



Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Einleitung

- Containerformat, kein Codec
- Verwendet EBML
- Umfangreiche Anzahl an Informationen und Inhalten möglich
- Nicht nur für Videos
- ist Grundlage für webm
- Selbst gestecktes Ziel: „ Matroska aims to become THE standard of multimedia container formats.“

Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Extensible Binary Meta Language (EBML)

- „Extensible Binary Meta Language“ (früher: Extensible Binary Markup Language)
- Von XML inspiriert
- Sprache, die beschreibt wie Daten in binärform interpretiert werden bzw. vorliegen müssen
- Genauer Inhalt ist nicht festgelegt, wird zu Beginn der Datei definiert
- Meist Audio und/oder Video

EBML

- Element ID coded with an UTF-8 like system :

```
bits, big-endian
1xxx xxxx - Class A IDs (2^7 -1 possible values) (base 0x8X)
01xx xxxx xxxx xxxx - Class B IDs (2^14-1 possible values) (base 0x4X 0xXX)
001x xxxx xxxx xxxx xxxx xxxx - Class C IDs (2^21-1 possible values) (base 0x2X 0xXX 0xXX)
0001 xxxx xxxx xxxx xxxx xxxx - Class D IDs (2^28-1 possible values) (base 0x1X 0xXX 0xXX 0xXX)
```

Some Notes:

- The leading bits of the EBML IDs are used to identify the length of the ID. The number of leading 0's + 1 is the length of the ID in octets. We will refer to the leading bits as the Length Descriptor.
- Any ID where all x's are composed entirely of 1's is a Reserved ID, thus the -1 in the definitions above.
- The Reserved IDs (all x set to 1) are the only IDs that may change the Length Descriptor.

- Data size, in octets, is also coded with an UTF-8 like system :

```
bits, big-endian
1xxx xxxx - value 0 to 2^7-2
01xx xxxx xxxx xxxx - value 0 to 2^14-2
001x xxxx xxxx xxxx xxxx xxxx - value 0 to 2^21-2
0001 xxxx xxxx xxxx xxxx xxxx xxxx - value 0 to 2^28-2
0000 1xxx xxxx xxxx xxxx xxxx xxxx xxxx - value 0 to 2^35-2
0000 01xx xxxx xxxx xxxx xxxx xxxx xxxx xxxx - value 0 to 2^42-2
0000 0001 xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx - value 0 to 2^49-2
0000 0000 xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx - value 0 to 2^56-2
```

Since modern computers do not easily deal with data coded in sizes greater than 64 bits, any larger Element Sizes are left undefined at the moment. Currently, the Element Size coding allows for an Element to grow to 72000 To, i.e. 7×10^{16} octets or 72000 terabytes, which will be sufficient for the time being.

There is only one reserved word for Element Size encoding, which is an Element Size encoded to all 1's. Such a coding indicates that the size of the Element is unknown, which is a special case that we believe will be useful for live streaming purposes. However, avoid using this reserved word unnecessarily, because it makes parsing slower and more difficult to implement.

- Data
 - Integers are stored in their standard big-endian form (no UTF-like encoding), only the size may differ from their usual form (24 or 40 bits for example).
 - The Signed Integer is just the big-endian representation trimmed from some 0x00 and 0xFF where they are not meaningful (sign). For example -2 can be coded as 0xFFFFFFFFFFFFE or 0xFFFE or 0xFE and 5 can be coded 0x0000000000005 or 0x0005 or 0x05.

http://matroska.org/technical/specs/index.html

Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Entstehung

- Erste Bekanntgabe 6.12.2002
- Zweig von MCF wegen Differenzen zwischen Entwicklern
- Geplante Unterbrechung in der Entwicklung und diese Differenzen führten zu Abwandern vieler Entwickler zu Matroska

Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Merkmale

- Es gibt momentan vier Endungen: mkv, mka, mk3d und mks
- „inoffiziell“ (d.h. nicht als Standard akzeptiert): audio/x-matroska, video/x-matroska und video/x-matroska-3d
- Kann als Stream via HTTP, RTP gesendet werden
- Lizenzen:
 - Matroska: GNU Lesser General Public License
 - Parser: BSD-Lizenz

Unterstützte Codecs - Video

- Raw
- MPEG 1 + 2
- MPEG 4
 - Microsoft V3
 - DivX4
 - DivX5, XviD, FFMPEG
- Real Video
- QuickTime (bzw. Videodaten daraus)
- Theora
- Apple ProRes

Unterstützte Codecs - Audio

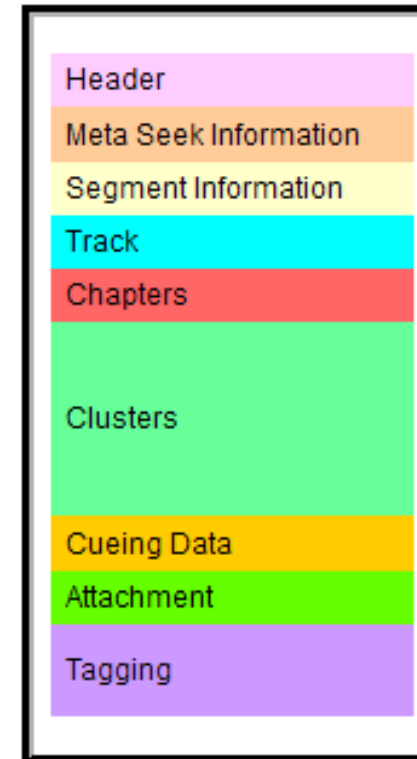
- MPEG Audio 1 - 2.5 Layer I – III (MP3)
- AC3 von Dolby
- Apple Lossless Audio Codec
- Vorbis
- FLAC
- AAC
- Realmedia Audio codecs
- Audio Codec Manager (Microsoft)
- QuickTime audio codecs (QDesign Music v1 + v2)
- WavPack lossless audio compressor
- Digital Theatre System (+Express/Lossless)
- (Musepack)

Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Aufbau Header

- Header
 - EBML Version (→ Matroska)
- Meta Seek Information
 - Index für andere Elemente
- Segment Information
 - Eindeutige ID und Namen
 - Bei Reihen vorherige und nachfolgende ID und Namen
- Track
 - Format, Auflösung, Codec(-Informationen)
- Chapters
 - Kapitel
- Clusters
 - Video- und Audioframes
- Cueing Data
 - Enthält hilfreiche Informationen zum Suchen anderer Teile
- Attachment
 - Alles mögliche
- Tagging
 - Tags ähnlich wie ID3 Tags von mp3's



<http://www.matroska.org/technical/diagram/index.html>

Aufbau Header

Elements semantic

A more detailed description of the column headers can be found in the [Specification Notes](#).

