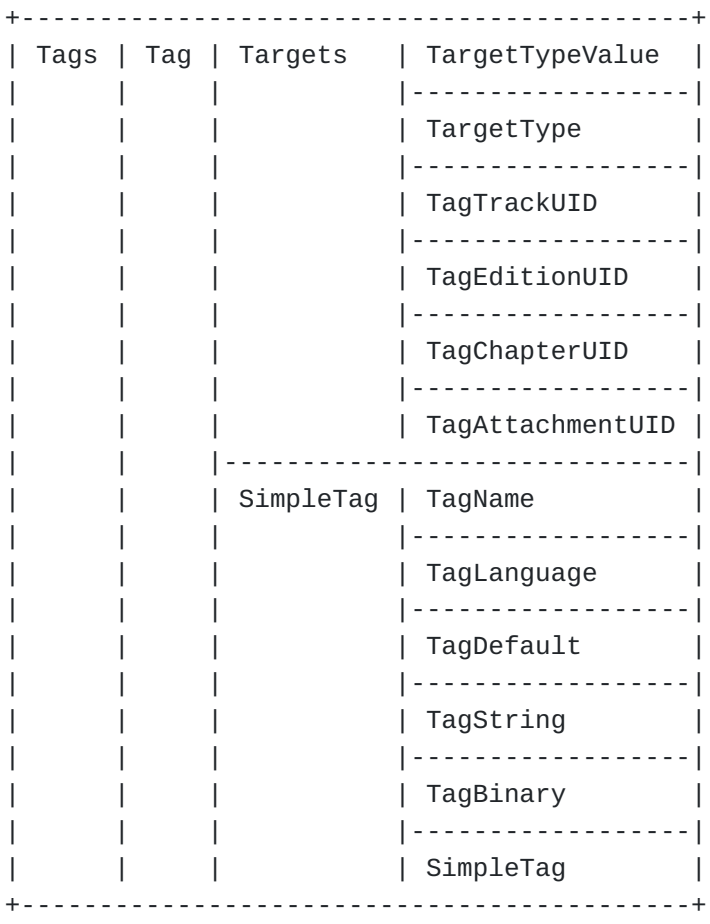

Figure 10: Representation of a Tags Element and three levels of its Children Elements.

5. 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.

Here the definition of each Matroska Element is provided.

5.1. Segment Element

name: Segment

path: \Segment

id: 0x18538067

minOccurs: 1

maxOccurs: 1

type:
master

unknownsizeallowed: 1

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

5.1.1. SeekHead Element

name: SeekHead

path: \Segment\SeekHead

id: 0x114D9B74

maxOccurs: 2

type: master

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

5.1.1.1. Seek Element

name: Seek

path: \Segment\SeekHead\Seek

id: 0x4DBB

minOccurs: 1

type: master

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

5.1.1.1.1. SeekID Element

name: SeekID

path: \Segment\SeekHead\Seek\SeekID

id: 0x53AB

minOccurs: 1

maxOccurs: 1

type: binary

definition:
The binary ID corresponding to the Element name.

5.1.1.1.2. SeekPosition Element

name: SeekPosition

path: \Segment\SeekHead\Seek\SeekPosition

id: 0x53AC

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: The Segment Position of the Element.

5.1.2. Info Element

name: Info

path: \Segment\Info

id: 0x1549A966

minOccurs: 1

maxOccurs: 1

type: master

recurring: 1

definition: Contains general information about the Segment.

5.1.2.1. SegmentUID Element

name: SegmentUID

path: \Segment\Info\SegmentUID

id: 0x73A4

maxOccurs: 1

range: not 0

type: binary

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**.

5.1.2.2. SegmentFilename Element

name: SegmentFilename

path: \Segment\Info\SegmentFilename

id: 0x7384

maxOccurs: 1

type: utf-8

definition: A filename corresponding to this Segment.

5.1.2.3. PrevUID Element

name: PrevUID

path: \Segment\Info\PrevUID

id: 0x3CB923

maxOccurs: 1

type: binary

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.

5.1.2.4. PrevFilename Element

name: PrevFilename

path: \Segment\Info\PrevFilename

id: 0x3C83AB

maxOccurs: 1

type:
utf-8

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.

5.1.2.5. NextUID Element

name: NextUID

path: \Segment\Info\NextUID

id: 0x3EB923

maxOccurs: 1

type: binary

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.

5.1.2.6. NextFilename Element

name: NextFilename

path: \Segment\Info\NextFilename

id: 0x3E83BB

maxOccurs: 1

type: utf-8

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.

5.1.2.7. SegmentFamily Element

name:
SegmentFamily

path: \Segment\Info\SegmentFamily

id: 0x4444

type: binary

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

usage notes: If the Segment Info contains a ChapterTranslate element, this Element is **REQUIRED**.

5.1.2.8. ChapterTranslate Element

name: ChapterTranslate

path: \Segment\Info\ChapterTranslate

id: 0x6924

type: master

definition: The mapping between this Segment and a segment value in the given Chapter Codec.

rationale: Chapter Codec may need to address different segments, but they may not know of the way to identify such segment when stored in Matroska. This element and its child elements add a way to map the internal segments known to the Chapter Codec to the Segment IDs in Matroska. This allows remuxing a file with Chapter Codec without changing the content of the codec data, just the Segment mapping.

5.1.2.8.1. ChapterTranslateID Element

name: ChapterTranslateID

path: \Segment\Info\ChapterTranslate\ChapterTranslateID

id: 0x69A5

minOccurs: 1

maxOccurs: 1

type: binary

definition: The binary value used to represent this Segment in the chapter codec data. The format depends on the ChapProcessCodecID used; see Section 5.1.7.1.4.15.

5.1.2.8.2. ChapterTranslateCodec Element

name: ChapterTranslateCodec

path: \Segment\Info\ChapterTranslate\ChapterTranslateCodec

id: 0x69BF

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: This ChapterTranslate applies to this chapter codec of the given chapter edition(s); see Section 5.1.7.1.4.15.

defined values:

value	label	definition
0	Matroska Script	Chapter commands using the Matroska Script codec.
1	DVD-menu	Chapter commands using the DVD-like codec.

Table 1: ChapterTranslateCodec values

5.1.2.8.3. ChapterTranslateEditionUID Element

name: ChapterTranslateEditionUID

path: \Segment\Info\ChapterTranslate\ChapterTranslateEditionUID

id: 0x69FC

type:
uinteger

definition: Specify a chapter edition UID on which this ChapterTranslate applies.

usage notes: When no ChapterTranslateEditionUID is specified in the ChapterTranslate, the ChapterTranslate applied to all chapter editions found in the Segment using the given ChapterTranslateCodec.

5.1.2.9. TimestampScale Element

name: TimestampScale

path: \Segment\Info\TimestampScale

id: 0x2AD7B1

minOccurs: 1

maxOccurs: 1

range: not 0

default: 1000000

type: uinteger

definition: Base unit for Segment Ticks and Track Ticks, in nanoseconds. A TimestampScale value of 1.000.000 means scaled timestamps in the Segment are expressed in milliseconds; see Section 10 on how to interpret timestamps.

5.1.2.10. Duration Element

name: Duration

path: \Segment\Info\Duration

id: 0x4489

maxOccurs: 1

range: > 0x0p+0

type: float

definition: Duration of the Segment, expressed in Segment Ticks which is based on TimestampScale; see Section 10.1.

5.1.2.11. DateUTC Element

name: DateUTC

path: \Segment\Info\DateUTC

id: 0x4461

maxOccurs: 1

type: date

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

5.1.2.12. Title Element

name: Title

path: \Segment\Info\Title

id: 0x7BA9

maxOccurs: 1

type: utf-8

definition: General name of the Segment.

5.1.2.13. MuxingApp Element

name: MuxingApp

path: \Segment\Info\MuxingApp

id: 0x4D80

minOccurs: 1

maxOccurs: 1

type: utf-8

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.

5.1.2.14. WritingApp Element

name:
WritingApp

path: \Segment\Info\WritingApp

id: 0x5741

minOccurs: 1

maxOccurs: 1

type: utf-8

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

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

5.1.3. Cluster Element

name: Cluster

path: \Segment\Cluster

id: 0x1F43B675

type: master

unknownsizeallowed: 1

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

5.1.3.1. Timestamp Element

name: Timestamp

path: \Segment\Cluster\Timestamp

id: 0xE7

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: Absolute timestamp of the cluster, expressed in Segment Ticks which is based on TimestampScale; see Section 10.1.

usage notes: This element **SHOULD** be the first child element of the Cluster it belongs to, or the second if that Cluster contains a CRC-32 element (Section 6.2).

5.1.3.2. Position Element

name: Position

path: \Segment\Cluster\Position

id: 0xA7

maxOccurs: 1

type: uinteger

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

5.1.3.3. PrevSize Element

name: PrevSize

path: \Segment\Cluster\PrevSize

id: 0xAB

maxOccurs: 1

type: uinteger

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

5.1.3.4. SimpleBlock Element

name: SimpleBlock

path: \Segment\Cluster\SimpleBlock

id: 0xA3

type: binary

minver: 2

definition: Similar to Block, see Section 9, but without all the extra information, mostly used to reduced overhead when no extra feature is needed; see Section 9.3 on SimpleBlock Structure.

5.1.3.5. BlockGroup Element

name: BlockGroup

path: \Segment\Cluster\BlockGroup

id: 0xA0

type: master

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

5.1.3.5.1. Block Element

name: Block

path: \Segment\Cluster\BlockGroup\Block

id: 0xA1

minOccurs: 1

maxOccurs: 1

type: binary

definition: Block containing the actual data to be rendered and a timestamp relative to the Cluster Timestamp; see Section 9 on Block Structure.

5.1.3.5.2. BlockAdditions Element

name: BlockAdditions

path:
\Segment\Cluster\BlockGroup\BlockAdditions

id: 0x75A1

maxOccurs: 1

type: master

definition: 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.

5.1.3.5.2.1. BlockMore Element

name: BlockMore

path: \Segment\Cluster\BlockGroup\BlockAdditions\BlockMore

id: 0xA6

minOccurs: 1

type: master

definition: Contain the BlockAdditional and some parameters.

5.1.3.5.2.2. BlockAddID Element

name: BlockAddID

path:
\Segment\Cluster\BlockGroup\BlockAdditions\BlockMore\BlockAddID

id: 0xEE

minOccurs: 1

maxOccurs: 1

range: not 0

default: 1

type: uinteger

definition: An ID to identify the BlockAdditional level. If BlockAddIDType of the corresponding block is 0, this value is also the value of BlockAddIDType for the meaning of the content of BlockAdditional.

5.1.3.5.2.3. BlockAdditional Element

name: BlockAdditional

path: \Segment\Cluster\BlockGroup\BlockAdditions\BlockMore\BlockAdditional

id: 0xA5

minOccurs: 1

maxOccurs: 1

type: binary

definition: Interpreted by the codec as it wishes (using the BlockAddID).

5.1.3.5.3. BlockDuration Element

name: BlockDuration

path: \Segment\Cluster\BlockGroup\BlockDuration

id: 0x9B

minOccurs: see implementation notes

maxOccurs: 1

default: see implementation notes

type: uinteger

definition: The duration of the Block, expressed in Track Ticks; see Section 10.1. The BlockDuration Element can be useful at the end of a Track to define the duration of the last frame (as there is no subsequent Block available), or when there is a break in a track like for subtitle tracks.

notes:

attribute	note
minOccurs	BlockDuration **MUST** be set (minOccurs=1) if the associated TrackEntry stores a DefaultDuration value.
default	When not written and with no DefaultDuration, the value is assumed to be the difference between the timestampof this

attribute	note
	Block and the timestamp of the next Block in "display" order (not coding order).

Table 2: BlockDuration implementation notes

5.1.3.5.4. ReferencePriority Element

name: ReferencePriority

path: \Segment\Cluster\BlockGroup\ReferencePriority

id: 0xFA

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: 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.

5.1.3.5.5. ReferenceBlock Element

name: ReferenceBlock

path: \Segment\Cluster\BlockGroup\ReferenceBlock

id: 0xFB

type: integer

definition: A timestamp value, relative to the timestamp of the Block in this BlockGroup, expressed in Track Ticks; see Section 10.1. This is used to reference other frames necessary to decode this frame. The relative value **SHOULD** correspond to a valid Block this Block depends on. Historically Matroska Writer didn't write the actual Block(s) this Block depends on, but *some* Block in the past.

The value "0" **MAY** also be used to signify this Block cannot be decoded on its own, but without knownledge of which Block is necessary. In this case, other ReferenceBlock **MUST NOT** be found in the same BlockGroup.

If the BlockGroup doesn't have any ReferenceBlock element, then the Block it contains can be decoded without using any other Block data.

5.1.3.5.6. CodecState Element

name: CodecState

path: \Segment\Cluster\BlockGroup\CodecState

id: 0xA4

maxOccurs: 1

type: binary

minver: 2

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

5.1.3.5.7. DiscardPadding Element

name: DiscardPadding

path: \Segment\Cluster\BlockGroup\DiscardPadding

id: 0x75A2

maxOccurs: 1

type: integer

minver: 4

definition: Duration of the silent data added to the Block, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1 (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.

5.1.4. Tracks Element

name: Tracks

path: \Segment\Tracks

id: 0x1654AE6B

maxOccurs: 1

type: master

recurring:
1

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

5.1.4.1. TrackEntry Element

name: TrackEntry

path: \Segment\Tracks\TrackEntry

id: 0xAE

minOccurs: 1

type: master

definition: Describes a track with all Elements.

5.1.4.1.1. TrackNumber Element

name: TrackNumber

path: \Segment\Tracks\TrackEntry\TrackNumber

id: 0xD7

minOccurs: 1

maxOccurs: 1

range: not 0

type: uinteger

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

5.1.4.1.2. TrackUID Element

name: TrackUID

path: \Segment\Tracks\TrackEntry\TrackUID

id: 0x73C5

minOccurs: 1

maxOccurs: 1

range:
not 0

type: uinteger

definition: A unique ID to identify the Track.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.3. TrackType Element

name: TrackType

path: \Segment\Tracks\TrackEntry\TrackType

id: 0x83

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: The TrackType defines the type of each frame found in the Track. The value **SHOULD** be stored on 1 octet.

defined values:

value	label	each frame contains
1	video	An image.
2	audio	Audio samples.
3	complex	A mix of different other TrackType. The codec needs to define how the Matroska Player should interpret such data.
16	logo	An image to be rendered over the video track(s).
17	subtitle	Subtitle or closed caption data to be rendered over the video track(s).
18	buttons	Interactive button(s) to be rendered over the video track(s).
32	control	Metadata used to control the player of the Matroska Player.
33	metadata	Timed metadata that can be passed on to the Matroska Player.

Table 3: TrackType values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.4. FlagEnabled Element

name: FlagEnabled

path: \Segment\Tracks\TrackEntry\FlagEnabled

id: 0xB9

minOccurs: 1

maxOccurs: 1

range: 0-1

default: 1

type: uinteger

minver: 2

definition: Set to 1 if the track is usable. It is possible to turn a not usable track into a usable track using chapter codecs or control tracks.

5.1.4.1.5. FlagDefault Element

name: FlagDefault

path: \Segment\Tracks\TrackEntry\FlagDefault

id: 0x88

minOccurs: 1

maxOccurs: 1

range: 0-1

default: 1

type: uinteger

definition: Set if that track (audio, video or subs) **SHOULD** be eligible for automatic selection by the player; see Section 19 for more details.

5.1.4.1.6. FlagForced Element

name: FlagForced

path: \Segment\Tracks\TrackEntry\FlagForced

id:
0x55AA

minOccurs: 1

maxOccurs: 1

range: 0-1

default: 0

type: uinteger

definition: Applies only to subtitles. Set if that track **SHOULD** be eligible for automatic selection by the player if it matches the user's language preference, even if the user's preferences would normally not enable subtitles with the selected audio track; this can be used for tracks containing only translations of foreign-language audio or onscreen text. See Section 19 for more details.

5.1.4.1.7. FlagHearingImpaired Element

name: FlagHearingImpaired

path: \Segment\Tracks\TrackEntry\FlagHearingImpaired

id: 0x55AB

maxOccurs: 1

range: 0-1

type: uinteger

minver: 4

definition: Set to 1 if that track is suitable for users with hearing impairments, set to 0 if it is unsuitable for users with hearing impairments.

5.1.4.1.8. FlagVisualImpaired Element

name: FlagVisualImpaired

path: \Segment\Tracks\TrackEntry\FlagVisualImpaired

id: 0x55AC

maxOccurs: 1

range: 0-1

type:
uinteger

minver: 4

definition: Set to 1 if that track is suitable for users with visual impairments, set to 0 if it is unsuitable for users with visual impairments.

5.1.4.1.9. FlagTextDescriptions Element

name: FlagTextDescriptions

path: \Segment\Tracks\TrackEntry\FlagTextDescriptions

id: 0x55AD

maxOccurs: 1

range: 0-1

type: uinteger

minver: 4

definition: Set to 1 if that track contains textual descriptions of video content, set to 0 if that track does not contain textual descriptions of video content.

5.1.4.1.10. FlagOriginal Element

name: FlagOriginal

path: \Segment\Tracks\TrackEntry\FlagOriginal

id: 0x55AE

maxOccurs: 1

range: 0-1

type: uinteger

minver: 4

definition: Set to 1 if that track is in the content's original language, set to 0 if it is a translation.

5.1.4.1.11. FlagCommentary Element

name: FlagCommentary

path:
\Segment\Tracks\TrackEntry\FlagCommentary

id: 0x55AF

maxOccurs: 1

range: 0-1

type: uinteger

minver: 4

definition: Set to 1 if that track contains commentary, set to 0 if it does not contain commentary.

5.1.4.1.12. FlagLacing Element

name: FlagLacing

path: \Segment\Tracks\TrackEntry\FlagLacing

id: 0x9C

minOccurs: 1

maxOccurs: 1

range: 0-1

default: 1

type: uinteger

definition: Set to 1 if the track **MAY** contain blocks using lacing. When set to 0 all blocks **MUST** have their lacing flags set to No lacing; see Section 9.4 on Block Lacing.

5.1.4.1.13. MinCache Element

name: MinCache

path: \Segment\Tracks\TrackEntry\MinCache

id: 0x6DE7

minOccurs: 1

maxOccurs: 1

default: 0

type:
uinteger

definition: 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.