If you are interested in WebM you can have a look at this page that describes what [parts of Matroska it kept](#).

- Element Name - The full name of the described element.
- L - Level - The level within an EBML tree that the element may occur at. + is for a recursive level (can be its own child), g: global element (can be found at any level)
- EBML ID - The Element ID displayed as octets.
- Ma - Mandatory - This element is mandatory in the file.
- Mu - Multiple - The element may appear multiple times within its parent element.
- Rng - Range - Valid range of values to store in the element.
- Default - The default value of the element.
- T - Element Type - The form of data the element contains. m: Master, u: unsigned int, l: signed integer, s: string, & UTF-8 string, b: binary, f: float, d: date
- 1 - The element is contained in Matroska version 1.
- 2 - The element is contained in Matroska version 2.
- 3 - The element is contained in Matroska version 3.
- 4 - The element is contained in Matroska version 4 (v4 is still work in progress; further additions are possible).
- W - All elements available for use in WebM.
- Description - A short description of the element's purpose.

Element Name	L	EBML ID	Ma	Mu	Rng	Default	T	1	2	3	4	W	Description
EBML Header													
EBML	0	[1A][45][DF][A3]	-	-	-	-	m	-	-	-	-	-	Set the EBML characteristics of the data to follow. Each EBML document has to start with this.
EBMLVersion	1	[42][86]	-	-	-	1	u	-	-	-	-	-	The version of EBML parser used to create the file.
EBMLReadVersion	1	[42][F7]	-	-	-	1	u	-	-	-	-	-	The minimum EBML version a parser has to support to read this file.
EBMLMaxIDLength	1	[42][F2]	-	-	-	4	u	-	-	-	-	-	The maximum length of the IDs you'll find in this file (4 or less in Matroska).
EBMLMaxSizeLength	1	[42][F3]	-	-	-	8	u	-	-	-	-	-	The maximum length of the sizes you'll find in this file (8 or less in Matroska). This does not override the element size indicated at the beginning of an element. Elements that have an indicated size which is larger than what is allowed by EBMLMaxSizeLength shall be considered invalid.
DocType	1	[42][82]	-	-	-	matroska	s	-	-	-	-	-	A string that describes the type of document that follows this EBML header. 'matroska' in our case or 'webm' for webm files.
DocTypeVersion	1	[42][87]	-	-	-	1	u	-	-	-	-	-	The version of DocType Interpreter used to create the file.
DocTypeReadVersion	1	[42][85]	-	-	-	1	u	-	-	-	-	-	The minimum DocType version an Interpreter has to support to read this file.
Global elements (used everywhere in the format)													
Void	g	[EC]	-	-	-	-	b	-	-	-	-	-	Used to void damaged data, to avoid unexpected behaviors when using damaged data. The content is discarded. Also used to reserve space

http://matroska.org/technical/specs/index.html

Aufbau Header

V_REAL/????	Real Video(TM)	The stream is one of the Real Video(TM) video streams listed below. Source for the codec names are from Karl Lillevold on Doom9 . The CodecPrivate element contains a "real_video_props_t" structure in Big Endian byte order as found in librmff .	
	V_REAL/RV10	RealVideo 1.0 aka RealVideo 5	Individual slices from the Real container are combined into a single frame.
	V_REAL/RV20	RealVideo G2 and RealVideo G2+SVT	Individual slices from the Real container are combined into a single frame.
	V_REAL/RV30	RealVideo 8	Individual slices from the Real container are combined into a single frame.
	V_REAL/RV40	rv40 : RealVideo 9	Individual slices from the Real container are combined into a single frame.
V_QUICKTIME	Video taken from QuickTime(TM) files	Several codecs as stored in QuickTime, e.g. Sorenson or Cinepak. The CodecPrivate contains the complete contents of the 'stsd' atom in unmodified form which means that all elements are stored in big endian form.	
V_THEORA	Theora	<p>The private data contains the first three Theora packets in order. The lengths of the packets precedes them. The actual layout is:</p> <p>Byte 1: number of distinct packets '#p' minus one inside the CodecPrivate block. This should be '2' for current Theora headers.</p> <p>Bytes 2..n: lengths of the first '#p' packets, coded in Xiph-style lacing. The length of the last packet is the length of the CodecPrivate block minus the lengths coded in these bytes minus one.</p> <p>Bytes n+1..: The Theora identification header, followed by the comment header followed by the codec setup header. Those are described in the Theora specs.</p>	