5.1.4.1.14. MaxCache Element

name: MaxCache

path: \Segment\Tracks\TrackEntry\MaxCache

id: 0x6DF8

maxOccurs: 1

type: uinteger

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

5.1.4.1.15. DefaultDuration Element

name: DefaultDuration

path: \Segment\Tracks\TrackEntry\DefaultDuration

id: 0x23E383

maxOccurs: 1

range: not 0

type: uinteger

definition: Number of nanoseconds per frame, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1 (frame in the Matroska sense -- one Element put into a (Simple)Block).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.16. DefaultDecodedFieldDuration Element

name: DefaultDecodedFieldDuration

path: \Segment\Tracks\TrackEntry\DefaultDecodedFieldDuration

id: 0x234E7A

maxOccurs:
1

range: not 0

type: uinteger

minver: 4

definition: The period between two successive fields at the output of the decoding process, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1. see Section 8 for more information

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.17. TrackTimestampScale Element

name: TrackTimestampScale

path: \Segment\Tracks\TrackEntry\TrackTimestampScale

id: 0x23314F

minOccurs: 1

maxOccurs: 1

range: > 0x0p+0

default: 0x1p+0

type: float

maxver: 3

definition: 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).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.18. MaxBlockAdditionID Element

name: MaxBlockAdditionID

path: \Segment\Tracks\TrackEntry\MaxBlockAdditionID

id: 0x55EE

minOccurs:
1

maxOccurs: 1

default: 0

type: uinteger

definition: The maximum value of BlockAddID (Section 5.1.3.5.2.2). A value 0 means there is no BlockAdditions (Section 5.1.3.5.2) for this track.

5.1.4.1.19. BlockAdditionMapping Element

name: BlockAdditionMapping

path: \Segment\Tracks\TrackEntry\BlockAdditionMapping

id: 0x41E4

type: master

minver: 4

definition: Contains elements that extend the track format, by adding content either to each frame, with BlockAddID (Section 5.1.3.5.2.2), or to the track as a whole with BlockAddIDExtraData.

5.1.4.1.19.1. BlockAddIDValue Element

name: BlockAddIDValue

path:
\Segment\Tracks\TrackEntry\BlockAdditionMapping\BlockAddIDValue

id: 0x41F0

maxOccurs: 1

range: >=2

type: uinteger

minver: 4

definition: If the track format extension needs content beside frames, the value refers to the BlockAddID (Section 5.1.3.5.2.2), value being described. To keep MaxBlockAdditionID as low as possible, small values **SHOULD** be used.

5.1.4.1.19.2. BlockAddIDName Element

name: BlockAddIDName

path:
\Segment\Tracks\TrackEntry\BlockAdditionMapping\BlockAddIDName

id: 0x41A4

maxOccurs: 1

type: string

minver: 4

definition: A human-friendly name describing the type of BlockAdditional data, as defined by the associated Block Additional Mapping.

5.1.4.1.19.3. BlockAddIDType Element

name: BlockAddIDType

path:
\Segment\Tracks\TrackEntry\BlockAdditionMapping\BlockAddIDType

id: 0x41E7

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 4

definition: Stores the registered identifier of the Block Additional Mapping to define how the BlockAdditional data should be handled.

5.1.4.1.19.4. BlockAddIDExtraData Element

name: BlockAddIDExtraData

path:
\Segment\Tracks\TrackEntry\BlockAdditionMapping\BlockAddIDExtraDa
ta

id: 0x41ED

maxOccurs:
1

type: binary

minver: 4

definition: Extra binary data that the BlockAddIDType can use to interpret the BlockAdditional data. The interpretation of the binary data depends on the BlockAddIDType value and the corresponding Block Additional Mapping.

5.1.4.1.20. Name Element

name: Name

path: \Segment\Tracks\TrackEntry\Name

id: 0x536E

maxOccurs: 1

type: utf-8

definition: A human-readable track name.

5.1.4.1.21. Language Element

name: Language

path: \Segment\Tracks\TrackEntry\Language

id: 0x22B59C

minOccurs: 1

maxOccurs: 1

default: eng

type: string

definition: Specifies the language of the track in the Matroska languages form; see Section 11 on language codes. This Element **MUST** be ignored if the LanguageIETF Element is used in the same TrackEntry.

5.1.4.1.22. LanguageIETF Element

name: LanguageIETF

path:
\Segment\Tracks\TrackEntry\LanguageIETF

id: 0x22B59D

maxOccurs: 1

type: string

minver: 4

definition: Specifies the language of the track according to [BCP47] and using the IANA Language Subtag Registry [IANALangRegistry]. If this Element is used, then any Language Elements used in the same TrackEntry **MUST** be ignored.

5.1.4.1.23. CodecID Element

name: CodecID

path: \Segment\Tracks\TrackEntry\CodecID

id: 0x86

minOccurs: 1

maxOccurs: 1

type: string

definition: An ID corresponding to the codec, see [MatroskaCodec] for more info.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.24. CodecPrivate Element

name: CodecPrivate

path: \Segment\Tracks\TrackEntry\CodecPrivate

id: 0x63A2

maxOccurs: 1

type: binary

definition: Private data only known to the codec.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.25. CodecName Element

name: CodecName

path: \Segment\Tracks\TrackEntry\CodecName

id: 0x258688

maxOccurs: 1

type: utf-8

definition: A human-readable string specifying the codec.

5.1.4.1.26. AttachmentLink Element

name: AttachmentLink

path: \Segment\Tracks\TrackEntry\AttachmentLink

id: 0x7446

maxOccurs: 1

range: not 0

type: uinteger

maxver: 3

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

usage notes: The value **MUST** match the FileUID value of an attachment found in this Segment.

5.1.4.1.27. TrackOverlay Element

name: TrackOverlay

path: \Segment\Tracks\TrackEntry\TrackOverlay

id: 0x6FAB

type: uinteger

definition: 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 Section 26.1 on 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.

5.1.4.1.28. CodecDelay Element

name: CodecDelay

path: \Segment\Tracks\TrackEntry\CodecDelay

id: 0x56AA

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 4

definition: CodecDelay is The codec-built-in delay, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1. It represents the amount of codec samples that will be discarded by the decoder during playback. This timestamp value **MUST** be subtracted from each frame timestamp in order to get the timestamp that will be actually played. The value **SHOULD** be small so the muxing of tracks with the same actual timestamp are in the same Cluster.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.29. SeekPreRoll Element

name: SeekPreRoll

path:
\Segment\Tracks\TrackEntry\SeekPreRoll

id: 0x56BB

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 4

definition: After a discontinuity, SeekPreRoll is the duration of the data the decoder **MUST** decode before the decoded data is valid, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.30. TrackTranslate Element

name: TrackTranslate

path: \Segment\Tracks\TrackEntry\TrackTranslate

id: 0x6624

type: master

definition: The mapping between this TrackEntry and a track value in the given Chapter Codec.

rationale: Chapter Codec may need to address content in specific track, but they may not know of the way to identify tracks in Matroska. This element and its child elements add a way to map the internal tracks known to the Chapter Codec to the track IDs in Matroska. This allows remuxing a file with Chapter Codec without changing the content of the codec data, just the track mapping.

5.1.4.1.30.1. TrackTranslateTrackID Element

name: TrackTranslateTrackID

path:
\Segment\Tracks\TrackEntry\TrackTranslate\TrackTranslateTrackID

id:
0x66A5

minOccurs: 1

maxOccurs: 1

type: binary

definition: The binary value used to represent this TrackEntry in the chapter codec data. The format depends on the ChapProcessCodecID used; see Section 5.1.7.1.4.15.

5.1.4.1.30.2. TrackTranslateCodec Element

name: TrackTranslateCodec

path: \Segment\Tracks\TrackEntry\TrackTranslate\TrackTranslateCodec

id: 0x66BF

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: This TrackTranslate applies to this chapter codec of the given chapter edition(s); see Section 5.1.7.1.4.15.

defined values:

value	label	definition
0	Matroska Script	Chapter commands using the Matroska Script codec.
1	DVD-menu	Chapter commands using the DVD-like codec.

Table 4: TrackTranslateCodec values

5.1.4.1.30.3. TrackTranslateEditionUID Element

name: TrackTranslateEditionUID

path: \Segment\Tracks\TrackEntry\TrackTranslate\TrackTranslateEditionUID

id: 0x66FC

type: uinteger

definition: Specify a chapter edition UID on which this TrackTranslate applies.

usage notes: When no TrackTranslateEditionUID is specified in the TrackTranslate, the TrackTranslate applies to all chapter editions found in the Segment using the given TrackTranslateCodec.

5.1.4.1.31. Video Element

name: Video

path: \Segment\Tracks\TrackEntry\Video

id: 0xE0

maxOccurs: 1

type: master

definition: Video settings.

5.1.4.1.31.1. FlagInterlaced Element

name: FlagInterlaced

path: \Segment\Tracks\TrackEntry\Video\FlagInterlaced

id: 0x9A

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 2

definition:
Specify whether the video frames in this track are interlaced or not.

defined values:

value	label	definition
0	undetermined	Unknown status.This value **SHOULD** be avoided.
1	interlaced	Interlaced frames.
2	progressive	No interlacing.

Table 5: FlagInterlaced values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.2. FieldOrder Element

name: FieldOrder

path: \Segment\Tracks\TrackEntry\Video\FieldOrder

id: 0x9D

minOccurs: 1

maxOccurs: 1

default: 2

type: uinteger

minver: 4

definition: Specify the field ordering of video frames in this track.

defined values:

value	label	definition
0	progressive	Interlaced frames.This value **SHOULD** be avoided, setting FlagInterlaced to 2 is sufficient.
1	tff	Top field displayed first. Top field stored first.
2	undetermined	Unknown field order.This value **SHOULD** be avoided.
6	bff	Bottom field displayed first. Bottom field stored first.

value	label	definition
9	bff(swapped)	Top field displayed first. Fields are interleaved in storage
with the top line of the top field stored first.		
14	tff(swapped)	Bottom field displayed first. Fields are interleaved in storage
with the top line of the top field stored first.		

Table 6: FieldOrder values

usage notes: If FlagInterlaced is not set to 1, this Element **MUST** be ignored.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.3. StereoMode Element

name: StereoMode

path: \Segment\Tracks\TrackEntry\Video\StereoMode

id: 0x53B8

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 3

definition: Stereo-3D video mode. There are some more details in Section 18.10.

restrictions:

value	label
0	mono
1	side by side (left eye first)
2	top - bottom (right eye is first)
3	top - bottom (left eye is first)
4	checkboard (right eye is first)

value	label
5	checkboard (left eye is first)
6	row interleaved (right eye is first)
7	row interleaved (left eye is first)
8	column interleaved (right eye is first)
9	column interleaved (left eye is first)
10	anaglyph (cyan/red)
11	side by side (right eye first)
12	anaglyph (green/magenta)
13	both eyes laced in one Block (left eye is first)
14	both eyes laced in one Block (right eye is first)

Table 7: StereoMode values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.4. AlphaMode Element

name: AlphaMode

path: \Segment\Tracks\TrackEntry\Video\AlphaMode

id: 0x53C0

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 3

definition: Indicate whether the BlockAdditional Element with BlockAddID of "1" contains Alpha data, as defined by to the Codec Mapping for the CodecID. Undefined values **SHOULD NOT** be used as the behavior of known implementations is different (considered either as 0 or 1).

defined values:

value	label	definition
0	none	The BlockAdditional Element with BlockAddID of "1" does not exist or **SHOULD NOT** be considered as containing such data.
1	present	The BlockAdditional Element with BlockAddID of "1" contains alpha channel data.

Table 8: AlphaMode values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.5. PixelWidth Element

name: PixelWidth

path: \Segment\Tracks\TrackEntry\Video\PixelWidth

id: 0xB0

minOccurs: 1

maxOccurs: 1

range: not 0

type: uinteger

definition: Width of the encoded video frames in pixels.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.6. PixelHeight Element

name: PixelHeight

path: \Segment\Tracks\TrackEntry\Video\PixelHeight

id: 0xBA

minOccurs: 1

maxOccurs: 1

range: not 0

type: uinteger

definition: Height of the encoded video frames in pixels.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.7. PixelCropBottom Element

name: PixelCropBottom

path:
\Segment\Tracks\TrackEntry\Video\PixelCropBottom

id: 0x54AA

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: The number of video pixels to remove at the bottom of the image.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.8. PixelCropTop Element

name: PixelCropTop

path: \Segment\Tracks\TrackEntry\Video\PixelCropTop

id: 0x54BB

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.9. PixelCropLeft Element

name: PixelCropLeft

path: \Segment\Tracks\TrackEntry\Video\PixelCropLeft

id: 0x54CC

minOccurs: 1

maxOccurs:
1

default: 0

type: uinteger

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.10. PixelCropRight Element

name: PixelCropRight

path: \Segment\Tracks\TrackEntry\Video\PixelCropRight

id: 0x54DD

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.11. DisplayWidth Element

name: DisplayWidth

path: \Segment\Tracks\TrackEntry\Video\DisplayWidth

id: 0x54B0

maxOccurs: 1

range: not 0

default: see implementation notes

type: uinteger

definition:
Width of the video frames to display. Applies to the video frame after cropping (PixelCrop* Elements).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

notes:

attribute	note
default	If the DisplayUnit of the same TrackEntry is 0, then the default value for DisplayWidth is equal toPixelWidth - PixelCropLeft - PixelCropRight, else there is no default value.

Table 9: DisplayWidth implementation notes

5.1.4.1.31.12. DisplayHeight Element

name: DisplayHeight

path: \Segment\Tracks\TrackEntry\Video\DisplayHeight

id: 0x54BA

maxOccurs: 1

range: not 0

default: see implementation notes

type: uinteger

definition: Height of the video frames to display. Applies to the video frame after cropping (PixelCrop* Elements).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

notes:

attribute	note
default	If the DisplayUnit of the same TrackEntry is 0, then the default value for DisplayHeight is equal toPixelHeight - PixelCropTop - PixelCropBottom, else there is no default value.

Table 10: DisplayHeight implementation notes

5.1.4.1.31.13. DisplayUnit Element

name: DisplayUnit

path:
\Segment\Tracks\TrackEntry\Video\DisplayUnit

id: 0x54B2

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: How DisplayWidth & DisplayHeight are interpreted.

restrictions:

value	label
0	pixels
1	centimeters
2	inches
3	display aspect ratio
4	unknown

Table 11: DisplayUnit values

5.1.4.1.31.14. UncompressedFourCC Element

name: UncompressedFourCC

path: \Segment\Tracks\TrackEntry\Video\UncompressedFourCC

id: 0x2EB524

minOccurs: see implementation notes

maxOccurs: 1

type: binary

definition: Specify the uncompressed pixel format used for the Track's data as a FourCC. This value is similar in scope to the biCompression value of AVI's BITMAPINFO [AVIFormat]. See the YUV video formats [FourCC-YUV] and RGB video formats [FourCC-RGB] for common values.

usage notes: This Element **MUST NOT** be used if the CodecID Element of the TrackEntry is set to "V_UNCOMPRESSED".

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

notes:

attribute	note
minOccurs	UncompressedFourCC **MUST** be set (minOccurs=1) in TrackEntry, when the CodecID Element of the TrackEntry is set to "V_UNCOMPRESSED".

Table 12: UncompressedFourCC implementation notes

5.1.4.1.31.15. Colour Element

name: Colour

path: \Segment\Tracks\TrackEntry\Video\Colour

id: 0x55B0

maxOccurs: 1

type: master

minver: 4

definition: Settings describing the colour format.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.16. MatrixCoefficients Element

name: MatrixCoefficients

path: \Segment\Tracks\TrackEntry\Video\Colour\MatrixCoefficients

id: 0x55B1

minOccurs: 1

maxOccurs: 1

default: 2

type: uinteger

minver: 4

definition: 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 23001-8:2016 or ITU-T H.273.

restrictions:

value	label
0	Identity
1	ITU-R BT.709
2	unspecified
3	reserved
4	US FCC 73.682
5	ITU-R BT.470BG
6	SMPTE 170M
7	SMPTE 240M
8	YCoCg
9	BT2020 Non-constant Luminance
10	BT2020 Constant Luminance
11	SMPTE ST 2085
12	Chroma-derived Non-constant Luminance
13	Chroma-derived Constant Luminance
14	ITU-R BT.2100-0

Table 13: MatrixCoefficients values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.17. BitsPerChannel Element

name: BitsPerChannel

path: \Segment\Tracks\TrackEntry\Video\Colour\BitsPerChannel

id: 0x55B2

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 4

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.18. ChromaSubsamplingHorz Element

name: ChromaSubsamplingHorz

path:
\Segment\Tracks\TrackEntry\Video\Colour\ChromaSubsamplingHorz

id: 0x55B3

maxOccurs: 1

type: uinteger

minver: 4

definition: 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.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.19. ChromaSubsamplingVert Element

name: ChromaSubsamplingVert

path: \Segment\Tracks\TrackEntry\Video\Colour\ChromaSubsamplingVert

id: 0x55B4

maxOccurs: 1

type: uinteger

minver: 4

definition: 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.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.20. CbSubsamplingHorz Element

name: CbSubsamplingHorz

path: \Segment\Tracks\TrackEntry\Video\Colour\CbSubsamplingHorz

id: 0x55B5

maxOccurs: 1

type:
uinteger

minver: 4

definition: 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.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.21. CbSubsamplingVert Element

name: CbSubsamplingVert

path: \Segment\Tracks\TrackEntry\Video\Colour\CbSubsamplingVert

id: 0x55B6

maxOccurs: 1

type: uinteger

minver: 4

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.22. ChromaSitingHorz Element

name: ChromaSitingHorz

path: \Segment\Tracks\TrackEntry\Video\Colour\ChromaSitingHorz

id: 0x55B7

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver:
4

definition: How chroma is subsampled horizontally.

restrictions:

value	label
0	unspecified
1	left collocated
2	half

Table 14:
ChromaSitingHorz values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.23. ChromaSitingVert Element

name: ChromaSitingVert

path: \Segment\Tracks\TrackEntry\Video\Colour\ChromaSitingVert

id: 0x55B8

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 4

definition: How chroma is subsampled vertically.

restrictions:

value	label
0	unspecified
1	top collocated
2	half

Table 15:
ChromaSitingVert values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.24. Range Element

name: Range

path: \Segment\Tracks\TrackEntry\Video\Colour\Range

id: 0x55B9

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 4

definition: Clipping of the color ranges.

restrictions:

value	label
0	unspecified
1	broadcast range
2	full range (no clipping)
3	defined by MatrixCoefficients / TransferCharacteristics

Table 16: Range values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.25. TransferCharacteristics Element

name: TransferCharacteristics

path:
\Segment\Tracks\TrackEntry\Video\Colour\TransferCharacteristics

id: 0x55BA

minOccurs: 1

maxOccurs: 1

default: 2

type: uinteger

minver: 4

definition:

The transfer characteristics of the video. For clarity, the value and meanings for TransferCharacteristics are adopted from Table 3 of ISO/IEC 23091-4 or ITU-T H.273.

restrictions:

value	label
0	reserved
1	ITU-R BT.709
2	unspecified
3	reserved
4	Gamma 2.2 curve - BT.470M
5	Gamma 2.8 curve - BT.470BG
6	SMPTE 170M
7	SMPTE 240M
8	Linear
9	Log
10	Log Sqrt
11	IEC 61966-2-4
12	ITU-R BT.1361 Extended Colour Gamut
13	IEC 61966-2-1
14	ITU-R BT.2020 10 bit
15	ITU-R BT.2020 12 bit
16	ITU-R BT.2100 Perceptual Quantization
17	SMPTE ST 428-1
18	ARIB STD-B67 (HLG)