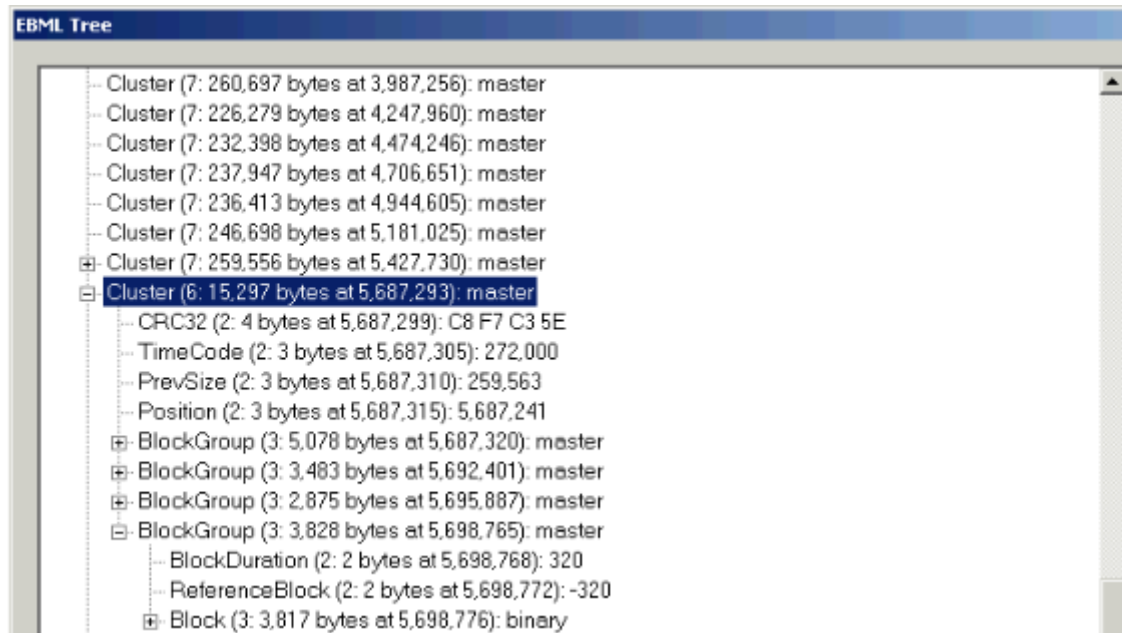
http://matroska.org/technical/specs/codecid/index.html

EBML Header

```
EBML (header: 5 bytes, data: 35 bytes, pos.: 0): master
├── EBMLVersion (header: 3 bytes, data: 1 bytes, pos.: 5): 1
├── EBMLReadVersion (header: 3 bytes, data: 1 bytes, pos.: 9): 1
├── EBMLMaxIDLength (header: 3 bytes, data: 1 bytes, pos.: 13): 4
├── EBMLMaxSizeLength (header: 3 bytes, data: 1 bytes, pos.: 17): 8
├── DocTypeVersion (header: 3 bytes, data: 1 bytes, pos.: 21): 1
├── DocTypeReadVersion (header: 3 bytes, data: 1 bytes, pos.: 25): 1
├── DocType (header: 3 bytes, data: 8 bytes, pos.: 29): matroska
└── Segment (header: 12 bytes, data: 4,696,195,032 bytes, pos.: 40): master
```

<http://matroska.org/files/matroska.pdf>

EBML Header



<http://matroska.org/files/matroska.pdf>

Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Bearbeitung...

- VirtualDubMod
- MKVToolNix
- Matroska Muxer
- AVIMux-GUI
- ...

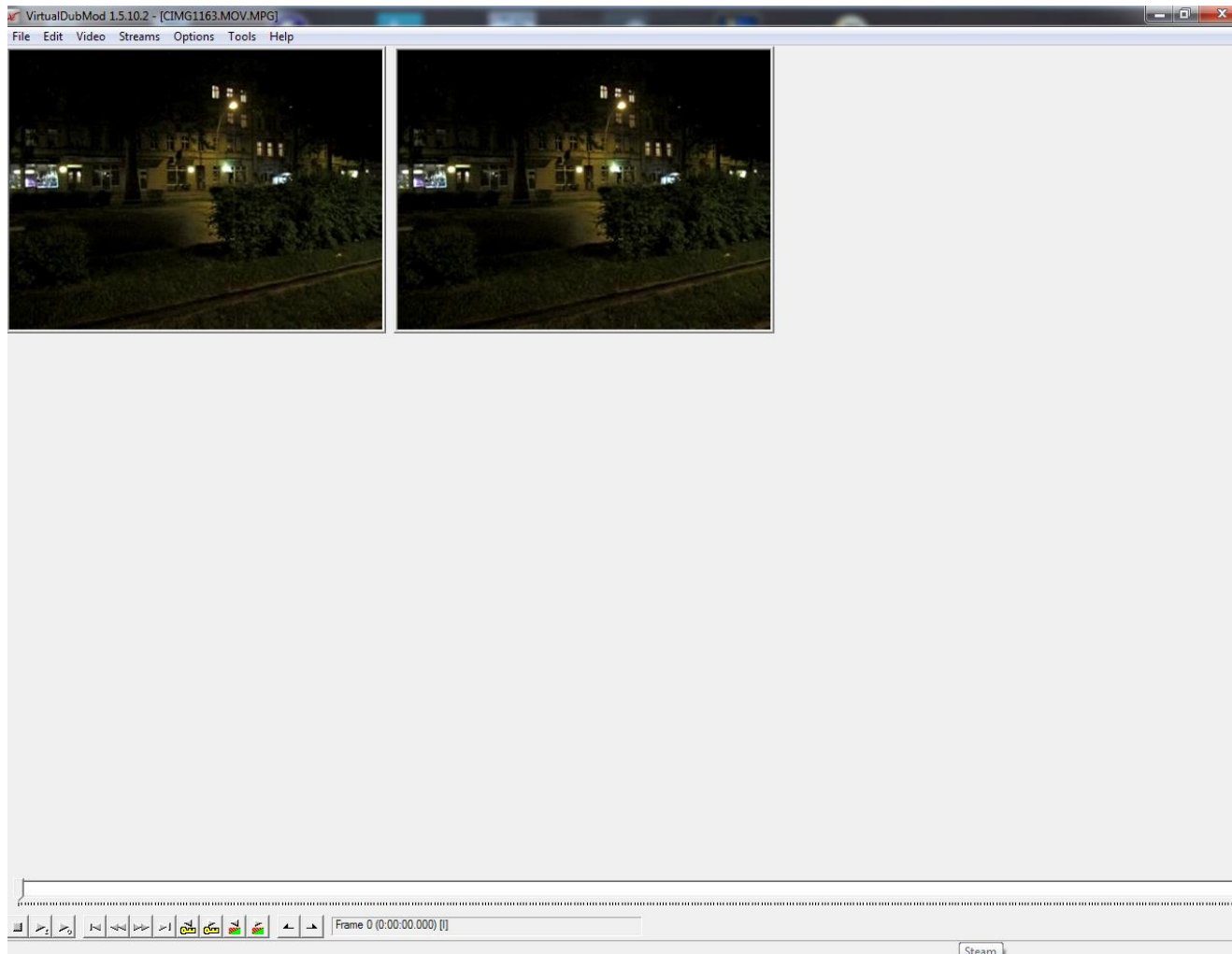
und Betrachtung

- CorePlayer (Pro and Mobile)
- VLC Mediaplayer
- Zoom Mediaplayer
- MPlayer
- foobar2000
- ...