Table 17: TransferCharacteristics values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.26. Primaries Element

name: Primaries

path: \Segment\Tracks\TrackEntry\Video\Colour\Primaries

id: 0x55BB

minOccurs: 1

maxOccurs: 1

default: 2

type: uinteger

minver:
4

definition: The colour primaries of the video. For clarity, the value and meanings for Primaries are adopted from Table 2 of ISO/IEC 23091-4 or ITU-T H.273.

restrictions:

value	label
0	reserved
1	ITU-R BT.709
2	unspecified
3	reserved
4	ITU-R BT.470M
5	ITU-R BT.470BG - BT.601 625
6	ITU-R BT.601 525 - SMPTE 170M
7	SMPTE 240M
8	FILM
9	ITU-R BT.2020
10	SMPTE ST 428-1
11	SMPTE RP 432-2
12	SMPTE EG 432-2
22	EBU Tech. 3213-E - JEDEC P22 phosphors

Table 18: Primaries values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.27. MaxCLL Element

name: MaxCLL

path: \Segment\Tracks\TrackEntry\Video\Colour\MaxCLL

id: 0x55BC

maxOccurs: 1

type: uinteger

minver: 4

definition: Maximum brightness of a single pixel (Maximum Content Light Level) in candelas per square meter (cd/m^2).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.28. MaxFALL Element

name: MaxFALL

path: \Segment\Tracks\TrackEntry\Video\Colour\MaxFALL

id: 0x55BD

maxOccurs: 1

type: uinteger

minver: 4

definition: Maximum brightness of a single full frame (Maximum Frame-Average Light Level) in candelas per square meter (cd/m^2).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.29. MasteringMetadata Element

name: MasteringMetadata

path: \Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata

id: 0x55D0

maxOccurs: 1

type: master

minver: 4

definition: SMPTE 2086 mastering data.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.30. PrimaryRChromaticityX Element

name: PrimaryRChromaticityX

path: \Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\PrimaryRChromaticityX

id: 0x55D1

maxOccurs: 1

range:
0-1

type: float

minver: 4

definition: Red X chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.31. PrimaryRChromaticityY Element

name: PrimaryRChromaticityY

path:
\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\PrimaryRChromaticityY

id: 0x55D2

maxOccurs: 1

range: 0-1

type: float

minver: 4

definition: Red Y chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.32. PrimaryGChromaticityX Element

name: PrimaryGChromaticityX

path:
\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\PrimaryGChromaticityX

id: 0x55D3

maxOccurs: 1

range: 0-1

type: float

minver:
4

definition: Green X chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.33. PrimaryGChromaticityY Element

name: PrimaryGChromaticityY

path:
\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\PrimaryGChromaticityY

id: 0x55D4

maxOccurs: 1

range: 0-1

type: float

minver: 4

definition: Green Y chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.34. PrimaryBChromaticityX Element

name: PrimaryBChromaticityX

path:
\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\PrimaryBChromaticityX

id: 0x55D5

maxOccurs: 1

range: 0-1

type: float

minver: 4

definition:
Blue X chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.35. PrimaryBChromaticityY Element

name: PrimaryBChromaticityY

path:
\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\PrimaryBChromaticityY

id: 0x55D6

maxOccurs: 1

range: 0-1

type: float

minver: 4

definition: Blue Y chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.36. WhitePointChromaticityX Element

name: WhitePointChromaticityX

path: \Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\WhitePointChromaticityX

id: 0x55D7

maxOccurs: 1

range: 0-1

type: float

minver: 4

definition: White X chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.37. WhitePointChromaticityY Element

name: WhitePointChromaticityY

path: \Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\WhitePointChromaticityY

id: 0x55D8

maxOccurs: 1

range: 0-1

type: float

minver: 4

definition: White Y chromaticity coordinate, as defined by CIE 1931.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.38. LuminanceMax Element

name: LuminanceMax

path:
\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\LuminanceMax

id: 0x55D9

maxOccurs: 1

range: >= 0x0p+0

type: float

minver: 4

definition: Maximum luminance. Represented in candelas per square meter (cd/m^2).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.39. LuminanceMin Element

name: LuminanceMin

path:
\Segment\Tracks\TrackEntry\Video\Colour\MasteringMetadata\LuminanceMin

id: 0x55DA

maxOccurs: 1

range: >= 0x0p+0

type: float

minver: 4

definition: Minimum luminance. Represented in candelas per square meter (cd/m^2).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.40. Projection Element

name: Projection

path: \Segment\Tracks\TrackEntry\Video\Projection

id: 0x7670

maxOccurs:
1

type: master

minver: 4

definition: Describes the video projection details. Used to render spherical, VR videos or flipping videos horizontally/vertically.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.41. ProjectionType Element

name: ProjectionType

path: \Segment\Tracks\TrackEntry\Video\Projection\ProjectionType

id: 0x7671

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 4

definition: Describes the projection used for this video track.

restrictions:

value	label
0	rectangular
1	equirectangular
2	cubemap
3	mesh

Table 19: ProjectionType values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.42. ProjectionPrivate Element

name: ProjectionPrivate

path:
\Segment\Tracks\TrackEntry\Video\Projection\ProjectionPrivate

id: 0x7672

maxOccurs: 1

type: binary

minver: 4

definition: Private data that only applies to a specific projection.

*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 ISOBMFF 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 ISOBMFF 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 ISOBMFF Mesh Projection Box ('mshp').

usage notes: ISOBMFF 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.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.43. ProjectionPoseYaw Element

name: ProjectionPoseYaw

path: \Segment\Tracks\TrackEntry\Video\Projection\ProjectionPoseYaw

id: 0x7673

minOccurs: 1

maxOccurs: 1

range: >= -0xB4p+0, <= 0xB4p+0

default:
0x0p+0

type: float

minver: 4

definition: Specifies a yaw rotation to the projection.

Value 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 element **MUST** be in the -180 to 180 degree range, both included.

Setting ProjectionPoseYaw to 180 or -180 degrees, with the ProjectionPoseRoll and ProjectionPosePitch set to 0 degrees flips the image horizontally.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.44. ProjectionPosePitch Element

name: ProjectionPosePitch

path:
\Segment\Tracks\TrackEntry\Video\Projection\ProjectionPosePitch

id: 0x7674

minOccurs: 1

maxOccurs: 1

range: >= -0x5Ap+0, <= 0x5Ap+0

default: 0x0p+0

type: float

minver: 4

definition: Specifies a pitch rotation to the projection.

Value 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 element **MUST** be in the -90 to 90 degree range, both included.

usage notes:

The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.31.45. ProjectionPoseRoll Element

name: ProjectionPoseRoll

path:
\Segment\Tracks\TrackEntry\Video\Projection\ProjectionPoseRoll

id: 0x7675

minOccurs: 1

maxOccurs: 1

range: >= -0xB4p+0, <= 0xB4p+0

default: 0x0p+0

type: float

minver: 4

definition: Specifies a roll rotation to the projection.

Value 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 element **MUST** be in the -180 to 180 degree range, both included.

Setting ProjectionPoseRoll to 180 or -180 degrees, the ProjectionPoseYaw to 180 or -180 degrees with ProjectionPosePitch set to 0 degrees flips the image vertically.

Setting ProjectionPoseRoll to 180 or -180 degrees, with the ProjectionPoseYaw and ProjectionPosePitch set to 0 degrees flips the image horizontally and vertically.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.32. Audio Element

name: Audio

path: \Segment\Tracks\TrackEntry\Audio

id: 0xE1

maxOccurs: 1

type:
master

definition: Audio settings.

5.1.4.1.32.1. SamplingFrequency Element

name: SamplingFrequency

path: \Segment\Tracks\TrackEntry\Audio\SamplingFrequency

id: 0xB5

minOccurs: 1

maxOccurs: 1

range: > 0x0p+0

default: 0x1.f4p+12

type: float

definition: Sampling frequency in Hz.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.32.2. OutputSamplingFrequency Element

name: OutputSamplingFrequency

path: \Segment\Tracks\TrackEntry\Audio\OutputSamplingFrequency

id: 0x78B5

maxOccurs: 1

range: > 0x0p+0

default: see implementation notes

type: float

definition: Real output sampling frequency in Hz (used for SBR techniques).

notes:

attribute	note
default	The default value for OutputSamplingFrequency of the same TrackEntry is equal to the SamplingFrequency.

Table 20: OutputSamplingFrequency implementation notes

5.1.4.1.32.3. Channels Element

name: Channels

path: \Segment\Tracks\TrackEntry\Audio\Channels

id: 0x9F

minOccurs: 1

maxOccurs: 1

range: not 0

default: 1

type: uinteger

definition: Numbers of channels in the track.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.32.4. BitDepth Element

name: BitDepth

path: \Segment\Tracks\TrackEntry\Audio\BitDepth

id: 0x6264

maxOccurs: 1

range: not 0

type: uinteger

definition: Bits per sample, mostly used for PCM.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.33. TrackOperation Element

name: TrackOperation

path: \Segment\Tracks\TrackEntry\TrackOperation

id:
0xE2

maxOccurs: 1

type: master

minver: 3

definition: Operation that needs to be applied on tracks to create this virtual track. For more details look at Section 18.8.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.33.1. TrackCombinePlanes Element

name: TrackCombinePlanes

path: \Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes

id: 0xE3

maxOccurs: 1

type: master

minver: 3

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.33.2. TrackPlane Element

name: TrackPlane

path:
\Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes\TrackPlane

id: 0xE4

minOccurs: 1

type: master

minver: 3

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.33.3. TrackPlaneUID Element

name: TrackPlaneUID

path:
\Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes\TrackPlane\TrackPlaneUID

id: 0xE5

minOccurs: 1

maxOccurs: 1

range: not 0

type: uinteger

minver: 3

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.33.4. TrackPlaneType Element

name: TrackPlaneType

path:
\Segment\Tracks\TrackEntry\TrackOperation\TrackCombinePlanes\TrackPlane\TrackPlaneType

id: 0xE6

minOccurs: 1

maxOccurs: 1

type: uinteger

minver: 3

definition:
The kind of plane this track corresponds to.

restrictions:

value	label
0	left eye
1	right eye
2	background

Table 21: TrackPlaneType values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.33.5. TrackJoinBlocks Element

name: TrackJoinBlocks

path: \Segment\Tracks\TrackEntry\TrackOperation\TrackJoinBlocks

id: 0xE9

maxOccurs: 1

type: master

minver: 3

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.33.6. TrackJoinUID Element

name: TrackJoinUID

path:
\Segment\Tracks\TrackEntry\TrackOperation\TrackJoinBlocks\TrackJoinUID

id: 0xED

minOccurs: 1

range: not 0

type:
uinteger

minver: 3

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34. ContentEncodings Element

name: ContentEncodings

path: \Segment\Tracks\TrackEntry\ContentEncodings

id: 0x6D80

maxOccurs: 1

type: master

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.1. ContentEncoding Element

name: ContentEncoding

path: \Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding

id: 0x6240

minOccurs: 1

type: master

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.2. ContentEncodingOrder Element

name: ContentEncodingOrder

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncodingOrder

id: 0x5031

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: Tell in which order to apply each ContentEncoding of the ContentEncodings. The decoder/demuxer **MUST** start with the ContentEncoding with the highest ContentEncodingOrder and work its way down to the ContentEncoding with the lowest ContentEncodingOrder. This value **MUST** be unique over for each ContentEncoding found in the ContentEncodings of this TrackEntry.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.3. ContentEncodingScope Element

name: ContentEncodingScope

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncodingScope

id: 0x5032

minOccurs: 1

maxOccurs: 1

default: 1

type: uinteger

definition: A bit field that describes which Elements have been modified in this way. Values (big-endian) can be OR'ed.

defined values:

value	label	definition
1	Block	All frame contents, excluding lacing data.

value	label	definition
2	Private	The track's private data.
4	Next	The next ContentEncoding (next ContentEncodingOrder. Either the data inside ContentCompression and/or ContentEncryption).This value **SHOULD NOT** be used.

Table 22: ContentEncodingScope values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.4. ContentEncodingType Element

name: ContentEncodingType

path: \Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncodingType

id: 0x5033

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: A value describing what kind of transformation is applied.

restrictions:

value	label
0	Compression
1	Encryption

Table 23: ContentEncodingType values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.5. ContentCompression Element

name: ContentCompression

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentCompression

id: 0x5034

maxOccurs: 1

type: master

definition: 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.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.6. ContentCompAlgo Element

name: ContentCompAlgo

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentCompression\ContentCompAlgo

id: 0x4254

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: The compression algorithm used.

defined values:

value	label	definition
0	zlib	zlib compression [RFC1950].
1	bzlib	bzip2 compression [BZIP2], **SHOULD NOT** be used; see usage notes.
2	lzo1x	

value	label	definition
		Lempel-Ziv-Oberhumer compression [LZO], **SHOULD NOT** be used; see usage notes.
3	Header Stripping	Octets in ContentCompSettings (Section 5.1.4.1.34.7) have been stripped from each frame.

Table 24: ContentCompAlgo values

usage notes: Compression method "1" (bzlib) and "2" (lzo1x) are lacking proper documentation on the format which limits implementation possibilities. Due to licensing conflicts on commonly available libraries compression methods "2" (lzo1x) does not offer widespread interoperability. Decoding implementations **MAY** support methods "1" and "2" as possible. The use of these compression methods **SHOULD NOT** be used as a default.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.7. ContentCompSettings Element

name: ContentCompSettings

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentCompression\ContentCompSettings

id: 0x4255

maxOccurs: 1

type: binary

definition: 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.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.8. ContentEncryption Element

name: ContentEncryption

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption

id: 0x5035

maxOccurs:
1

type: master

definition: Settings describing the encryption used. This Element **MUST** be present if the value of ContentEncodingType is 1 (encryption) and **MUST** be ignored otherwise.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.9. ContentEncAlgo Element

name: ContentEncAlgo

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentEncAlgo

id: 0x47E1

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: The encryption algorithm used. The value "0" means that the contents have not been encrypted.

defined values:

value	label	definition
0	Not encrypted	
1	DES	Data Encryption Standard (DES) [FIPS.46-3].
2	3DES	Triple Data Encryption Algorithm [RFC1851].
3	Twofish	Twofish Encryption Algorithm [Twofish].
4	Blowfish	Blowfish Encryption Algorithm [Blowfish].
5	AES	Advanced Encryption Standard (AES) [FIPS.197].

Table 25: ContentEncAlgo values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.10. ContentEncKeyID Element

name:

ContentEncKeyID

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentEncKeyID

id: 0x47E2

maxOccurs: 1

type: binary

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.11. ContentEncAESSettings Element

name: ContentEncAESSettings

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentEncAESSettings

id: 0x47E7

maxOccurs: 1

type: master

minver: 4

definition: Settings describing the encryption algorithm used. It **MUST** be ignored if ContentEncAlgo is not AES (5).

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.4.1.34.12. AESSettingsCipherMode Element

name: AESSettingsCipherMode

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentEncAESSettings\AESSettingsCipherMode

id: 0x47E8

minOccurs: 1

maxOccurs:
1

type: uinteger

minver: 4

definition: The AES cipher mode used in the encryption. It **MUST** be ignored if ContentEncAlgo is not AES (5).

defined values:

value	label	definition
1	AES-CTR	Counter [SP.800-38A].
2	AES-CBC	Cipher Block Chaining [SP.800-38A].

Table 26: AESSettingsCipherMode values

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.5. Cues Element

name: Cues

path: \Segment\Cues

id: 0x1C53BB6B

minOccurs: see implementation notes

maxOccurs: 1

type: master

definition: A Top-Level Element to speed seeking access. All entries are local to the Segment.

notes:

attribute	note
minOccurs	This Element **SHOULD** be set when the Segment is not transmitted as a live stream (see #livestreaming).

Table 27: Cues implementation notes

5.1.5.1. CuePoint Element

name: CuePoint

path:
\Segment\Cues\CuePoint

id: 0xBB

minOccurs: 1

type: master

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

5.1.5.1.1. CueTime Element

name: CueTime

path: \Segment\Cues\CuePoint\CueTime

id: 0xB3

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: Absolute timestamp of the seek point, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1.

5.1.5.1.2. CueTrackPositions Element

name: CueTrackPositions

path: \Segment\Cues\CuePoint\CueTrackPositions

id: 0xB7

minOccurs: 1

type: master

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

5.1.5.1.2.1. CueTrack Element

name: CueTrack

path: \Segment\Cues\CuePoint\CueTrackPositions\CueTrack

id: 0xF7

minOccurs:
1

maxOccurs: 1

range: not 0

type: uinteger

definition: The track for which a position is given.

5.1.5.1.2.2. CueClusterPosition Element

name: CueClusterPosition

path: \Segment\Cues\CuePoint\CueTrackPositions\CueClusterPosition

id: 0xF1

minOccurs: 1

maxOccurs: 1

type: uinteger

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

5.1.5.1.2.3. CueRelativePosition Element

name: CueRelativePosition

path: \Segment\Cues\CuePoint\CueTrackPositions\CueRelativePosition

id: 0xF0

maxOccurs: 1

type: uinteger

minver: 4

definition: The relative position inside the Cluster of the referenced SimpleBlock or BlockGroup with 0 being the first possible position for an Element inside that Cluster.

5.1.5.1.2.4. CueDuration Element

name: CueDuration

path: \Segment\Cues\CuePoint\CueTrackPositions\CueDuration

id:
0xB2

maxOccurs: 1

type: uinteger

minver: 4

definition: The duration of the block, expressed in Segment Ticks which is based on TimestampScale; see Section 10.1. If missing, the track's DefaultDuration does not apply and no duration information is available in terms of the cues.

5.1.5.1.2.5. CueBlockNumber Element

name: CueBlockNumber

path: \Segment\Cues\CuePoint\CueTrackPositions\CueBlockNumber

id: 0x5378

maxOccurs: 1

range: not 0

type: uinteger

definition: Number of the Block in the specified Cluster.

5.1.5.1.2.6. CueCodecState Element

name: CueCodecState

path: \Segment\Cues\CuePoint\CueTrackPositions\CueCodecState

id: 0xEA

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

minver: 2

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

5.1.5.1.2.7. CueReference Element

name: CueReference

path: \Segment\Cues\CuePoint\CueTrackPositions\CueReference

id: 0xDB

type: master

minver: 2

definition: The Clusters containing the referenced Blocks.

5.1.5.1.2.8. CueRefTime Element

name: CueRefTime

path:
\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefTime

id: 0x96

minOccurs: 1

maxOccurs: 1

type: uinteger

minver: 2

definition: Timestamp of the referenced Block, expressed in
Matroska Ticks -- ie in nanoseconds; see Section 10.1.

5.1.6. Attachments Element

name: Attachments

path: \Segment\Attachments

id: 0x1941A469

maxOccurs: 1

type: master

definition: Contain attached files.

5.1.6.1. AttachedFile Element

name: AttachedFile

path:
\Segment\Attachments\AttachedFile

id: 0x61A7

minOccurs: 1

type: master

definition: An attached file.

5.1.6.1.1. FileDescription Element

name: FileDescription

path: \Segment\Attachments\AttachedFile\FileDescription

id: 0x467E

maxOccurs: 1

type: utf-8

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

5.1.6.1.2. FileName Element

name: FileName

path: \Segment\Attachments\AttachedFile\FileName

id: 0x466E

minOccurs: 1

maxOccurs: 1

type: utf-8

definition: Filename of the attached file.

5.1.6.1.3. FileMimeType Element

name: FileMimeType

path: \Segment\Attachments\AttachedFile\FileMimeType

id: 0x4660

minOccurs: 1

maxOccurs:
1

type: string

definition: MIME type of the file.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.6.1.4. FileData Element

name: FileData

path: \Segment\Attachments\AttachedFile\FileData

id: 0x465C

minOccurs: 1

maxOccurs: 1

type: binary

definition: The data of the file.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.6.1.5. FileUID Element

name: FileUID

path: \Segment\Attachments\AttachedFile\FileUID

id: 0x46AE

minOccurs: 1

maxOccurs: 1

range: not 0

type: uinteger

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.7. Chapters Element

name:
Chapters

path: \Segment\Chapters

id: 0x1043A770

maxOccurs: 1

type: master

recurring: 1

definition: A system to define basic menus and partition data. For more detailed information, look at the Chapters explanation in Section 20.

5.1.7.1. EditionEntry Element

name: EditionEntry

path: \Segment\Chapters\EditionEntry

id: 0x45B9

minOccurs: 1

type: master

definition: Contains all information about a Segment edition.

5.1.7.1.1. EditionUID Element

name: EditionUID

path: \Segment\Chapters\EditionEntry\EditionUID

id: 0x45BC

maxOccurs: 1

range: not 0

type: uinteger

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

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.7.1.2. EditionFlagDefault Element

name: EditionFlagDefault

path: \Segment\Chapters\EditionEntry\EditionFlagDefault

id: 0x45DB

minOccurs: 1

maxOccurs: 1

range: 0-1

default: 0

type: uinteger

definition: Set to 1 if the edition **SHOULD** be used as the default one.

5.1.7.1.3. EditionFlagOrdered Element

name: EditionFlagOrdered

path: \Segment\Chapters\EditionEntry\EditionFlagOrdered

id: 0x45DD

minOccurs: 1

maxOccurs: 1

range: 0-1

default: 0

type: uinteger

definition: Set to 1 if the chapters can be defined multiple times and the order to play them is enforced; see Section 20.1.3.

5.1.7.1.4. ChapterAtom Element

name: ChapterAtom

path: \Segment\Chapters\EditionEntry\+ChapterAtom

id: 0xB6

minOccurs: 1

type:
master

recursive: 1

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

5.1.7.1.4.1. ChapterUID Element

name: ChapterUID

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterUID

id: 0x73C4

minOccurs: 1

maxOccurs: 1

range: not 0

type: uinteger

definition: A unique ID to identify the Chapter.

usage notes: The value of this Element **SHOULD** be kept the same when making a direct stream copy to another file.

5.1.7.1.4.2. ChapterStringUID Element

name: ChapterStringUID

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterStringUID

id: 0x5654

maxOccurs: 1

type: utf-8

minver: 3

definition: A unique string ID to identify the Chapter. Use for WebVTT cue identifier storage [WebVTT].

5.1.7.1.4.3. ChapterTimeStart Element

name: ChapterTimeStart

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterTimeStart

id:
0x91

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: Timestamp of the start of Chapter, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1.

5.1.7.1.4.4. ChapterTimeEnd Element

name: ChapterTimeEnd

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterTimeEnd

id: 0x92

minOccurs: see implementation notes

maxOccurs: 1

type: uinteger

definition: Timestamp of the end of Chapter timestamp excluded, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1. The value **MUST** be greater than or equal to the ChapterTimeStart of the same ChapterAtom.

usage notes: The ChapterTimeEnd timestamp value being excluded, it **MUST** take in account the duration of the last frame it includes, especially for the ChapterAtom using the last frames of the Segment.

notes:

attribute	note
minOccurs	ChapterTimeEnd **MUST** be set (minOccurs=1) if the Edition is an ordered edition; see Section 20.1.3, unless it's a Parent Chapter; see Section 20.2.3

Table 28: ChapterTimeEnd implementation notes

5.1.7.1.4.5. ChapterFlagHidden Element

name: ChapterFlagHidden

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterFlagHidden

id: 0x98

minOccurs:
1

maxOccurs: 1

range: 0-1

default: 0

type: uinteger

definition: Set to 1 if a chapter is hidden. Hidden chapters **SHOULD NOT** be available to the user interface (but still to Control Tracks; see Section 20.2.5 on Chapter flags).

5.1.7.1.4.6. ChapterSegmentUID Element

name: ChapterSegmentUID

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterSegmentUID

id: 0x6E67

minOccurs: see implementation notes

maxOccurs: 1

range: >0

type: binary

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

usage notes: The value **MUST NOT** be the SegmentUID value of the Segment it belongs to.

notes:

attribute	note
minOccurs	ChapterSegmentUID **MUST** be set (minOccurs=1) if ChapterSegmentEditionUID is used; see Section 17.2 on medium-linking Segments.

Table 29: ChapterSegmentUID implementation notes

5.1.7.1.4.7. ChapterSegmentEditionUID Element

name: ChapterSegmentEditionUID

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapterSegmentEditionUID

id:
0x6EBC

maxOccurs: 1

range: not 0

type: uinteger

definition: The EditionUID to play from the Segment linked in ChapterSegmentUID. If ChapterSegmentEditionUID is undeclared, then no Edition of the linked Segment is used; see Section 17.2 on medium-linking Segments.

5.1.7.1.4.8. ChapterPhysicalEquiv Element

name: ChapterPhysicalEquiv

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterPhysicalEquiv

id: 0x63C3

maxOccurs: 1

type: uinteger

definition: Specify the physical equivalent of this ChapterAtom like "DVD" (60) or "SIDE" (50); see Section 20.4 for a complete list of values.

5.1.7.1.4.9. ChapterDisplay Element

name: ChapterDisplay

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterDisplay

id: 0x80

type: master

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

5.1.7.1.4.10. ChapString Element

name: ChapString

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapterDisplay\ChapString

id:
0x85

minOccurs: 1

maxOccurs: 1

type: utf-8

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

5.1.7.1.4.11. ChapLanguage Element

name: ChapLanguage

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapterDisplay\ChapLanguage

id: 0x437C

minOccurs: 1

default: eng

type: string

definition: A language corresponding to the string, in the bibliographic ISO-639-2 form [ISO639-2]. This Element **MUST** be ignored if a ChapLanguageIETF Element is used within the same ChapterDisplay Element.

5.1.7.1.4.12. ChapLanguageIETF Element

name: ChapLanguageIETF

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapterDisplay\ChapLanguageIETF

id: 0x437D

type: string

minver: 4

definition: Specifies a language corresponding to the ChapString in the format defined in [BCP47] and using the IANA Language Subtag Registry [IANALangRegistry]. If a ChapLanguageIETF Element is used, then any ChapLanguage and ChapCountry Elements used in the same ChapterDisplay **MUST** be ignored.

5.1.7.1.4.13. ChapCountry Element

name:
ChapCountry

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapterDisplay\ChapCountry

id: 0x437E

type: string

definition: A country corresponding to the string, using the same 2 octets country-codes as in Internet domains [IANADomains] based on [ISO3166-1] alpha-2 codes. This Element **MUST** be ignored if a ChapLanguageIETF Element is used within the same ChapterDisplay Element.

5.1.7.1.4.14. ChapProcess Element

name: ChapProcess

path: \Segment\Chapters\EditionEntry\+ChapterAtom\ChapProcess

id: 0x6944

type: master

definition: Contains all the commands associated to the Atom.

5.1.7.1.4.15. ChapProcessCodecID Element

name: ChapProcessCodecID

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapProcess\ChapProcessCodecID

id: 0x6955

minOccurs: 1

maxOccurs: 1

default: 0

type: uinteger

definition: 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; see Section 20.3 on DVD menus. More codec IDs can be added later.

5.1.7.1.4.16. ChapProcessPrivate Element

name:
ChapProcessPrivate

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapProcess\ChapProcessPrivate

id: 0x450D

maxOccurs: 1

type: binary

definition: Some optional data attached to the ChapProcessCodecID information. For ChapProcessCodecID = 1, it is the "DVD level" equivalent; see Section 20.3 on DVD menus.

5.1.7.1.4.17. ChapProcessCommand Element

name: ChapProcessCommand

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapProcess\ChapProcessCommand

id: 0x6911

type: master

definition: Contains all the commands associated to the Atom.

5.1.7.1.4.18. ChapProcessTime Element

name: ChapProcessTime

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapProcess\ChapProcessCommand\ChapProcessTime

id: 0x6922

minOccurs: 1

maxOccurs: 1

type: uinteger

definition: Defines when the process command **SHOULD** be handled

restrictions:

value	label
0	during the whole chapter

value	label
1	before starting playback
2	after playback of the chapter

Table 30: ChapProcessTime values

5.1.7.1.4.19. ChapProcessData Element

name: ChapProcessData

path: \Segment\Chapters\EditionEntry\
+ChapterAtom\ChapProcess\ChapProcessCommand\ChapProcessData

id: 0x6933

minOccurs: 1

maxOccurs: 1

type: binary

definition: 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; see Section 20.3 on DVD menus.

5.1.8. Tags Element

name: Tags

path: \Segment\Tags

id: 0x1254C367

type: master

definition: Element containing metadata describing Tracks, Editions, Chapters, Attachments, or the Segment as a whole. A list of valid tags can be found in [MatroskaTags].

5.1.8.1. Tag Element

name: Tag

path: \Segment\Tags\Tag

id: 0x7373

minOccurs: 1

type: master

definition:
A single metadata descriptor.

5.1.8.1.1. Targets Element

name: Targets

path: \Segment\Tags\Tag\Targets

id: 0x63C0

minOccurs: 1

maxOccurs: 1

type: master

definition: 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.

5.1.8.1.1.1. TargetTypeValue Element

name: TargetTypeValue

path: \Segment\Tags\Tag\Targets\TargetTypeValue

id: 0x68CA

minOccurs: 1

maxOccurs: 1

default: 50

type: uinteger

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

defined values:

value	label	definition
70	COLLECTION	The highest hierarchical level that tags can describe.
60	EDITION / ISSUE / VOLUME / OPUS / SEASON / SEQUEL	A list of lower levels grouped together.
50	ALBUM / OPERA / CONCERT / MOVIE / EPISODE	The most common grouping level of music and video (equals to an episode for TV series).

value	label	definition
40	PART / SESSION	When an album or episode has different logical parts.
30	TRACK / SONG / CHAPTER	The common parts of an album or movie.
20	SUBTRACK / PART / MOVEMENT / SCENE	Corresponds to parts of a track for audio (like a movement).
10	SHOT	The lowest hierarchy found in music or movies.

Table 31: TargetTypeValue values

5.1.8.1.1.2. TargetType Element

name: TargetType

path: \Segment\Tags\Tag\Targets\TargetType

id: 0x63CA

maxOccurs: 1

type: string

definition: An informational string that can be used to display the logical level of the target like "ALBUM", "TRACK", "MOVIE", "CHAPTER", etc ; see Section 6.4 of [MatroskaTags].

restrictions:

value	label
COLLECTION	COLLECTION
EDITION	EDITION
ISSUE	ISSUE
VOLUME	VOLUME
OPUS	OPUS
SEASON	SEASON
SEQUEL	SEQUEL
ALBUM	ALBUM
OPERA	OPERA
CONCERT	CONCERT
MOVIE	MOVIE
EPISODE	EPISODE
PART	PART
SESSION	SESSION
TRACK	TRACK
SONG	SONG
CHAPTER	CHAPTER
SUBTRACK	SUBTRACK

value	label
PART	PART
MOVEMENT	MOVEMENT
SCENE	SCENE
SHOT	SHOT

Table 32: TargetType values

5.1.8.1.1.3. TagTrackUID Element

name: TagTrackUID

path: \Segment\Tags\Tag\Targets\TagTrackUID

id: 0x63C5

default: 0

type: uinteger

definition: A unique ID to identify the Track(s) the tags belong to.

usage notes: If the value is 0 at this level, the tags apply to all tracks in the Segment. If set to any other value, it **MUST** match the TrackUID value of a track found in this Segment.

5.1.8.1.1.4. TagEditionUID Element

name: TagEditionUID

path: \Segment\Tags\Tag\Targets\TagEditionUID

id: 0x63C9

default: 0

type: uinteger

definition: A unique ID to identify the EditionEntry(s) the tags belong to.

usage notes: If the value is 0 at this level, the tags apply to all editions in the Segment. If set to any other value, it **MUST** match the EditionUID value of an edition found in this Segment.

5.1.8.1.1.5. TagChapterUID Element

name: TagChapterUID

path: \Segment\Tags\Tag\Targets\TagChapterUID

id:
0x63C4

default: 0

type: uinteger

definition: A unique ID to identify the Chapter(s) the tags belong to.

usage notes: If the value is 0 at this level, the tags apply to all chapters in the Segment. If set to any other value, it **MUST** match the ChapterUID value of a chapter found in this Segment.

5.1.8.1.1.6. TagAttachmentUID Element

name: TagAttachmentUID

path: \Segment\Tags\Tag\Targets\TagAttachmentUID

id: 0x63C6

default: 0

type: uinteger

definition: A unique ID to identify the Attachment(s) the tags belong to.

usage notes: If the value is 0 at this level, the tags apply to all the attachments in the Segment. If set to any other value, it **MUST** match the FileUID value of an attachment found in this Segment.

5.1.8.1.2. SimpleTag Element

name: SimpleTag

path: \Segment\Tags\Tag\+SimpleTag

id: 0x67C8

minOccurs: 1

type: master

recursive: 1

definition: Contains general information about the target.

5.1.8.1.2.1. TagName Element

name:
TagName

path: \Segment\Tags\Tag\+SimpleTag\TagName

id: 0x45A3

minOccurs: 1

maxOccurs: 1

type: utf-8

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

5.1.8.1.2.2. TagLanguage Element

name: TagLanguage

path: \Segment\Tags\Tag\+SimpleTag\TagLanguage

id: 0x447A

minOccurs: 1

maxOccurs: 1

default: und

type: string

definition: Specifies the language of the tag specified, in the Matroska languages form; see Section 11 on language codes. This Element **MUST** be ignored if the TagLanguageIETF Element is used within the same SimpleTag Element.

5.1.8.1.2.3. TagLanguageIETF Element

name: TagLanguageIETF

path: \Segment\Tags\Tag\+SimpleTag\TagLanguageIETF

id: 0x447B

maxOccurs: 1

type: string

minver: 4

definition: Specifies the language used in the TagString according to [BCP47] and using the IANA Language Subtag Registry [IANALangRegistry]. If this Element is used, then any TagLanguage Elements used in the same SimpleTag **MUST** be ignored.

5.1.8.1.2.4. TagDefault Element

name: TagDefault

path: \Segment\Tags\Tag\+SimpleTag\TagDefault

id: 0x4484

minOccurs: 1

maxOccurs: 1

range: 0-1

default: 1

type: uinteger

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

5.1.8.1.2.5. TagString Element

name: TagString

path: \Segment\Tags\Tag\+SimpleTag\TagString

id: 0x4487

maxOccurs: 1

type: utf-8

definition:
The value of the Tag.

5.1.8.1.2.6. TagBinary Element

name: TagBinary

path: \Segment\Tags\Tag\+SimpleTag\TagBinary

id: 0x4485

maxOccurs: 1

type: binary

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

6. Matroska Element Ordering

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.

6.1. 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.

6.2. 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.

6.3. 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.

6.4. 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.

6.5. Info

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

6.6. Chapters Element

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 is **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.

6.7. 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.

6.8. 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.

6.9. Optimum layout from a muxer

*SeekHead

*Info

*Tracks

*Chapters

*Attachments

*Tags

*Clusters

*Cues

6.10. Optimum layout after editing tags

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

6.11. Optimum layout with Cues at the front

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

7. 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.

8. 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:

* Blu-ray movie: 1000000000ns/(48/1.001) = 20854167ns

* PAL broadcast/DVD: 1000000000ns/(50/1.000) = 20000000ns

* N/ATSC broadcast: 1000000000ns/(60/1.001) = 16683333ns

* hard-telecined DVD: 1000000000ns/(60/1.001) = 16683333ns (60 encoded interlaced fields per second)

* soft-telecined DVD: 1000000000ns/(60/1.001) = 16683333ns (48 encoded interlaced fields per second, with "repeat*first*field = 1")

9. Block Structure

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 timestamps might not be consecutive. But a frame with a past timestamp **MUST** reference a frame already known, otherwise it's considered bad/void.

9.1. 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**	Timestamp (relative to Cluster timestamp, signed int16)

Table 33: Block Header base parts

9.2. 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

Table 34: Block Header flags part

9.3. SimpleBlock Structure

The SimpleBlock is inspired by the Block structure; see Section 9. The main differences are the added Keyframe flag and Discardable flag. Otherwise everything is the same.

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 timestamps might not be consecutive. But a frame with a past timestamp **MUST** reference a frame already known, otherwise it's considered bad/void.

9.3.1. 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**	Timestamp (relative to Cluster timestamp, signed int16)

Table 35: SimpleBlock Header base parts

9.3.2. 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

Table 36: SimpleBlock Header flags part

9.4. Block 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). It packs multiple frames into a single Block or SimpleBlock.

Lacing **MUST NOT** be used to store a single frame in a Block or SimpleBlock.

There are 3 types of lacing:

1. Xiph, inspired by what is found in the Ogg container [RFC3533]

2. EBML, which is the same with sizes coded differently

3. fixed-size, where the size is not coded

When lacing is not used, i.e. to store a single frame, the lacing bits 5 and 6 of the Block or SimpleBlock **MUST** be set to zero.

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.

It is possible not to use lacing at all and just store a single frame without any extra data. When the FlagLacing -- Section 5.1.4.1.12 -- is set to "0" all blocks of that track **MUST NOT** use lacing.

9.4.1. No lacing

When no lacing is used, the number of frames in the lace is ommitted and only one frame can be stored in the Block. The bits 5-6 of the Block Header flags are set to 00.

The Block for a 800 octets frame is as follows:

Block Octets	Value	Description
4-803	<frame>	Single frame data

Table 37: No lacing

When a Block contains a single frame, it **MUST** use this No lacing mode.