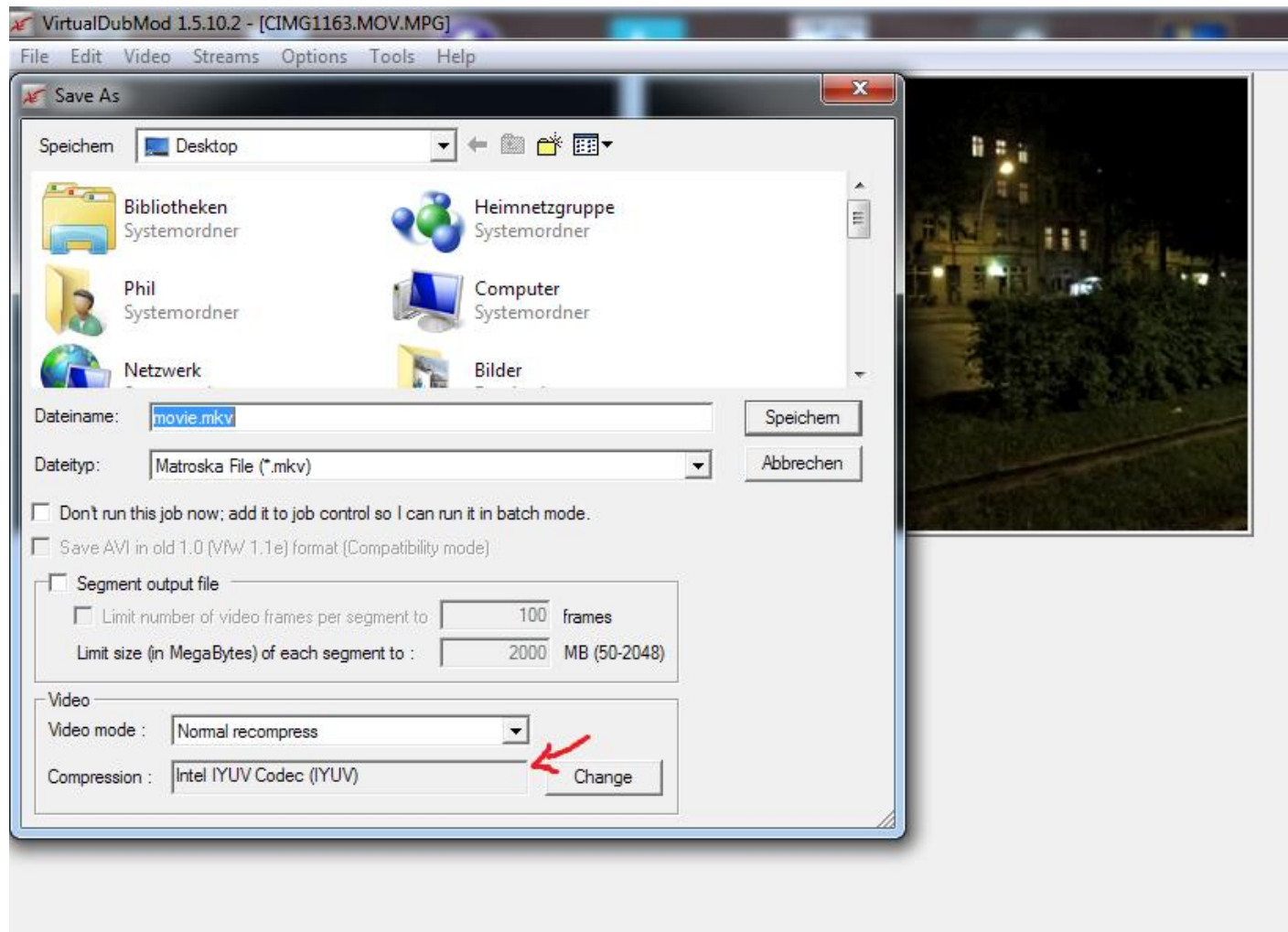
Bearbeitung

Name	Änderungsdatum	Typ	Größe
aviproxy	04.06.2012 16:36	Dateiordner	
plugins	04.06.2012 16:36	Dateiordner	
template	10.06.2013 14:03	Dateiordner	
AuxSetup (2).exe	25.08.2005 22:10	Anwendung	40 KB
auxsetup.exe	24.12.2010 12:18	Anwendung	68 KB
AviSynthLexer.lexer	14.11.2002 18:55	LEXER-Datei	56 KB
Codecs.ini	25.08.2005 23:44	Konfigurationsein...	15 KB
copying	13.09.2009 22:13	Datei	18 KB
corona.dll	10.03.2003 16:42	Anwendungserwe...	123 KB
license_corona.txt	22.05.2002 06:18	TXT-Datei	1 KB
ogg.dll	11.03.2003 22:10	Anwendungserwe...	21 KB
readme_virtualdubmod_dlls.txt	17.03.2003 20:41	TXT-Datei	2 KB
SciLexer.dll	26.04.2003 00:29	Anwendungserwe...	144 KB
vdicmdrv.dll	24.12.2010 12:18	Anwendungserwe...	68 KB
vdremote.dll	24.12.2010 12:18	Anwendungserwe...	72 KB
vdsvrlnk.dll	24.12.2010 12:17	Anwendungserwe...	64 KB
vdub.exe	24.12.2010 12:18	Anwendung	9 KB
VirtualDub.chm	24.12.2010 12:21	Kompilierte HTML...	241 KB
VirtualDub.exe	24.12.2010 12:18	Anwendung	2.608 KB
VirtualDub.vdhelp	01.12.2003 23:11	VDHELP-Datei	73 KB
VirtualDub.vdi	24.12.2010 12:18	Virtual Disk Image	216 KB
VirtualDubMod.chm	03.12.2003 20:26	Kompilierte HTML...	206 KB
VirtualDubMod.exe	20.02.2006 23:15	Anwendung	908 KB
VirtualDubMod.exe.manifest	12.09.2004 13:22	MANIFEST-Datei	1 KB
VirtualDubMod.vdi	20.02.2006 23:15	Virtual Disk Image	135 KB
vorbis.dll	11.03.2003 22:50	Anwendungserwe...	48 KB

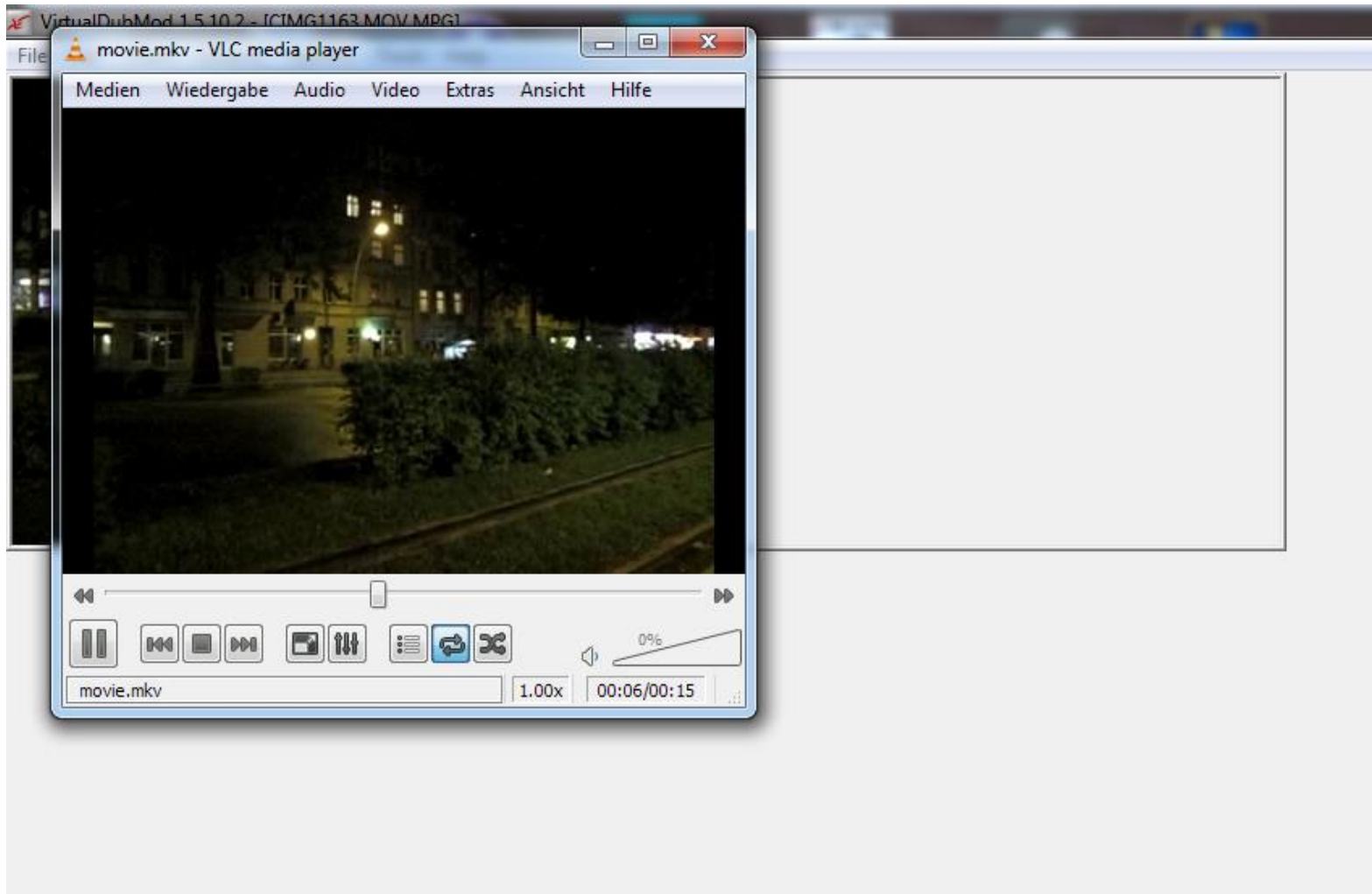
Bearbeitung



Bearbeitung



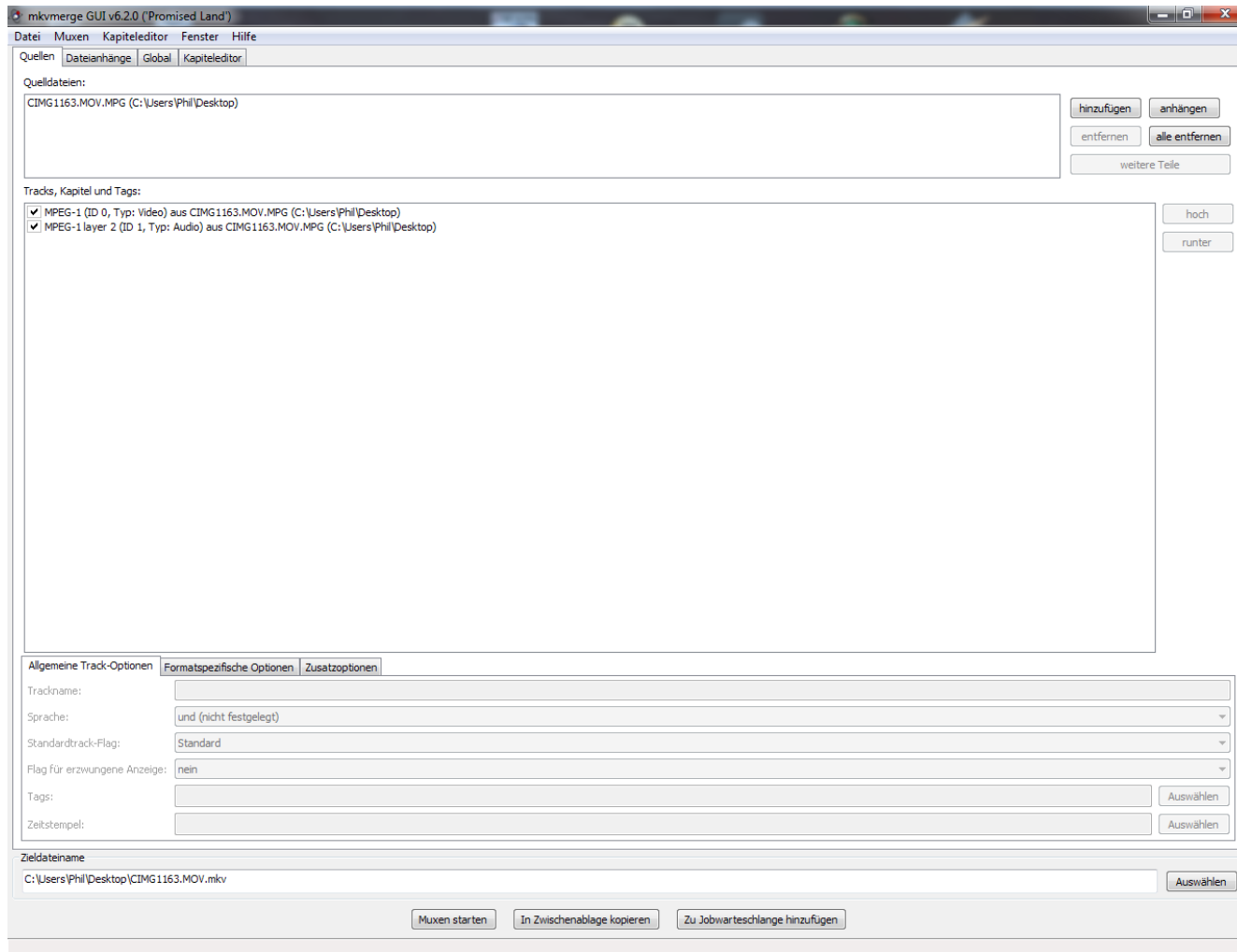