9.4.2. Xiph lacing

The Xiph lacing uses the same coding of size as found in the Ogg container [RFC3533]. The bits 5-6 of the Block Header flags are set to 01.

The Block data with laced frames is stored as follows:

- *Lacing Head on 1 Octet: Number of frames in the lace minus 1.
- *Lacing size of each frame except the last one.
- *Binary data of each frame consecutively.

The lacing size is split into 255 values, stored as unsigned octets -- for example, 500 is coded 255;245 or [0xFF 0xF5]. 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 or [0xFF 0xFF 0xFF 0x00].

The size of the last frame is deduced from the size remaining in the Block after the other frames.

Because large sizes result in large coding of the sizes, it is **RECOMMENDED** to use Xiph lacing only with small frames.

In our example, the 800, 500 and 1000 frames are stored with Xiph lacing in a Block as follows:

Block Octet	Value	Description
4	0x02	Number of frames minus 1
5-8	0xFF 0xFF 0xFF 0x23	Size of the first frame (255;255;255;35)

Block Octet	Value	Description
9-10	0xFF 0xF5	Size of the second frame (255;245)
11-810		First frame data
811-1310		Second frame data
1311-2310		Third frame data

Table 38: Xiph lacing example

The Block is 2311 octets large and the last frame starts at 1311, so we can deduce the size of the last frame is 2311 - 1311 = 1000.

9.4.3. EBML lacing

The EBML lacing encodes the frame size with an EBML-like encoding [RFC8794]. The bits 5-6 of the Block Header flags are set to 11.

The Block data with laced frames is stored as follows:

- *Lacing Head on 1 Octet: Number of frames in the lace minus 1.
- *Lacing size of each frame except the last one.
- *Binary data of each frame consecutively.

The first frame size is encoded as an EBML Unsigned Integer Element value. The other frame sizes are encoded as a difference with the previous frame size as EBML Signed Integer Element values. That corresponds to an EBML Data Size Values with two's complement notation with the leftmost bit being the sign bit as found in [RFC8794], giving this range of values:

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 xxxx xxxx	value $-(2^{27}-1)$ to $2^{27}-1$
0000 1xxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	value $-(2^{34}-1)$ to $2^{34}-1$
0000 01xx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	value $-(2^{41}-1)$ to $2^{41}-1$
0000 001x xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	value $-(2^{48}-1)$ to $2^{48}-1$

Table 39: EBML Lacing bits usage

In our example, the 800, 500 and 1000 frames are stored with EBML lacing in a Block as follows:

Block Octets	Value	Description
4	0x02	Number of frames minus 1
5-6	0x43 0x20	Size of the first frame (800 = 0x320 + 0x4000)
7-8	0x5E 0xD3	Size of the second frame (500 - 800 = -300 = -0x12C + 0x1FFF + 0x4000)
8-807	<frame1>	First frame data
808-1307	<frame2>	Second frame data
1308-2307	<frame3>	Third frame data

Table 40: EBML lacing example

The Block is 2308 octets large and the last frame starts at 1308, so we can deduce the size of the last frame is 2308 - 1308 = 1000.

9.4.4. Fixed-size lacing

The Fixed-size lacing doesn't store the frame size, only the number of frames in the lace. Each frame **MUST** have the same size. The frame size of each frame is deduced from the total size of the Block. The bits 5-6 of the Block Header flags are set to 10.

The Block data with laced frames is stored as follows:

*Lacing Head on 1 Octet: Number of frames in the lace minus 1.

*Binary data of each frame consecutively.

For example, for 3 frames of 800 octets each:

Block Octets	Value	Description
4	0x02	Number of frames minus 1
5-804	<frame1>	First frame data
805-1604	<frame2>	Second frame data
1605-2404	<frame3>	Third frame data

Table 41: Fixed-size lacing example

This gives a Block of 2405 octets. When reading the Block we find that there are 3 frames (Octet 4). The data start at Octet 5, so the size of each frame is (2405 - 5) / 3 = 800.

9.4.5. Laced Frames Timestamp

A Block only contains a single timestamp value. But when lacing is used, it contains more than one frame. Each frame originally has its own timestamp, or Presentation Timestamp (PTS). That timestamp applies to the first frame in the lace.

In the lace, each frame after the first one has an undertermined timestamp. But each of these frames **MUST** be contiguous -- i.e. the decoded data **MUST NOT** contain any gap between them. If there is a gap in the stream, the frames around the gap **MUST NOT** be in the same Block.

Lacing is only useful for small contiguous data to save space. This is usually the case for audio tracks and not the case for video -- which use a lot of data -- or subtitle tracks -- which have long gaps. For audio, there is usually a fixed output sampling frequency for the whole track. So the decoder should be able to recover the timestamp of each sample, knowing each output sample is contiguous with a fixed frequency. For subtitles this is usually not the case so lacing **SHOULD NOT** be used.

9.5. Random Access Points

Random Access Points (RAP) are positions where the parser can seek to and start playback without decoding of what was before. In Matroska BlockGroups and SimpleBlocks can be RAPs. To seek to these elements it is still necessary to seek to the Cluster containing them and start playback from the BlockGroup or SimpleBlock that is a RAP.

Because a Matroska File is usually composed of multiple tracks playing at the same time -- video, audio and subtitles -- to seek properly to a RAP, each selected track must be taken in account. Usually all audio and subtitle BlockGroup or SimpleBlock are RAP. They are independent of each other and can be played randomly.

Video tracks on the other hand often use references to previous and future frames for better coding efficiency. Frames with such reference **MUST** either contain one or more ReferenceBlock Elements in their BlockGroup or **MUST** be marked as non-keyframe in a SimpleBlock; see Section 9.3.2.

Frames that are RAP -- i.e. they don't depend on other frames -- **MUST** set the keyframe flag if they are in a SimpleBlock or their parent BlockGroup **MUST NOT** contain a ReferenceBlock.

There may be cases where the use of BlockGroup is necessary, as the frame may need a BlockDuration, BlockAdditions, CodecState or a DiscardPadding element. For thoses cases a SimpleBlock **MUST NOT** be

used, the reference information **SHOULD** be recovered for non-RAP frames.

When a frame in a BlockGroup is not a RAP, all references **SHOULD** be listed as a ReferenceBlock, at least some of them, even if not accurate, or one ReferenceBlock with the value "0" corresponding to a self or unknown reference. The lack of ReferenceBlock would mean such a frame is a RAP and seeking on that frame that actually depends on other frames **MAY** create bogus output or even crash.

Intra-only video frames, such as the ones found in AV1 or VP9, can be decoded without any other frame, but they don't reset the codec state. So seeking to these frames is not possible as the next frames may need frames that are not known from this seeking point. Such intra-only frames **MUST NOT** be considered as keyframes so the keyframe flag **MUST NOT** be set in the SimpleBlock or a ReferenceBlock **MUST** be used to signify the frame is not a RAP. The timestamp value of the ReferenceBlock **MUST** be "0", meaning it's referencing itself.

Because a video SimpleBlock has less references information than a video BlockGroup, it is possible to remux a video track using BlockGroup into a SimpleBlock, as long as it doesn't use any other BlockGroup features than ReferenceBlock.

10. Timestamps

Historically timestamps in Matroska were mistakenly called timecodes. The Timestamp Element was called Timecode, the TimestampScale Element was called TimecodeScale, the TrackTimestampScale Element was called TrackTimecodeScale and the ReferenceTimestamp Element was called ReferenceTimeCode.

10.1. Timestamp Ticks

All timestamp values in Matroska are expressed in multiples of a tick. They are usually stored as integers. There are three types of ticks possible:

10.1.1. Matroska Ticks

For such elements, the timestamp value is stored directly in nanoseconds.

The elements storing values in Matroska Ticks/nanoseconds are:

*TrackEntry\DefaultDuration; defined in Section 5.1.4.1.15

*TrackEntry\DefaultDecodedFieldDuration; defined in Section 5.1.4.1.16

*TrackEntry\SeekPreRoll; defined in Section 5.1.4.1.29

*TrackEntry\CodecDelay; defined in Section 5.1.4.1.28

*BlockGroup\DiscardPadding; defined in Section 5.1.3.5.7

*ChapterAtom\ChapterTimeStart; defined in Section 5.1.7.1.4.3

*ChapterAtom\ChapterTimeEnd; defined in Section 5.1.7.1.4.4

*CuePoint\CueTime; defined in Section 5.1.5.1.1

*CueReference\CueRefTime; defined in Section 5.1.5.1.1

10.1.2. Segment Ticks

Elements in Segment Ticks involve the use of the TimestampScale Element of the Segment to get the timestamp in nanoseconds of the element, with the following formula:

```
timestamp in nanosecond = element value * TimestampScale
```

This allows storing smaller integer values in the elements.

When using the default value of TimestampScale of "1,000,000", one Segment Tick represents one millisecond.

The elements storing values in Segment Ticks are:

*Cluster\Timestamp; defined in Section 5.1.3.1

*Info\Duration is stored as a floating point but the same formula applies; defined in Section 5.1.2.10

*CuePoint\CueTrackPositions\CueDuration; defined in Section 5.1.5.1.2.4

10.1.3. Track Ticks

Elements in Track Ticks involve the use of the TimestampScale Element of the Segment and the TrackTimestampScale Element of the Track to get the timestamp in nanoseconds of the element, with the following formula:

```
timestamp in nanoseconds =
    element value * TrackTimestampScale * TimestampScale
```

This allows storing smaller integer values in the elements. The resulting floating point values of the timestamps are still expressed in nanoseconds.

When using the default values for TimestampScale and TrackTimestampScale of "1,000,000" and of "1.0" respectively, one Track Tick represents one millisecond.

The elements storing values in Track Ticks are:

* Cluster\BlockGroup\Block and Cluster\SimpleBlock timestamps; detailed in Section 10.2

* Cluster\BlockGroup\BlockDuration; defined in Section 5.1.3.5.3

* Cluster\BlockGroup\ReferenceBlock; defined in Section 5.1.3.5.5

When the TrackTimestampScale is interpreted as "1.0", Track Ticks are equivalent to Segment Ticks and give an integer value in nanoseconds. This is the most common case as TrackTimestampScale is usually omitted.

A value of TrackTimestampScale other than "1.0" **MAY** be used to scale the timestamps more in tune with each Track sampling frequency. For historical reasons, a lot of Matroska readers don't take the TrackTimestampScale value in account. So using a value other than "1.0" **MAY** not work in many places.

10.2. Block Timestamps

A Block Element and SimpleBlock Element timestamp is the time when the decoded data of the first frame in the Block/SimpleBlock **MUST** be presented, if the track of that Block/SimpleBlock is selected for playback. This is also known as the Presentation Timestamp (PTS).

The Block Element and SimpleBlock Element store their timestamps as signed integers, relative to the Cluster\Timestamp value of the Cluster they are stored in. To get the timestamp of a Block or SimpleBlock in nanoseconds you have to use the following formula:

(Cluster\Timestamp + (block timestamp * TrackTimestampScale)) * TimestampScale

The Block Element and SimpleBlock Element store their timestamps as 16bit signed integers, allowing a range from "-32768" to "+32767" Track Ticks. Although these values can be negative, when added to the Cluster\Timestamp, the resulting frame timestamp **SHOULD NOT** be negative.

When a CodecDelay Element is set, its value **MUST** be substracted from each Block timestamp of that track. To get the timestamp in nanoseconds of the first frame in a Block or SimpleBlock, the formula becomes:

```
( ( Cluster\Timestamp + ( block timestamp * TrackTimestampScale ) ) *
  TimestampScale ) - CodecDelay
```

The resulting frame timestamp **SHOULD NOT** be negative.

During playback, when a frame has a negative timestamp, the content **MUST** be decoded by the decoder but not played to the user.

10.3. TimestampScale Rounding

The default Track Tick duration is one millisecond.

The TimestampScale is a floating value, which is usually 1.0. But when it's not, the multiplied Block Timestamp is a floating values in nanoseconds. The Matroska Reader **SHOULD** use the nearest rounding value in nanosecond to get the proper nanosecond timestamp of a Block. This allows some clever TimestampScale values to have more refined timestampt precision per frame.

11. Language Codes

Matroska from version 1 through 3 uses language codes that can be either the 3 letters bibliographic ISO-639-2 form [ISO639-2] (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 [ISO639-2] or [BCP47] **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 2 octets country-codes as in Internet domains [IANADomains] based on [ISO3166-1] alpha-2 codes.

12. 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.

13. Image Presentation

13.1. 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.

Cropping has to be performed before resizing and the display dimensions given by DisplayWidth, DisplayHeight and DisplayUnit apply to the already cropped image.

13.2. Rotation

The ProjectionPoseRoll Element (see Section 5.1.4.1.31.45) can be used to indicate that the image from the associated video track **SHOULD** be rotated for presentation. For instance, the following representation of the Projection Element Section 5.1.4.1.31.40) and the ProjectionPoseRoll Element represents a video track where the image **SHOULD** be presentation with a 90 degree counter-clockwise rotation.

```
<Projection>
  <ProjectionPoseRoll>90</ProjectionPoseRoll>
</Projection>
```

Figure 11: Rotation example.

14. 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 Unknown Elements in Section 7.

15. File Extensions

The file extensions for Matroska files are usually as follows:

*".mkv" for files containing video tracks

*".mka" for files containing audio tracks with no video tracks

*".mk3d" for files containing some stereoscopic video tracks

16. 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.

16.1. Segment Position Exception

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

16.2. 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'.

17. Linked Segments

Matroska provides several methods to link two or more Segment Elements together to create a Linked Segment. A Linked Segment is a set of multiple Segments linked together into a single presentation by using Hard Linking or Medium Linking.

All Segments within a Linked Segment **MUST** have a SegmentUID.

All Segments within a Linked Segment **SHOULD** be stored within the same directory or be accessible quickly based on their SegmentUID in order to have seamless transition between segments.

All Segments within a Linked Segment **MAY** set a SegmentFamily with a common value to make it easier for a Matroska Player to know which Segments are meant to be played together.

The SegmentFilename, PrevFilename and NextFilename elements **MAY** also give hints on the original filenames that were used when the Segment links were created, in case some SegmentUID are damaged.

17.1. Hard Linking

Hard Linking, also called splitting, is the process of creating a Linked Segment by linking multiple Segment Elements using the NextUID and PrevUID Elements.

All Segments within a Hard Linked Segment **MUST** use the same Tracks list and TimestampScale.

Within a Linked Segment, the timestamps of Block and SimpleBlock **MUST** follow consecutively the timestamps of Block and SimpleBlock from the previous Segment in linking order.

With Hard Linking, the chapters of any Segment within the Linked Segment **MUST** only reference the current Segment. The NextUID and PrevUID reference the respective SegmentUID values of the next and previous Segments.

The first Segment of a Linked Segment **MUST NOT** have a PrevUID Element. The last Segment of a Linked Segment **MUST NOT** have a NextUID Element.

For each node of the chain of Segments of a Linked Segment at least one Segment **MUST** reference the other Segment of the node.

In a chain of Segments of a Linked Segment the NextUID always takes precedence over the PrevUID. So if SegmentA has a NextUID to SegmentB and SegmentB has a PrevUID to SegmentC, the link to use is NextUID between SegmentA and SegmentB, SegmentC is not part of the Linked Segment.

If SegmentB has a PrevUID to SegmentA but SegmentA has no NextUID, then the Matroska Player **MAY** consider these two Segments linked as SegmentA followed by SegmentB.

As an example, three 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		Invalid	

file name	SegmentUID	PrevUID	NextUID
	71000c23cd310998 53fbc94dd984a5dd		a77b3598941cb803 eac0fcdafe44fac9
middle.mkv	a77b3598941cb803 eac0fcdafe44fac9	71000c23cd310998 53fbc94dd984a5dd	6c92285fa6d3e827 b198d120ea3ac674
end.mkv	6c92285fa6d3e827 b198d120ea3ac674	a77b3598941cb803 eac0fcdafe44fac9	Invalid

Table 42: Usual Hard Linking UIDs

An other example where only the NextUID Element is used:

file name	SegmentUID	PrevUID	NextUID
start.mkv	71000c23cd310998 53fbc94dd984a5dd	Invalid	a77b3598941cb803 eac0fcdafe44fac9
middle.mkv	a77b3598941cb803 eac0fcdafe44fac9	n/a	6c92285fa6d3e827 b198d120ea3ac674
end.mkv	6c92285fa6d3e827 b198d120ea3ac674	n/a	Invalid

Table 43: Hard Linking without PrevUID

An example where only the PrevUID Element is used:

file name	SegmentUID	PrevUID	NextUID
start.mkv	71000c23cd310998 53fbc94dd984a5dd	Invalid	n/a
middle.mkv	a77b3598941cb803 eac0fcdafe44fac9	71000c23cd310998 53fbc94dd984a5dd	n/a
end.mkv	6c92285fa6d3e827 b198d120ea3ac674	a77b3598941cb803 eac0fcdafe44fac9	Invalid

Table 44: Hard Linking without NextUID

In this example only the middle.mkv is using the PrevUID and NextUID Elements:

file name	SegmentUID	PrevUID	NextUID
start.mkv	71000c23cd310998 53fbc94dd984a5dd	Invalid	n/a
middle.mkv	a77b3598941cb803 eac0fcdafe44fac9	71000c23cd310998 53fbc94dd984a5dd	6c92285fa6d3e827 b198d120ea3ac674
end.mkv	6c92285fa6d3e827 b198d120ea3ac674	n/a	Invalid

Table 45: Hard Linking with mixed UID links

17.2. Medium Linking

Medium Linking creates relationships between Segments using Ordered Chapters (Section 20.1.3) and the ChapterSegmentUID Element. A Chapter 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 as a single presentation.

The ChapterSegmentUID represents the Segment that holds the content to play in place of the Linked Chapter. The ChapterSegmentUID **MUST NOT** be the SegmentUID of its own Segment.

There are 2 ways to use a chapter link: * Linked-Duration linking, * Linked-Edition linking

17.2.1. Linked-Duration

A Matroska Player **MUST** play the content of the linked Segment from the ChapterTimeStart until ChapterTimeEnd timestamp in place of the Linked Chapter.

ChapterTimeStart and ChapterTimeEnd represent timestamps in the Linked Segment matching the value of ChapterSegmentUID. Their values **MUST** be in the range of the linked Segment duration.

The ChapterTimeEnd value **MUST** be set when using linked-duration chapter linking. ChapterSegmentEditionUID **MUST NOT** be set.

17.2.2. Linked-Edition

A Matroska Player **MUST** play the whole linked Edition of the linked Segment in place of the Linked Chapter.

ChapterSegmentEditionUID represents a valid Edition from the Linked Segment matching the value of ChapterSegmentUID.

When using linked-edition chapter linking. ChapterTimeEnd is **OPTIONAL**.

18. Track Flags

18.1. Default flag

The "default track" flag is a hint for a Matroska Player indicating that a given track **SHOULD** be eligible to be automatically selected as the default track for a given language. If no tracks in a given language have the default track flag set, then all tracks in that language are eligible for automatic selection. This can be used to indicate that a track provides "regular service" suitable for users with default settings, as opposed to specialized services, such as commentary, hearing-impaired captions, or descriptive audio.

The Matroska Player **MAY** override the "default track" flag for any reason, including user preferences to prefer tracks providing accessibility services.

18.2. Forced flag

The "forced" flag tells the Matroska Player that it **SHOULD** display this subtitle track, even if user preferences usually would not call for any subtitles to be displayed alongside the current selected audio track. This can be used to indicate that a track contains translations of onscreen text, or of dialogue spoken in a different language than the track's primary one.

18.3. Hearing-impaired flag

The "hearing impaired" flag tells the Matroska Player that it **SHOULD** prefer this track when selecting a default track for a hearing-impaired user, and that it **MAY** prefer to select a different track when selecting a default track for a non-hearing-impaired user.

18.4. Visual-impaired flag

The "visual impaired" flag tells the Matroska Player that it **SHOULD** prefer this track when selecting a default track for a visually-impaired user, and that it **MAY** prefer to select a different track when selecting a default track for a non-visually-impaired user.

18.5. Descriptions flag

The "descriptions" flag tells the Matroska Player that this track is suitable to play via a text-to-speech system for a visually-impaired

user, and that it **SHOULD NOT** automatically select this track when selecting a default track for a non-visually-impaired user.

18.6. Original flag

The "original" flag tells the Matroska Player that this track is in the original language, and that it **SHOULD** prefer it if configured to prefer original-language tracks of this track's type.

18.7. Commentary flag

The "commentary" flag tells the Matroska Player that this track contains commentary on the content.

18.8. 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 timestamps, 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.

18.9. 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.

18.10. 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 colors 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 18.8 on how TrackOperation works.

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 were 0: mono, 1: right eye, 2: left eye, 3: both eyes.

19. Default track selection

This section provides some example sets of Tracks and hypothetical user settings, along with indications of which ones a similarly-configured Matroska Player **SHOULD** automatically select for playback by default in such a situation. A player **MAY** provide additional settings with more detailed controls for more nuanced scenarios. These examples are provided as guidelines to illustrate the intended usages of the various supported Track flags, and their expected behaviors.

Track names are shown in English for illustrative purposes; actual files may have titles in the language of each track, or provide titles in multiple languages.

19.1. Audio Selection

Example track set:

No.	Type	Lang	Layout	Original	Default	Other flags	Name
1	Video	und	N/A	N/A	N/A	None	
2	Audio	eng	5.1	1	1	None	
3	Audio	eng	2.0	1	1	None	
4	Audio	eng	2.0	1	0	Visual-impaired	Descriptive audio
5	Audio	esp	5.1	0	1	None	
6	Audio	esp	2.0	0	0	Visual-impaired	Descriptive audio
7	Audio	eng	2.0	1	0	Commentary	Director's Commentary
8	Audio	eng	2.0	1	0	None	Karaoke

Table 46: Audio Tracks for default selection

Here we have a file with 7 audio tracks, of which 5 are in English and 2 are in Spanish.

The English tracks all have the Original flag, indicating that English is the original content language.

Generally the player will first consider the track languages: if the player has an option to prefer original-language audio and the user has enabled it, then it should prefer one of the Original-flagged tracks. If configured to specifically prefer audio tracks in English or Spanish, the player should select one of the tracks in the corresponding language. The player may also wish to prefer an Original-flagged track if no tracks matching any of the user's explicitly-preferred languages are available.

Two of the tracks have the Visual-impaired flag. If the player has been configured to prefer such tracks, it should select one; otherwise, it should avoid them if possible.

If selecting an English track, when other settings have left multiple possible options, it may be useful to exclude the tracks that lack the Default flag: here, one provides descriptive service for the visually impaired (which has its own flag and may be automatically selected by user configuration, but is unsuitable for users with default-configured players), one is a commentary track (which has its own flag, which the player may or may not have specialized handling for), and the last contains karaoke versions of the music that plays during the film, which is an unusual specialized audio service that Matroska has no built-in support for indicating, so it's indicated in the track name instead. By not setting the Default flag on these specialized tracks, the file's author hints that they should not be automatically selected by a default-configured player.

Having narrowed its choices down, our example player now may have to select between tracks 2 and 3. The only difference between these tracks is their channel layouts: 2 is 5.1 surround, while 3 is stereo. If the player is aware that the output device is a pair of headphones or stereo speakers, it may wish to prefer the stereo mix automatically. On the other hand, if it knows that the device is a surround system, it may wish to prefer the surround mix.

If the player finishes analyzing all of the available audio tracks and finds that multiple seem equally and maximally preferable, it **SHOULD** default to the first of the group.

19.2. Subtitle selection

Example track set:

No.	Type	Lang	Original	Default	Forced	Other flags	Name
1	Video	und	N/A	N/A	N/A	None	
2	Audio	fra	1	1	N/A	None	
3	Audio	por	0	1	N/A	None	
4	Subtitles	fra	1	1	0	None	
5	Subtitles	fra	1	0	0	Hearing-impaired	Captions for the hearing-impaired
6	Subtitles	por	0	1	0	None	
7	Subtitles	por	0	0	1	None	Signs
8	Subtitles	por	0	0	0	Hearing-impaired	SDH

Table 47: Subtitle Tracks for default selection

Here we have 2 audio tracks and 5 subtitle tracks. As we can see, French is the original language.

We'll start by discussing the case where the user prefers French (or Original-language) audio (or has explicitly selected the French audio track), and also prefers French subtitles.

In this case, if the player isn't configured to display captions when the audio matches their preferred subtitle languages, the player doesn't need to select a subtitle track at all.

If the user *has* indicated that they want captions to be displayed, the selection simply comes down to whether Hearing-impaired subtitles are preferred.

The situation for a user who prefers Portuguese subtitles starts out somewhat analogous. If they select the original French audio (either by explicit audio language preference, preference for Original-language tracks, or by explicitly selecting that track), then the selection once again comes down to the hearing-impaired preference.

However, the case where the Portuguese audio track is selected has an important catch: a Forced track in Portuguese is present. This may contain translations of onscreen text from the video track, or of portions of the audio that are not translated (music, for instance). This means that even if the user's preferences wouldn't normally call for captions here, the Forced track should be selected nonetheless, rather than selecting no track at all. On the other hand, if the user's preferences *do* call for captions, the non-Forced tracks should be preferred, as the Forced track will not contain captioning for the dialogue.

20. Chapters

The Matroska Chapters system can have multiple Editions and each Edition can consist of Simple Chapters where a chapter start time is used as marker in the timeline only. An Edition can be more complex with Ordered Chapters where a chapter end time stamp is additionally used or much more complex with Linked Chapters. The Matroska Chapters system can also have a menu structure, borrowed from the DVD menu system, or have it's own Native Matroska menu structure.

20.1. EditionEntry

The EditionEntry is also called an Edition. An Edition contains a set of Edition flags and **MUST** contain at least one ChapterAtom Element. Chapters are always inside an Edition (or a Chapter itself part of an Edition). Multiple Editions are allowed. Some of these Editions **MAY** be ordered and others not.

20.1.1. EditionFlagDefault

Only one Edition **SHOULD** have an EditionFlagDefault flag set to true.

20.1.2. Default Edition

The Default Edition is the Edition that a Matroska Player **SHOULD** use for playback by default.

The first Edition with the EditionFlagDefault flag set to true is the Default Edition.

When all EditionFlagDefault flags are set to false, then the first Edition is the Default Edition.

Edition	FlagDefault	Default Edition
Edition 1	true	X
Edition 2	true	
Edition 3	true	

Table 48: Default edition, all default

Edition	FlagDefault	Default Edition
Edition 1	false	X
Edition 2	false	
Edition 3	false	

Table 49: Default edition, no default

Edition	FlagDefault	Default Edition
Edition 1	false	
Edition 2	true	X
Edition 3	false	

Table 50: Default edition, with default

20.1.3. 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 timestamp marked in the ChapterTimeStart Element to the timestamp marked in to ChapterTimeEnd Element.

If the EditionFlagOrdered Flag evaluates to "0", 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. In that case these elements are informational only.

The following list shows the different Chapter elements only found in Ordered Chapters.

Ordered Chapter elements
ChapterAtom/ChapterSegmentUID
ChapterAtom/ChapterSegmentEditionUID

Ordered Chapter elements
ChapterAtom/ChapterTrack
ChapterAtom/ChapProcess
Info/ChapterTranslate
TrackEntry/TrackTranslate

Table 51: elements only found in ordered chapters

Furthermore there are other EBML Elements which could be used if the EditionFlagOrdered evaluates to "1".

20.1.3.1. Ordered-Edition and Matroska Segment-Linking

*Hard Linking: Ordered-Chapters supersedes the Hard Linking.

*Medium Linking: Ordered Chapters are used in a normal way and can be combined with the ChapterSegmentUID element which establishes a link to another Segment.

See Section 17 on the Linked Segments for more information about Hard Linking and Medium Linking.

20.2. ChapterAtom

The ChapterAtom is also called a Chapter.

20.2.1. ChapterTimeStart

The timestamp of the start of Chapter with nanosecond accuracy, not scaled by TimestampScale. For Simple Chapters this is the position of the chapter markers in the timeline.

20.2.2. ChapterTimeEnd

The timestamp of the end of Chapter with nanosecond accuracy, not scaled by TimestampScale. The timestamp defined by the ChapterTimeEnd is not part of the Chapter. A Matroska Player calculates the duration of this Chapter using the difference between the ChapterTimeEnd and ChapterTimeStart. The end timestamp **MUST** be greater than or equal to the start timestamp.

When the ChapterTimeEnd timestamp is equal to the ChapterTimeStart timestamp, the timestamps is included in the Chapter. It can be useful to put markers in a file or add chapter commands with ordered chapter commands without having to play anything; see Section 5.1.7.1.4.14.

Chapter	Start timestamp	End timestamp	Duration
Chapter 1	0	1000000000	1000000000

Chapter	Start timestamp	End timestamp	Duration
Chapter 2	1000000000	5000000000	4000000000
Chapter 3	6000000000	6000000000	0
Chapter 4	9000000000	8000000000	Invalid (-1000000000)

Table 52: ChapterTimeEnd usage possibilities

20.2.3. Nested Chapters

A ChapterAtom element can contain other ChapterAtom elements. That element is a Parent Chapter and the ChapterAtom elements it contains are Nested Chapters.

Nested Chapters can be useful to tag small parts of a Segment that already have tags or add Chapter Codec commands on smaller parts of a Segment that already have Chapter Codec commands.

The ChapterTimeStart of a Nested Chapter **MUST** be greater than or equal to the ChapterTimeStart its Parent Chapter.

If the Parent Chapter of a Nested Chapter has a ChapterTimeEnd, the ChapterTimeStart of that Nested Chapter **MUST** be smaller than or equal to the ChapterTimeEnd of the Parent Chapter.

20.2.4. Nested Chapters in Ordered Chapters

The ChapterTimeEnd of the lowest level of Nested Chapters **MUST** be set for Ordered Chapters.

When used with Ordered Chapters, the ChapterTimeEnd value of a Parent Chapter is useless for playback as the proper playback sections are described in its Nested Chapters. The ChapterTimeEnd **SHOULD NOT** be set in Parent Chapters and **MUST** be ignored for playback.

20.2.5. ChapterFlagHidden

Each Chapter ChapterFlagHidden flag works independently from parent chapters. A Nested Chapter with a ChapterFlagHidden that evaluates to "0" remains visible in the user interface even if the Parent Chapter ChapterFlagHidden flag is set to "1".

Chapter + Nested Chapter	ChapterFlagHidden	visible
Chapter 1	0	yes
Nested Chapter 1.1	0	yes
Nested Chapter 1.2	1	no
Chapter 2	1	no
Nested Chapter 2.1	0	yes
Nested Chapter 2.2	1	no

Table 53: ChapterFlagHidden nested visibility

20.3. 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).

The menu system, as well as Chapter Codecs in general, can do actions on the Matroska Player like jumping to another Chapter or Edition, selecting different tracks and possibly more. The scope of all the possibilities of Chapter Codecs is not covered in this document as it depends on the Chapter Codec features and its integration in a Matroska Player.

20.4. 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:

Value	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

Table 54: ChapterPhysicalEquiv meaning per track type

20.5. Chapter Examples

20.5.1. 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.

*00000ms - 05000ms : Intro

*05000ms - 25000ms : Before the crime

*25000ms - 27500ms : The crime

*27500ms - 38000ms : The killer arrested

*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>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>2311527</ChapterUID>
      <ChapterTimeStart>5000000000</ChapterTimeStart>
      <ChapterTimeEnd>25000000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Before the crime</ChapString>
      </ChapterDisplay>
      <ChapterDisplay>
        <ChapString>Avant le crime</ChapString>
        <ChapLanguage>fra</ChapLanguage>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>3430008</ChapterUID>
      <ChapterTimeStart>25000000000</ChapterTimeStart>
      <ChapterTimeEnd>27500000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>The crime</ChapString>
      </ChapterDisplay>
      <ChapterDisplay>
        <ChapString>Le crime</ChapString>
        <ChapLanguage>fra</ChapLanguage>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>4548489</ChapterUID>
      <ChapterTimeStart>27500000000</ChapterTimeStart>
      <ChapterTimeEnd>38000000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>After the crime</ChapString>
      </ChapterDisplay>
      <ChapterDisplay>
        <ChapString>Après le crime</ChapString>
        <ChapLanguage>fra</ChapLanguage>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>5666960</ChapterUID>
```

```
      <ChapterTimeStart>38000000000</ChapterTimeStart>
      <ChapterTimeEnd>43000000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Credits</ChapString>
      </ChapterDisplay>
      <ChapterDisplay>
        <ChapString>Générique</ChapString>
        <ChapLanguage>fra</ChapLanguage>
      </ChapterDisplay>
    </ChapterAtom>
  </EditionEntry>
</Chapters>
```

Figure 12: Basic Chapters Example.

20.5.2. Example 2 : nested chapters

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

20.5.2.1. The Micronauts "Bleep To Bleep"

*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)

*12:30 - 19:38 : Bleeper_O+2

*19:40 - 22:20 : Baby wants to bleep (pt.4)

*22:22 - 25:18 : Bleep to bleep

*25:20 - 33:35 : Baby wants to bleep (k)

*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>
      </ChapterDisplay>
      <ChapterAtom>
        <ChapterUID>2</ChapterUID>
        <ChapterTimeStart>0</ChapterTimeStart>
        <ChapterTimeEnd>278000000</ChapterTimeEnd>
        <ChapterDisplay>
          <ChapString>Baby wants to bleep (pt.1)</ChapString>
        </ChapterDisplay>
      </ChapterAtom>
      <ChapterAtom>
        <ChapterUID>3</ChapterUID>
        <ChapterTimeStart>278000000</ChapterTimeStart>
        <ChapterTimeEnd>432000000</ChapterTimeEnd>
        <ChapterDisplay>
          <ChapString>Baby wants to rock</ChapString>
        </ChapterDisplay>
      </ChapterAtom>
      <ChapterAtom>
        <ChapterUID>4</ChapterUID>
        <ChapterTimeStart>432000000</ChapterTimeStart>
        <ChapterTimeEnd>633000000</ChapterTimeEnd>
        <ChapterDisplay>
          <ChapString>Baby wants to bleep (pt.2)</ChapString>
        </ChapterDisplay>
      </ChapterAtom>
      <ChapterAtom>
        <ChapterUID>5</ChapterUID>
        <ChapterTimeStart>633000000</ChapterTimeStart>
        <ChapterTimeEnd>748000000</ChapterTimeEnd>
        <ChapterDisplay>
          <ChapString>Baby wants to bleep (pt.3)</ChapString>
        </ChapterDisplay>
      </ChapterAtom>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>6</ChapterUID>
      <ChapterTimeStart>750000000</ChapterTimeStart>
      <ChapterTimeEnd>1178500000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Bleeper_O+2</ChapString>
```

```
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>7</ChapterUID>
      <ChapterTimeStart>1180500000</ChapterTimeStart>
      <ChapterTimeEnd>1340000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Baby wants to bleep (pt.4)</ChapString>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>8</ChapterUID>
      <ChapterTimeStart>1342000000</ChapterTimeStart>
      <ChapterTimeEnd>1518000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Bleep to bleep</ChapString>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>9</ChapterUID>
      <ChapterTimeStart>1520000000</ChapterTimeStart>
      <ChapterTimeEnd>2015000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Baby wants to bleep (k)</ChapString>
      </ChapterDisplay>
    </ChapterAtom>
    <ChapterAtom>
      <ChapterUID>10</ChapterUID>
      <ChapterTimeStart>2017000000</ChapterTimeStart>
      <ChapterTimeEnd>2668000000</ChapterTimeEnd>
      <ChapterDisplay>
        <ChapString>Bleeper</ChapString>
      </ChapterDisplay>
    </ChapterAtom>
  </EditionEntry>
</Chapters>
```

Figure 13: Nested Chapters Example.

21. Attachments

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.

21.1. 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

Table 55: Cover Art Filenames

21.2. Font files

Font files **MAY** be added to a Matroska file as Attachments so that the font file may be used to display an associated subtitle track. This allows the presentation of a Matroska file to be consistent in

various environments where the needed fonts might not be available on the local system.

Depending on the font format in question, each font file can contain multiple font variants. Each font variant has a name which will be referred to as Font Name from now on. This Font Name can be different than the Attachment's FileName, even when disregarding the extension. In order to select a font for display, a Matroska player **SHOULD** consider both the Font Name and the base name of the Attachment's FileName, preferring the former when there are multiple matches.

Subtitle codecs, such as SubStation Alpha (SSA/ASS), usually refer to a font by its Font Name, not by its filename. If none of the Attachments are a match for the Font Name, the Matroska player **SHOULD** attempt to find a system font whose Font Name matches the one used in the subtitle track.

Since loading fonts temporarily can take a while, a Matroska player usually loads or installs all the fonts found in attachments so they are ready to be used during playback. Failure to use the font attachment might result in incorrect rendering of the subtitles.

If a selected subtitle track has some AttachmentLink elements, the player **MAY** use only these fonts.

A Matroska player **SHOULD** handle the official font MIME types from [RFC8081] when the system can handle the type: * font/sfnt: Generic SFNT Font Type, * font/ttf: TTF Font Type, * font/otf: OpenType Layout (OTF) Font Type, * font/collection: Collection Font Type, * font/woff: WOFF 1.0, * font/woff2: WOFF 2.0.