und Betrachtung



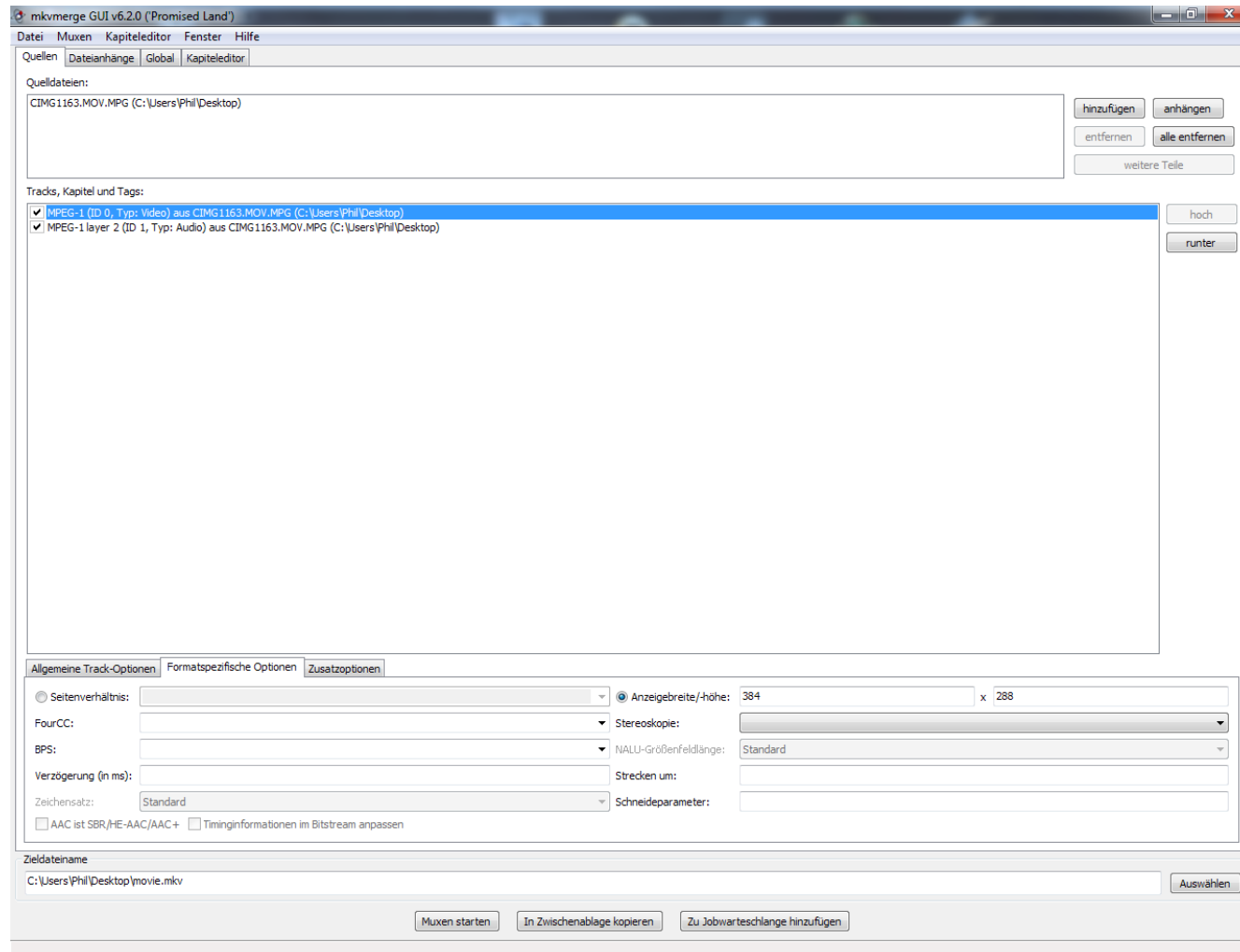
Aber dann...