Fonts in Matroska existed long before [RFC8081]. A few unofficial MIME types for fonts were used in existing files. Therefore it is **RECOMMENDED** for a Matroska player to support the following legacy MIME types for font attachments:

*application/x-truetype-font: Truetype fonts, equivalent to font/ttf and sometimes font/otf,

*application/x-font-ttf: TTF fonts, equivalent to font/ttf,

*application/vnd.ms-opentype: OpenType Layout fonts, equivalent to font/otf

*application/font-sfnt: Generic SFNT Font Type, equivalent to font/sfnt

*application/font-woff: WOFF 1.0, equivalent to font/woff

There may also be some font attachments with the application/octet-stream MIME type. In that case the Matroska player **MAY** try to guess the font type by checking the file extension of the AttachedFile\FileName string. Common file extensions for fonts are: * .ttf for Truetype fonts, equivalent to font/ttf, * .otf for OpenType Layout fonts, equivalent to font/otf, * .ttc for Collection fonts, equivalent to font/collection The file extension check **MUST** be case insensitive.

Matroska writers **SHOULD** use a valid font MIME type from [RFC8081] in the AttachedFile\FileMimeType of the font attachment. They **MAY** use the MIME types found in older files when compatibility with older players is necessary.

22. Cues

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.

22.1. Recommendations

The following recommendations are provided to optimize Matroska performance.

*Unless Matroska is used as a live stream, it **SHOULD** contain a Cues Element.

*For each video track, each keyframe **SHOULD** be referenced by a CuePoint Element.

*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.

*For each subtitle track present, each subtitle frame **SHOULD** be referenced by a CuePoint Element with a CueDuration Element.

*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.

* 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.

* If a CuePoint Element references Cluster Element that includes a CodecState Element, then that CuePoint Element **MUST** use a CueCodecState Element.

* CuePoint Elements **SHOULD** be numerically sorted in storage order by the value of the CueTime Element.

23. Matroska Streaming

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

23.1. 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.

23.2. 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.

24. Security Considerations

Matroska inherits security considerations from EBML. Attacks on a Matroska Reader could include: - 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. - A Matroska Attachment with an inaccurate mime-type.

25. IANA Considerations

25.1. Matroska Element IDs Registry

This document creates a new IANA registry called the "Matroska Element IDs" registry.

To register a new Element ID in this registry, one needs an Element ID, a Change Controller (IESG or email of registrant) and an optional Reference to a document describing the Element ID.

Element IDs are described in Section 5 of [RFC8794]. Element IDs are encoded using the VINT mechanism described in Section 4 of [RFC8794] and can be between one and five octets long. Five-octet-long Element IDs are possible only if declared in the EBML header.

One-octet Element IDs **MUST** be between 0x80 and 0xFE. These items are valuable because they are short, and they need to be used for commonly repeated elements. Element IDs are to be allocated within this range according to the "RFC Required" policy [RFC8126].

The following one-octet Element ID is RESERVED: 0xFF.

Values in the one-octet range of 0x00 to 0x7F are not valid for use as an Element ID.

Two-octet Element IDs **MUST** be between 0x407F and 0x7FFE. Element IDs are to be allocated within this range according to the "Specification Required" policy [RFC8126].

The following two-octet Element ID is RESERVED: 0x7FFF.

Values in the two-octet ranges of 0x0000 to 0x4000 and 0x8000 to 0xFFFF are not valid for use as an Element ID.

Three-octet Element IDs **MUST** be between 0x203FFF and 0x3FFFFE. Element IDs are to be allocated within this range according to the "First Come First Served" policy [RFC8126].

The following three-octet Element ID is RESERVED: 0x3FFFFF.

Values in the three-octet ranges of 0x000000 to 0x200000 and 0x400000 to 0xFFFFFF are not valid for use as an Element ID.

Four-octet Element IDs **MUST** be between 0x101FFFFF and 0x1FFFFFFE. Four-octet Element IDs are somewhat special in that they are useful for resynchronizing to major structures in the event of data corruption or loss. As such, four-octet Element IDs are split into two categories. Four-octet Element IDs whose lower three octets (as encoded) would make printable 7-bit ASCII values (0x20 to 0x7E, inclusive) **MUST** be allocated by the "Specification Required" policy. Sequential allocation of values is not required: specifications **SHOULD** include a specific request and are encouraged to do early allocations.

To be clear about the above category: four-octet Element IDs always start with hex 0x10 to 0x1F, and that octet may be chosen so that the entire VINT has some desirable property, such as a specific CRC. The other three octets, when ALL having values between 0x20 (32, ASCII Space) and 0x7E (126, ASCII "~"), fall into this category.

Other four-octet Element IDs may be allocated by the "First Come First Served" policy.

The following four-octet Element ID is RESERVED: 0x1FFFFFFF.

Values in the four-octet ranges of 0x00000000 to 0x10000000 and 0x20000000 to 0xFFFFFFFF are not valid for use as an Element ID.

Five-octet Element IDs (values from 0x080FFFFFFF to 0x0FFFFFFFFE) are RESERVED according to the "Experimental Use" policy [RFC8126]:

they may be used by anyone at any time, but there is no coordination.

EBML IDs defined for the EBML Header -- as defined in Section 17.1 of [RFC8794] -- **MUST NOT** be used as Matroska Element IDs.

Matroska Element IDs Values found in this document are assigned as initial values as follows:

Element ID	Element Name	Reference
0x1043A770	Chapters	Described in Section 5.1.7
0x114D9B74	SeekHead	Described in Section 5.1.1
0x1254C367	Tags	Described in Section 5.1.8
0x1549A966	Info	Described in Section 5.1.2
0x1654AE6B	Tracks	Described in Section 5.1.4
0x18538067	Segment	Described in Section 5.1
0x1941A469	Attachments	Described in Section 5.1.6
0x1C53BB6B	Cues	Described in Section 5.1.5
0x1F43B675	Cluster	Described in Section 5.1.3
0x22B59C	Language	Described in Section 5.1.4.1.21
0x22B59D	LanguageIETF	Described in Section 5.1.4.1.22
0x23314F	TrackTimestampScale	Described in Section 5.1.4.1.17
0x234E7A	DefaultDecodedFieldDuration	Described in Section 5.1.4.1.16
0x2383E3	FrameRate	Described in Section 26.24
0x23E383	DefaultDuration	Described in Section 5.1.4.1.15
0x258688	CodecName	Described in Section 5.1.4.1.25
0x26B240	CodecDownloadURL	Described in Section 26.19
0x2AD7B1	TimestampScale	Described in Section 5.1.2.9
0x2EB524	UncompressedFourCC	Described in Section 5.1.4.1.31.14
0x2FB523	GammaValue	Described in Section 26.23
0x3A9697	CodecSettings	Described in Section 26.17
0x3B4040	CodecInfoURL	Described in Section 26.18
0x3C83AB	PrevFilename	Described in Section 5.1.2.4
0x3CB923	PrevUID	Described in Section 5.1.2.3
0x3E83BB	NextFilename	Described in Section 5.1.2.6
0x3EB923	NextUID	Described in Section 5.1.2.5
0x41A4	BlockAddIDName	Described in Section 5.1.4.1.19.2
0x41E4	BlockAdditionMapping	Described in Section 5.1.4.1.19

Element ID	Element Name	Reference
0x41E7	BlockAddIDType	Described in Section 5.1.4.1.19.3
0x41ED	BlockAddIDExtraData	Described in Section 5.1.4.1.19.4
0x41F0	BlockAddIDValue	Described in Section 5.1.4.1.19.1
0x4254	ContentCompAlgo	Described in Section 5.1.4.1.34.6
0x4255	ContentCompSettings	Described in Section 5.1.4.1.34.7
0x437C	ChapLanguage	Described in Section 5.1.7.1.4.11
0x437D	ChapLanguageIETF	Described in Section 5.1.7.1.4.12
0x437E	ChapCountry	Described in Section 5.1.7.1.4.13
0x4444	SegmentFamily	Described in Section 5.1.2.7
0x4461	DateUTC	Described in Section 5.1.2.11
0x447A	TagLanguage	Described in Section 5.1.8.1.2.2
0x447B	TagLanguageIETF	Described in Section 5.1.8.1.2.3
0x4484	TagDefault	Described in Section 5.1.8.1.2.4
0x4485	TagBinary	Described in Section 5.1.8.1.2.6
0x4487	TagString	Described in Section 5.1.8.1.2.5
0x4489	Duration	Described in Section 5.1.2.10
0x44B4	TagDefaultBogus	Described in Section 26.41
0x450D	ChapProcessPrivate	Described in Section 5.1.7.1.4.16
0x45A3	TagName	Described in Section 5.1.8.1.2.1
0x45B9	EditionEntry	Described in Section 5.1.7.1
0x45BC	EditionUID	Described in Section 5.1.7.1.1
0x45DB	EditionFlagDefault	Described in Section 5.1.7.1.2
0x45DD	EditionFlagOrdered	Described in Section 5.1.7.1.3
0x465C	FileData	Described in Section 5.1.6.1.4
0x4660	FileMimeType	

Element ID	Element Name	Reference
		Described in Section 5.1.6.1.3
0x4661	FileUsedStartTime	Described in Section 26.39
0x4662	FileUsedEndTime	Described in Section 26.40
0x466E	FileName	Described in Section 5.1.6.1.2
0x4675	FileReferral	Described in Section 26.38
0x467E	FileDescription	Described in Section 5.1.6.1.1
0x46AE	FileUID	Described in Section 5.1.6.1.5
0x47E1	ContentEncAlgo	Described in Section 5.1.4.1.34.9
0x47E2	ContentEncKeyID	Described in Section 5.1.4.1.34.10
0x47E3	ContentSignature	Described in Section 26.31
0x47E4	ContentSigKeyID	Described in Section 26.32
0x47E5	ContentSigAlgo	Described in Section 26.33
0x47E6	ContentSigHashAlgo	Described in Section 26.34
0x47E7	ContentEncAESSettings	Described in Section 5.1.4.1.34.11
0x47E8	AESSettingsCipherMode	Described in Section 5.1.4.1.34.12
0x4D80	MuxingApp	Described in Section 5.1.2.13
0x4DBB	Seek	Described in Section 5.1.1.1
0x5031	ContentEncodingOrder	Described in Section 5.1.4.1.34.2
0x5032	ContentEncodingScope	Described in Section 5.1.4.1.34.3
0x5033	ContentEncodingType	Described in Section 5.1.4.1.34.4
0x5034	ContentCompression	Described in Section 5.1.4.1.34.5
0x5035	ContentEncryption	Described in Section 5.1.4.1.34.8
0x535F	CueRefNumber	Described in Section 26.36
0x536E	Name	Described in Section 5.1.4.1.20
0x5378	CueBlockNumber	Described in Section 5.1.5.1.2.5
0x537F	TrackOffset	Described in Section 26.16
0x53AB	SeekID	Described in Section 5.1.1.1.1
0x53AC	SeekPosition	Described in Section 5.1.1.1.2

Element ID	Element Name	Reference
0x53B8	StereoMode	Described in Section 5.1.4.1.31.3
0x53B9	OldStereoMode	Described in Section 26.21
0x53C0	AlphaMode	Described in Section 5.1.4.1.31.4
0x54AA	PixelCropBottom	Described in Section 5.1.4.1.31.7
0x54B0	DisplayWidth	Described in Section 5.1.4.1.31.11
0x54B2	DisplayUnit	Described in Section 5.1.4.1.31.13
0x54B3	AspectRatioType	Described in Section 26.22
0x54BA	DisplayHeight	Described in Section 5.1.4.1.31.12
0x54BB	PixelCropTop	Described in Section 5.1.4.1.31.8
0x54CC	PixelCropLeft	Described in Section 5.1.4.1.31.9
0x54DD	PixelCropRight	Described in Section 5.1.4.1.31.10
0x55AA	FlagForced	Described in Section 5.1.4.1.6
0x55AB	FlagHearingImpaired	Described in Section 5.1.4.1.7
0x55AC	FlagVisualImpaired	Described in Section 5.1.4.1.8
0x55AD	FlagTextDescriptions	Described in Section 5.1.4.1.9
0x55AE	FlagOriginal	Described in Section 5.1.4.1.10
0x55AF	FlagCommentary	Described in Section 5.1.4.1.11
0x55B0	Colour	Described in Section 5.1.4.1.31.15
0x55B1	MatrixCoefficients	Described in Section 5.1.4.1.31.16
0x55B2	BitsPerChannel	Described in Section 5.1.4.1.31.17
0x55B3	ChromaSubsamplingHorz	Described in Section 5.1.4.1.31.18
0x55B4	ChromaSubsamplingVert	Described in Section 5.1.4.1.31.19
0x55B5	CbSubsamplingHorz	Described in Section 5.1.4.1.31.20
0x55B6	CbSubsamplingVert	Described in Section 5.1.4.1.31.21

Element ID	Element Name	Reference
0x55B7	ChromaSitingHorz	Described in Section 5.1.4.1.31.22
0x55B8	ChromaSitingVert	Described in Section 5.1.4.1.31.23
0x55B9	Range	Described in Section 5.1.4.1.31.24
0x55BA	TransferCharacteristics	Described in Section 5.1.4.1.31.25
0x55BB	Primaries	Described in Section 5.1.4.1.31.26
0x55BC	MaxCLL	Described in Section 5.1.4.1.31.27
0x55BD	MaxFALL	Described in Section 5.1.4.1.31.28
0x55D0	MasteringMetadata	Described in Section 5.1.4.1.31.29
0x55D1	PrimaryRChromaticityX	Described in Section 5.1.4.1.31.30
0x55D2	PrimaryRChromaticityY	Described in Section 5.1.4.1.31.31
0x55D3	PrimaryGChromaticityX	Described in Section 5.1.4.1.31.32
0x55D4	PrimaryGChromaticityY	Described in Section 5.1.4.1.31.33
0x55D5	PrimaryBChromaticityX	Described in Section 5.1.4.1.31.34
0x55D6	PrimaryBChromaticityY	Described in Section 5.1.4.1.31.35
0x55D7	WhitePointChromaticityX	Described in Section 5.1.4.1.31.36
0x55D8	WhitePointChromaticityY	Described in Section 5.1.4.1.31.37
0x55D9	LuminanceMax	Described in Section 5.1.4.1.31.38
0x55DA	LuminanceMin	Described in Section 5.1.4.1.31.39
0x55EE	MaxBlockAdditionID	Described in Section 5.1.4.1.18
0x5654	ChapterStringUID	Described in Section 5.1.7.1.4.2
0x56AA	CodecDelay	Described in Section 5.1.4.1.28
0x56BB	SeekPreRoll	Described in Section 5.1.4.1.29
0x5741	WritingApp	Described in Section 5.1.2.14

Element ID	Element Name	Reference
0x5854	SilentTracks	Described in Section 26.1
0x58D7	SilentTrackNumber	Described in Section 26.2
0x61A7	AttachedFile	Described in Section 5.1.6.1
0x6240	ContentEncoding	Described in Section 5.1.4.1.34.1
0x6264	BitDepth	Described in Section 5.1.4.1.32.4
0x63A2	CodecPrivate	Described in Section 5.1.4.1.24
0x63C0	Targets	Described in Section 5.1.8.1.1
0x63C3	ChapterPhysicalEquiv	Described in Section 5.1.7.1.4.8
0x63C4	TagChapterUID	Described in Section 5.1.8.1.1.5
0x63C5	TagTrackUID	Described in Section 5.1.8.1.1.3
0x63C6	TagAttachmentUID	Described in Section 5.1.8.1.1.6
0x63C9	TagEditionUID	Described in Section 5.1.8.1.1.4
0x63CA	TargetType	Described in Section 5.1.8.1.1.2
0x6624	TrackTranslate	Described in Section 5.1.4.1.30
0x66A5	TrackTranslateTrackID	Described in Section 5.1.4.1.30.1
0x66BF	TrackTranslateCodec	Described in Section 5.1.4.1.30.2
0x66FC	TrackTranslateEditionUID	Described in Section 5.1.4.1.30.3
0x67C8	SimpleTag	Described in Section 5.1.8.1.2
0x68CA	TargetTypeValue	Described in Section 5.1.8.1.1.1
0x6911	ChapProcessCommand	Described in Section 5.1.7.1.4.17
0x6922	ChapProcessTime	Described in Section 5.1.7.1.4.18
0x6924	ChapterTranslate	Described in Section 5.1.2.8
0x6933	ChapProcessData	Described in Section 5.1.7.1.4.19
0x6944	ChapProcess	Described in Section 5.1.7.1.4.14
0x6955	ChapProcessCodecID	Described in Section 5.1.7.1.4.15

Element ID	Element Name	Reference
0x69A5	ChapterTranslateID	Described in Section 5.1.2.8.1
0x69BF	ChapterTranslateCodec	Described in Section 5.1.2.8.2
0x69FC	ChapterTranslateEditionUID	Described in Section 5.1.2.8.3
0x6D80	ContentEncodings	Described in Section 5.1.4.1.34
0x6DE7	MinCache	Described in Section 5.1.4.1.13
0x6DF8	MaxCache	Described in Section 5.1.4.1.14
0x6E67	ChapterSegmentUID	Described in Section 5.1.7.1.4.6
0x6EBC	ChapterSegmentEditionUID	Described in Section 5.1.7.1.4.7
0x6FAB	TrackOverlay	Described in Section 5.1.4.1.27
0x7373	Tag	Described in Section 5.1.8.1
0x7384	SegmentFilename	Described in Section 5.1.2.2
0x73A4	SegmentUID	Described in Section 5.1.2.1
0x73C4	ChapterUID	Described in Section 5.1.7.1.4.1
0x73C5	TrackUID	Described in Section 5.1.4.1.2
0x7446	AttachmentLink	Described in Section 5.1.4.1.26
0x75A1	BlockAdditions	Described in Section 5.1.3.5.2
0x75A2	DiscardPadding	Described in Section 5.1.3.5.7
0x7670	Projection	Described in Section 5.1.4.1.31.40
0x7671	ProjectionType	Described in Section 5.1.4.1.31.41
0x7672	ProjectionPrivate	Described in Section 5.1.4.1.31.42
0x7673	ProjectionPoseYaw	Described in Section 5.1.4.1.31.43
0x7674	ProjectionPosePitch	Described in Section 5.1.4.1.31.44
0x7675	ProjectionPoseRoll	Described in Section 5.1.4.1.31.45
0x78B5	OutputSamplingFrequency	Described in Section 5.1.4.1.32.2
0x7BA9	Title	

Element ID	Element Name	Reference
		Described in Section 5.1.2.12
0x7D7B	ChannelPositions	Described in Section 26.25
0x80	ChapterDisplay	Described in Section 5.1.7.1.4.9
0x83	TrackType	Described in Section 5.1.4.1.3
0x85	ChapString	Described in Section 5.1.7.1.4.10
0x86	CodecID	Described in Section 5.1.4.1.23
0x88	FlagDefault	Described in Section 5.1.4.1.5
0x8E	Slices	Described in Section 26.5
0x91	ChapterTimeStart	Described in Section 5.1.7.1.4.3
0x92	ChapterTimeEnd	Described in Section 5.1.7.1.4.4
0x96	CueRefTime	Described in Section 5.1.5.1.2.8
0x97	CueRefCluster	Described in Section 26.35
0x98	ChapterFlagHidden	Described in Section 5.1.7.1.4.5
0x9A	FlagInterlaced	Described in Section 5.1.4.1.31.1
0x9B	BlockDuration	Described in Section 5.1.3.5.3
0x9C	FlagLacing	Described in Section 5.1.4.1.12
0x9D	FieldOrder	Described in Section 5.1.4.1.31.2
0x9F	Channels	Described in Section 5.1.4.1.32.3
0xA0	BlockGroup	Described in Section 5.1.3.5
0xA1	Block	Described in Section 5.1.3.5.1
0xA2	BlockVirtual	Described in Section 26.3
0xA3	SimpleBlock	Described in Section 5.1.3.4
0xA4	CodecState	Described in Section 5.1.3.5.6
0xA5	BlockAdditional	Described in Section 5.1.3.5.2.3
0xA6	BlockMore	Described in Section 5.1.3.5.2.1
0xA7	Position	Described in Section 5.1.3.2
0xAA	CodecDecodeAll	Described in Section 26.20

Element ID	Element Name	Reference
0xAB	PrevSize	Described in Section 5.1.3.3
0xAE	TrackEntry	Described in Section 5.1.4.1
0xAF	EncryptedBlock	Described in Section 26.15
0xB0	PixelWidth	Described in Section 5.1.4.1.31.5
0xB2	CueDuration	Described in Section 5.1.5.1.2.4
0xB3	CueTime	Described in Section 5.1.5.1.1
0xB5	SamplingFrequency	Described in Section 5.1.4.1.32.1
0xB6	ChapterAtom	Described in Section 5.1.7.1.4
0xB7	CueTrackPositions	Described in Section 5.1.5.1.2
0xB9	FlagEnabled	Described in Section 5.1.4.1.4
0xBA	PixelHeight	Described in Section 5.1.4.1.31.6
0xBB	CuePoint	Described in Section 5.1.5.1
0xC0	TrickTrackUID	Described in Section 26.26
0xC1	TrickTrackSegmentUID	Described in Section 26.27
0xC4	TrickMasterTrackSegmentUID	Described in Section 26.30
0xC6	TrickTrackFlag	Described in Section 26.28
0xC7	TrickMasterTrackUID	Described in Section 26.29
0xC8	ReferenceFrame	Described in Section 26.12
0xC9	ReferenceOffset	Described in Section 26.13
0xCA	ReferenceTimestamp	Described in Section 26.14
0xCB	BlockAdditionID	Described in Section 26.9
0xCC	LaceNumber	Described in Section 26.7
0xCD	FrameNumber	Described in Section 26.8
0xCE	Delay	Described in Section 26.10
0xCF	SliceDuration	Described in Section 26.11
0xD7	TrackNumber	Described in Section 5.1.4.1.1
0xDB	CueReference	Described in Section 5.1.5.1.2.7
0xE0	Video	Described in Section 5.1.4.1.31
0xE1	Audio	Described in Section 5.1.4.1.32
0xE2	TrackOperation	Described in Section 5.1.4.1.33
0xE3	TrackCombinePlanes	Described in Section 5.1.4.1.33.1
0xE4	TrackPlane	

Element ID	Element Name	Reference
		Described in Section 5.1.4.1.33.2
0xE5	TrackPlaneUID	Described in Section 5.1.4.1.33.3
0xE6	TrackPlaneType	Described in Section 5.1.4.1.33.4
0xE7	Timestamp	Described in Section 5.1.3.1
0xE8	TimeSlice	Described in Section 26.6
0xE9	TrackJoinBlocks	Described in Section 5.1.4.1.33.5
0xEA	CueCodecState	Described in Section 5.1.5.1.2.6
0xEB	CueRefCodecState	Described in Section 26.37
0xED	TrackJoinUID	Described in Section 5.1.4.1.33.6
0xEE	BlockAddID	Described in Section 5.1.3.5.2.2
0xF0	CueRelativePosition	Described in Section 5.1.5.1.2.3
0xF1	CueClusterPosition	Described in Section 5.1.5.1.2.2
0xF7	CueTrack	Described in Section 5.1.5.1.2.1
0xFA	ReferencePriority	Described in Section 5.1.3.5.4
0xFB	ReferenceBlock	Described in Section 5.1.3.5.5
0xFD	ReferenceVirtual	Described in Section 26.4

Table 56: IDs and Names for Matroska Element IDs assigned by this document

25.2. Chapter Codec IDs Registry

This document creates a new IANA registry called the "Matroska Chapter Codec IDs" registry. The values correspond to the ChapProcessCodecID value described in Section 5.1.7.1.4.15.

ChapProcessCodecID values of "0" and "1" are RESERVED to the IETF for future use.

25.3. MIME Types

Matroska files and streams are found in three main forms: audio-video files, audio-only and occasionally with stereoscopic video tracks.

The MIME types to use for each type is:

- "video/matroska" for streams containing video tracks

- "audio/matroska" for streams containing audio tracks with no video tracks

- "video/matroska-3d" for streams containing at least a stereoscopic video track

Historically Matroska files and streams have used the following MIME types with a "x-" prefix:

- "video/x-matroska" for streams containing video tracks

- "audio/x-matroska" for streams containing audio tracks with no video tracks

- "video/x-matroska-3d" for streams containing at least a stereoscopic video track

For better compatibility a system **SHOULD** be able to handle both formats. Newer systems **SHOULD NOT** use the historic format and use the format that follows the [RFC6838] format instead.

26. Annex A: Historic Deprecated Elements

As Matroska evolved since 2002 many parts that were considered for use in the format were never used and often incorrectly designed. Many of the elements that were then defined are not found in any known files but were part of public specs. DivX also had a few custom elements that were designed for custom features.

We list these elements that have a known ID that **SHOULD NOT** be reused to avoid colliding with existing files.

26.1. SilentTracks Element

path: \Segment\Cluster\SilentTracks

id: 0x5854

type: master

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.

26.2. SilentTrackNumber Element

path:
\Segment\Cluster\SilentTracks\SilentTrackNumber

id: 0x58D7

type: uinteger

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.

26.3. BlockVirtual Element

path: \Segment\Cluster\BlockGroup\BlockVirtual

id: 0xA2

type: binary

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

26.4. ReferenceVirtual Element

path: \Segment\Cluster\BlockGroup\ReferenceVirtual

id: 0xFD

type: integer

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

26.5. Slices Element

path: \Segment\Cluster\BlockGroup\Slices

id: 0x8E

type: master

documentation: Contains slices description.

26.6. TimeSlice Element

path: \Segment\Cluster\BlockGroup\Slices\TimeSlice

id: 0xE8

type: master

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

26.7. LaceNumber Element

path: \Segment\Cluster\BlockGroup\Slices\TimeSlice\LaceNumber

id: 0xCC

type: uinteger

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.

26.8. FrameNumber Element

path: \Segment\Cluster\BlockGroup\Slices\TimeSlice\FrameNumber

id: 0xCD

type: uinteger

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).

26.9. BlockAdditionID Element

path: \Segment\Cluster\BlockGroup\Slices\TimeSlice\BlockAdditionID

id: 0xCB

type: uinteger

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

26.10. Delay Element

path: \Segment\Cluster\BlockGroup\Slices\TimeSlice\Delay

id: 0xCE

type: uinteger

documentation: The delay to apply to the Element, expressed in Track Ticks; see Section 10.1.

26.11. SliceDuration Element

path: \Segment\Cluster\BlockGroup\Slices\TimeSlice\SliceDuration

id: 0xCF

type: uinteger

documentation: The duration to apply to the Element, expressed in Track Ticks; see Section 10.1.

26.12. ReferenceFrame Element

path: \Segment\Cluster\BlockGroup\ReferenceFrame

id: 0xC8

type: master

documentation: Contains information about the last reference frame. See [DivXTrickTrack].

26.13. ReferenceOffset Element

path: \Segment\Cluster\BlockGroup\ReferenceFrame\ReferenceOffset

id: 0xC9

type: uinteger

documentation: The relative offset, in bytes, from the previous BlockGroup element for this Smooth FF/RW video track to the containing BlockGroup element. See [DivXTrickTrack].

26.14. ReferenceTimestamp Element

path: \Segment\Cluster\BlockGroup\ReferenceFrame\ReferenceTimestamp

id: 0xCA

type: uinteger

documentation: The timestamp of the BlockGroup pointed to by ReferenceOffset, expressed in Track Ticks; see Section 10.1. See [DivXTrickTrack].

26.15. EncryptedBlock Element

path: \Segment\Cluster\EncryptedBlock

id: 0xAF

type:
binary

documentation: Similar to SimpleBlock, see Section 9.3, but the data inside the Block are Transformed (encrypt and/or signed).

26.16. TrackOffset Element

path: \Segment\Tracks\TrackEntry\TrackOffset

id: 0x537F

type: integer

documentation: A value to add to the Block's Timestamp, expressed in Matroska Ticks -- ie in nanoseconds; see Section 10.1. This can be used to adjust the playback offset of a track.

26.17. CodecSettings Element

path: \Segment\Tracks\TrackEntry\CodecSettings

id: 0x3A9697

type: utf-8

documentation: A string describing the encoding setting used.

26.18. CodecInfoURL Element

path: \Segment\Tracks\TrackEntry\CodecInfoURL

id: 0x3B4040

type: string

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

26.19. CodecDownloadURL Element

path: \Segment\Tracks\TrackEntry\CodecDownloadURL

id: 0x26B240

type: string

documentation: A URL to download about the codec used.

26.20. CodecDecodeAll Element

path: \Segment\Tracks\TrackEntry\CodecDecodeAll

id:
0xAA

type: uinteger

documentation: Set to 1 if the codec can decode potentially damaged data.

26.21. OldStereoMode Element

path: \Segment\Tracks\TrackEntry\Video\OldStereoMode

id: 0x53B9

type: uinteger

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

26.22. AspectRatioType Element

path: \Segment\Tracks\TrackEntry\Video\AspectRatioType

id: 0x54B3

type: uinteger

documentation: Specify the possible modifications to the aspect ratio.

26.23. GammaValue Element

path: \Segment\Tracks\TrackEntry\Video\GammaValue

id: 0x2FB523

type: float

documentation: Gamma Value.

26.24. FrameRate Element

path: \Segment\Tracks\TrackEntry\Video\FrameRate

id: 0x2383E3

type: float

documentation: Number of frames per second. This value is Informational only. It is intended for constant frame rate

streams, and **SHOULD NOT** be used for a variable frame rate TrackEntry.

26.25. ChannelPositions Element

path: \Segment\Tracks\TrackEntry\Audio\ChannelPositions

id: 0x7D7B

type: binary

documentation: Table of horizontal angles for each successive channel.

26.26. TrickTrackUID Element

path: \Segment\Tracks\TrackEntry\TrickTrackUID

id: 0xC0

type: uinteger

documentation: The TrackUID of the Smooth FF/RW video in the paired EBML structure corresponding to this video track. See [DivXTrickTrack].

26.27. TrickTrackSegmentUID Element

path: \Segment\Tracks\TrackEntry\TrickTrackSegmentUID

id: 0xC1

type: binary

documentation: The SegmentUID of the Segment containing the track identified by TrickTrackUID. See [DivXTrickTrack].

26.28. TrickTrackFlag Element

path: \Segment\Tracks\TrackEntry\TrickTrackFlag

id: 0xC6

type: uinteger

documentation: Set to 1 if this video track is a Smooth FF/RW track. If set to 1, MasterTrackUID and MasterTrackSegUID should must be present and BlockGroups for this track must contain ReferenceFrame structures. Otherwise, TrickTrackUID and TrickTrackSegUID must be present if this track has a corresponding Smooth FF/RW track. See [DivXTrickTrack].

26.29. TrickMasterTrackUID Element

path: \Segment\Tracks\TrackEntry\TrickMasterTrackUID

id: 0xC7

type: uinteger

documentation: The TrackUID of the video track in the paired EBML structure that corresponds to this Smooth FF/RW track. See [DivXTrickTrack].

26.30. TrickMasterTrackSegmentUID Element

path: \Segment\Tracks\TrackEntry\TrickMasterTrackSegmentUID

id: 0xC4

type: binary

documentation: The SegmentUID of the Segment containing the track identified by MasterTrackUID. See [DivXTrickTrack].

26.31. ContentSignature Element

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentSignature

id: 0x47E3

type: binary

documentation: A cryptographic signature of the contents.

26.32. ContentSigKeyID Element

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\ContentEncryption\ContentSigKeyID

id: 0x47E4

type: binary

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

26.33. ContentSigAlgo Element

path:

\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\Conte
ntEncryption\ContentSigAlgo

id: 0x47E5

type: uinteger

documentation: The algorithm used for the signature.

26.34. ContentSigHashAlgo Element

path:
\Segment\Tracks\TrackEntry\ContentEncodings\ContentEncoding\Conte
ntEncryption\ContentSigHashAlgo

id: 0x47E6

type: uinteger

documentation: The hash algorithm used for the signature.

26.35. CueRefCluster Element

path:
\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefClust
er

id: 0x97

type: uinteger

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

26.36. CueRefNumber Element

path:
\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefNumbe
r

id: 0x535F

type: uinteger

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

26.37. CueRefCodecState Element

path:

\Segment\Cues\CuePoint\CueTrackPositions\CueReference\CueRefCodec
State

id: 0xEB

type: uinteger

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.

26.38. FileReferral Element

path: \Segment\Attachments\AttachedFile\FileReferral

id: 0x4675

type: binary

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

26.39. FileUsedStartTime Element

path: \Segment\Attachments\AttachedFile\FileUsedStartTime

id: 0x4661

type: uinteger

documentation: The timestamp at which this optimized font
attachment comes into context, expressed in Segment Ticks which
is based on TimestampScale. See [DivXWorldFonts].

26.40. FileUsedEndTime Element

path: \Segment\Attachments\AttachedFile\FileUsedEndTime

id: 0x4662

type: uinteger

documentation: The timestamp at which this optimized font
attachment goes out of context, expressed in Segment Ticks which
is based on TimestampScale. See [DivXWorldFonts].

26.41. TagDefaultBogus Element

path: \Segment\Tags\Tag\+SimpleTag\TagDefaultBogus

id: 0x44B4

type:
uinteger

documentation: A variant of the TagDefault element with a bogus Element ID; see Section 5.1.8.1.2.4.

27. Normative References

[BCP47] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", , DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[Blowfish] Schneier, B., "The Blowfish Encryption Algorithm", , 1993, <https://www.schneier.com/academic/blowfish/>.

[BZIP2] Seward, J., "bzip2", , 18 July 1996, <https://sourceware.org/bzip2/>.

[FIPS.197] US National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", , DOI 10.6028/NIST.FIPS.197, 26 November 2001, <https://csrc.nist.gov/publications/detail/fips/197/final>.

[FIPS.46-3] US National Institute of Standards and Technology, "Data Encryption Standard (DES)", , FIPS PUB 46, 25 October 1999, <https://csrc.nist.gov/publications/detail/fips/46/3/archive/1999-10-25>.

[IANADomains] "IANA Root Zone Database", , <https://www.iana.org/domains/root/db>.

[IANALangRegistry] "IANA Language Subtag Registry", , 28 February 2013, <https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry>.

[ISO3166-1] International Organization for Standardization, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country code", , ISO 3166-1:2020, August 2020, <https://www.iso.org/standard/72482.html>.

[ISO639-2] United States Library Of Congress, "Codes for the Representation of Names of Languages", , ISO 639-2:1998,

21 December 2017, <https://www.loc.gov/standards/iso639-2/php/code_list.php>.

[LZO] Tarreau, W., Rodgman, R., and M. Oberhumer, "Lempel–Ziv–Oberhumer compression", , 30 October 2018, <https://www.kernel.org/doc/Documentation/lzo.txt>.

[MatroskaCodec] Lhomme, S., Bunkus, M., and D. Rice, "Media Container Codec Specifications", , Work in Progress, Internet-Draft, draft-ietf-cellar-codec-06, 12 April 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-cellar-codec-06>.

[MatroskaTags] Lhomme, S., Bunkus, M., and D. Rice, "Matroska Media Container Tag Specifications", , Work in Progress, Internet-Draft, draft-ietf-cellar-tags-06, 12 April 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-cellar-tags-06>.

[RFC1851] Karn, P., Metzger, P., and W. Simpson, "The ESP Triple DES Transform", RFC 1851, DOI 10.17487/RFC1851, September 1995, <https://www.rfc-editor.org/info/rfc1851>.

[RFC1950] Deutsch, P. and J-L. Gailly, "ZLIB Compressed Data Format Specification version 3.3", RFC 1950, DOI 10.17487/RFC1950, May 1996, <https://www.rfc-editor.org/info/rfc1950>.

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

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC8081] Lilley, C., "The "font" Top-Level Media Type", RFC 8081, DOI 10.17487/RFC8081, February 2017, <https://www.rfc-editor.org/info/rfc8081>.

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

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

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

[SP.800-38A] US National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: Methods and Techniques", , DOI 10.6028/NIST.SP.800-38A, 1 December 2001, <https://csrc.nist.gov/publications/detail/fips/197/final>.

[Twofish] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., and N. Ferguson, "Twofish: A 128-Bit Block Cipher", , 15 June 1998, <https://www.schneier.com/academic/twofish/>.

[WebVTT] Pieters, S., Pfeiffer, S., Ed., Jägenstedt, P., and I. Hickson, "WebVTT Cue Identifier", , 4 April 2019, <https://www.w3.org/TR/webvtt1/#webvtt-cue-identifier>.

28. Informative References

[AVIFormat] Microsoft, "AVI RIFF File Reference", , 31 May 2018, <https://docs.microsoft.com/en-us/windows/win32/directshow/avi-riff-file-reference>.

[DivXTrickTrack] "DivX Trick Track Extensions", , 14 December 2010, <http://web.archive.org/web/20101222001148/http://labs.divx.com/node/16601>.

[DivXWorldFonts] "DivX World Fonts Extensions", , 14 December 2010, <http://web.archive.org/web/20110214132246/http://labs.divx.com/node/16602>.

[FourCC-RGB] Silicon.dk ApS, "RGB Pixel Format FourCCs", , <https://www.fourcc.org/rgb.php>.

[FourCC-YUV] Silicon.dk ApS, "YUV Pixel Format FourCCs", , <https://www.fourcc.org/yuv.php>.

[MCF] "Media Container Format", , 17 July 2002, <http://mukoli.free.fr/mcf/mcf.html>.

[RFC3533] Pfeiffer, S., "The Ogg Encapsulation Format Version 0", RFC 3533, DOI 10.17487/RFC3533, May 2003, <https://www.rfc-editor.org/info/rfc3533>.

Authors' Addresses

Steve Lhomme

Email: slhomme@matroska.org

Moritz Bunkus

Email: moritz@bunkus.org

Dave Rice

Email: dave@dericed.com