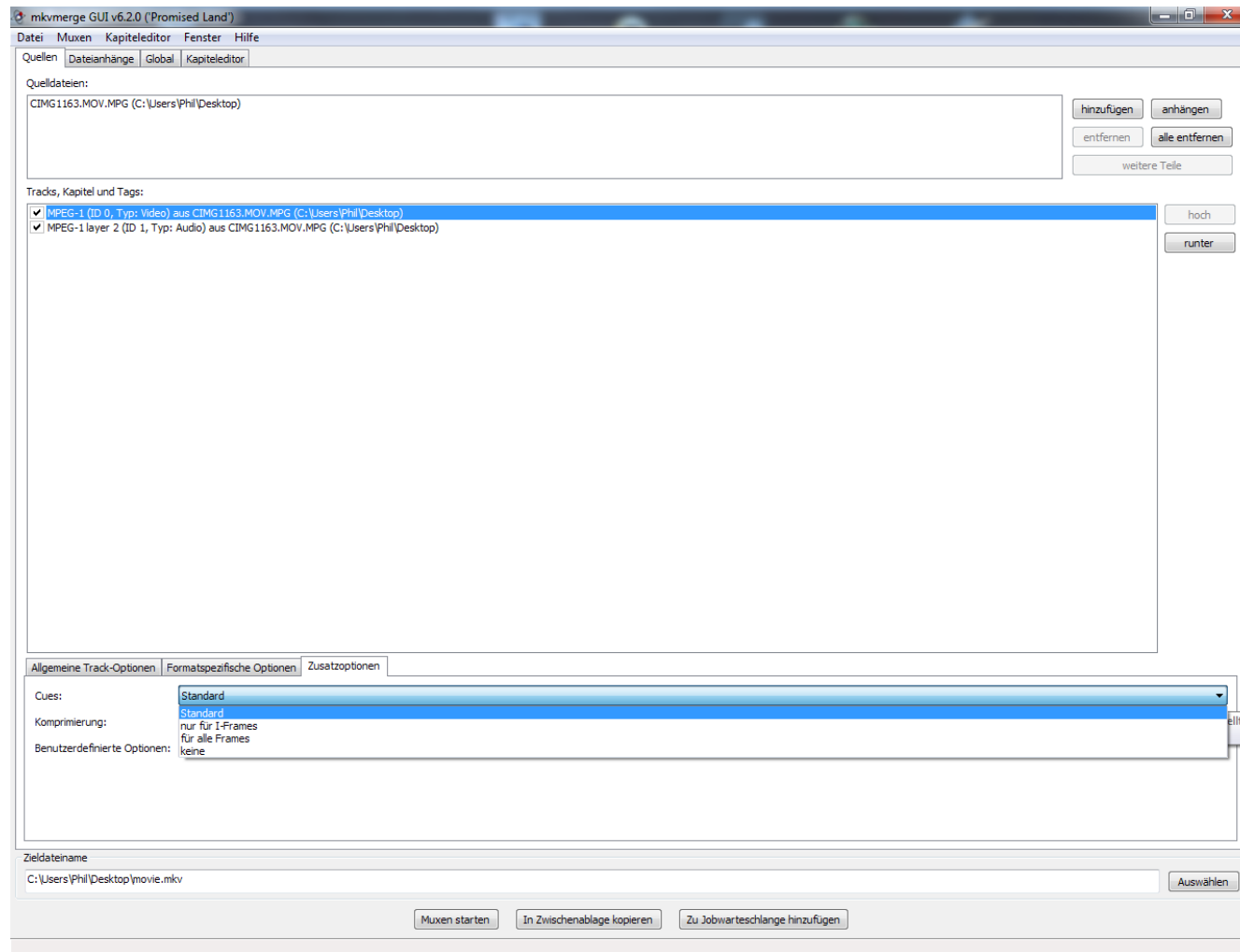
MKVToolNix



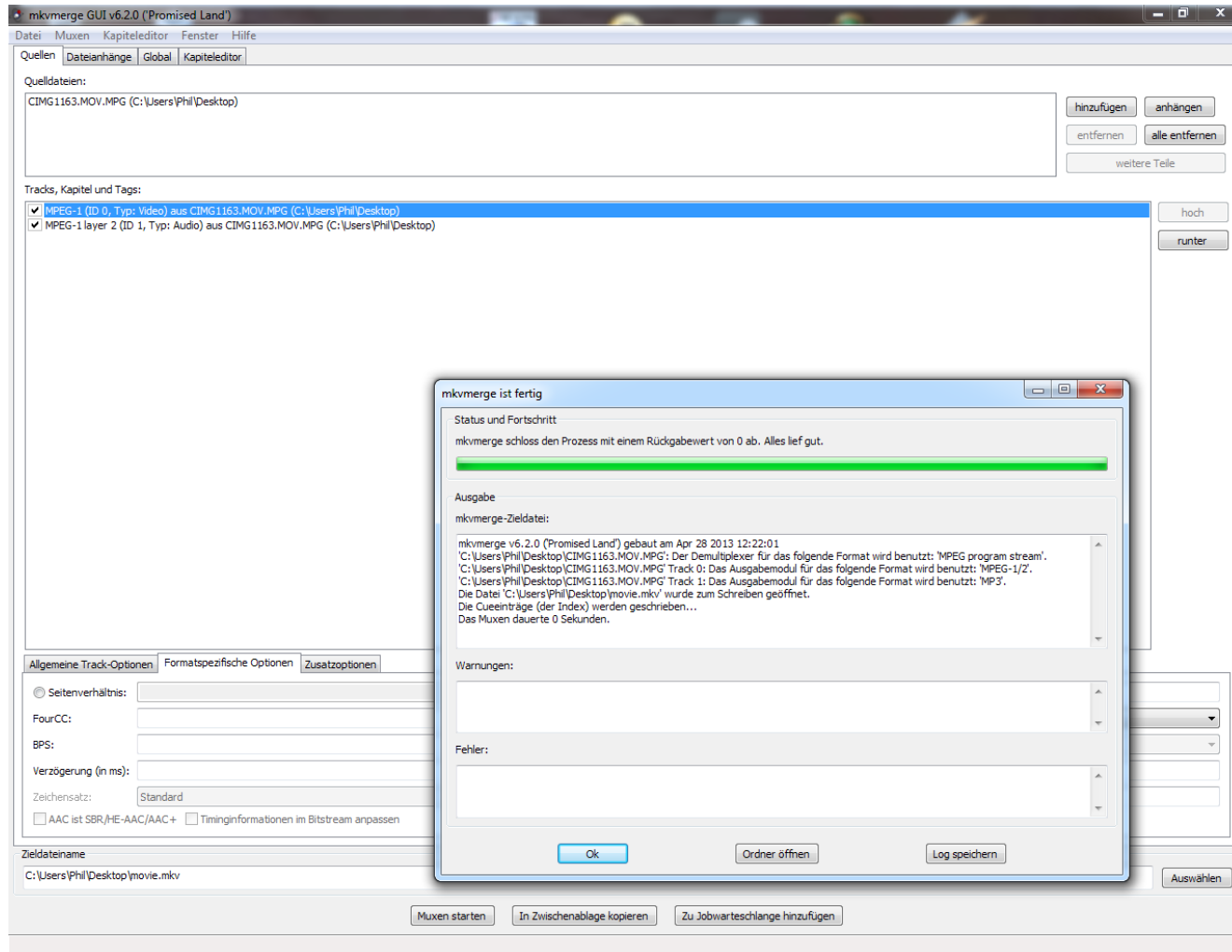
MKVToolNix



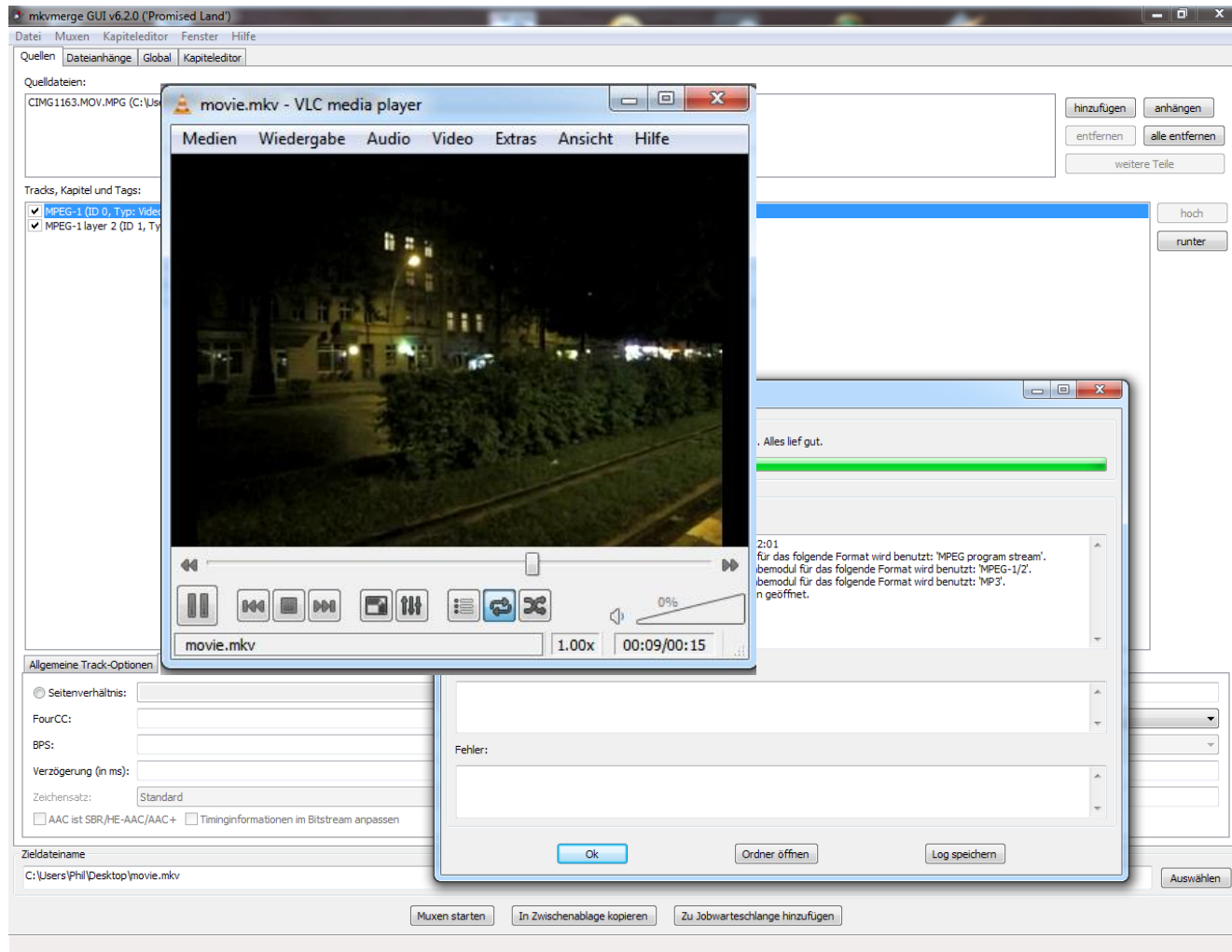
MKVToolNix



MKVToolNix



MKVToolNix



Gliederung

1. Einleitung
2. Extensible Binary Meta Language (EBML)
3. Entstehung
4. Merkmale
 - 4.1 Unterstützte Codecs
5. Aufbau
6. Bearbeitung und Betrachtung
7. Fazit

Fazit

- Durch Flexibilität für zukünftige Entwicklung offen
- Momentan ausreichend verbreitet
- Ob Ziel der Entwickler erreicht wird -> fraglich

Danke für eure Aufmerksamkeit!