



e·infra
austria

AP 5 Metadaten

Automatische Übertragung von Metadaten in Videodateien

Alexander Zartl

Universitätsbibliothek Wien

September 2019



Der Inhalt dieser Veröffentlichung steht unter einer Creative Commons
Namensnennung 4.0 Lizenz (<http://creativecommons.org/licenses/by/4.0/>)

[DOI 10.25651/1.2019.0016](https://doi.org/10.25651/1.2019.0016)

Über e-Infrastructures Austria Plus

Das Projekt „e-Infrastructures Austria Plus“ (2017-2019) ist ein vom österreichischen Bundesministerium für Bildung, Wissenschaft und Forschung (<https://bildung.bmbwf.gv.at/>) gefördertes Projekt von neun österreichischen Universitäten. Ziel des Projekts ist der koordinierte Aufbau eines österreichischen Netzwerks zur Einrichtung und Weiterentwicklung gemeinsamer e-Infrastrukturen durch Bündelung von Ressourcen und vorhandenem Wissen.

Inhalt

Einleitung	3
Vorbemerkung	3
Matroska-Dateien (MKV).....	4
2. Import und Export von MKV-Metadaten mit mkvmerge und mkvextract.....	4
2.1. Metadatenimport	4
2.2. Metadatenexport	7
3. Automatisierung durch Stapelverarbeitung	7
4. Automatische Metadatenextraktion beim Upload in ein Repository	7
MP4-Dateien.....	8
5. Überblick	8
6. Import und Export von MP4-Metadaten mit FFmpeg.....	8
6.1. Metadatenimport	9
6.2. Metadatenexport	10
7. Automatisierung durch Stapelverarbeitung	11
8. Automatische Metadatenextraktion beim Upload in ein Repository	11
Fazit	12
Anhang	13
9. Die Datei <i>Metadaten_fuer_MKV_Import.xml</i>	13
10. Die Datei <i>Metadaten_fuer_MP4_Import.txt</i>	16

Einleitung

Ziel des Arbeitspakets „*Metadaten*“ ist die Erarbeitung eines Metadatenframeworks für Repositorien in Österreich und von Guidelines, die mit FAIR-Richtlinien (FAIR Guiding Principles for scientific data management and stewardship) arbeiten. Die Suche von Forschungsdaten und anschließende Projektkooperationen können repositorienübergreifend nur funktionieren, wenn ein gemeinsamer Satz von Metadaten beim Einbringen der Daten erstellt wird und wenn die eingebrachten Daten mit Kontrolliertem Vokabular beschrieben werden. Das Projekt will diese beiden Voraussetzungen schaffen.

Vorbemerkung

Unter dem Begriff „automatische Übernahme von Metadaten“ soll in diesem Dokument das programmgesteuerte Auslesen und Speichern von Tags aus Videodateien verstanden werden. Untersucht werden dabei die beiden wichtigsten Containerformate Matroska (MKV) und MP4.

Tags sind Metadaten zur Beschreibung von Dateien, die an einer – je nach Format unterschiedlichen – genau spezifizierten Stelle direkt in diese eingebettet sind. Dies bedeutet, dass die Metadaten zu einem Objekt nicht gesondert verwaltet werden müssen und die Verbindung zwischen Objekt und Objektbeschreibung nicht verloren gehen kann. Voraussetzung zu einer sinnvollen Nutzung der vorgestellten Methoden ist, dass zu den Videos Metadaten in Form einer externen (maschinenlesbaren) Beschreibung vorliegen (XML-Dateien für Matroska, speziell strukturierte UTF8-Textdateien für MP4).

Dieses Dokument beabsichtigt nicht eine erschöpfende Abhandlung des Themas, sondern soll als erster Einstieg und Orientierungshilfe dienen. Bei den verwendeten Programmen wurde darauf geachtet, leicht zu beschaffende Freeware- und Open Source-Produkte zu verwenden. Das zugrundeliegende Betriebssystem ist Windows 7 bzw. 10, doch sollte es ohne großen Aufwand möglich sein, die Abläufe auch auf Linux-Geräten nachzustellen. Als Begleitmaterial werden zum Download zwei Datenpakete angeboten, die die verwendeten Beispielvideos und Softwarepakete (in den jeweiligen Windows-Versionen) enthalten.

Matroska-Dateien (MKV)

1. Überblick

Das Containerformat *Matroska* (übliche Dateierdung: *.mkv) hat in den letzten Jahren weite Verbreitung gefunden und wird von einer umfangreichen Community unterstützt. Das wichtigste Werkzeug zur Verarbeitung von MKV-Metadaten ist das Programmpaket *MKVToolNix*, das – mit Stand von Juli 2019 in der Version 35 – für Windows, macOS und zahlreiche Linux-Distributionen verfügbar ist.¹ Es enthält Programme für Metadatenimport (*mkvmerge*), Metadatenexport (*mkvextract*) u.a. und kann wahlweise über die Kommandozeile oder eine gemeinsame grafische Benutzeroberfläche bedient werden.

Grundsätzlich sei angemerkt, dass der Funktionsumfang von *MKVToolNix* weit über die Verarbeitung von Metadaten hinausgeht. Das Programmpaket dient ganz allgemein der Zusammenführung und Extraktion von Datenströmen als Bestandteilen von MKV-Dateien – die Metadatenintegration ist lediglich ein Spezialfall, bei dem ein Metadatenstrom in Form einer XML-Datei in den Container eingefügt wird. Für weitere Informationen vgl. die einschlägige Programmdokumentation.

2. Import und Export von MKV-Metadaten mit *mkvmerge* und *mkvextract*

Dieser Abschnitt zeigt anhand eines einfachen Beispiels, wie ein MKV-Video ohne Metadaten mit solchen angereichert und diese wieder ausgelesen werden können. Die Metadaten müssen dazu als XML-Datei mit UTF8-Codierung vorliegen. Als Begleitmaterial gibt es ein von PHAIDRA² herunterladbares Datenpaket mit den Beispieldateien und dem aktuellen *MKVToolNix*-Programmpaket.³

2.1. Metadatenimport

Die Aufgabe, die wir uns hier stellen, ist die Anreicherung einer MKV-Datei mit deskriptiven, auf den Videoinhalt bezogenen Metadaten. Wir benutzen dazu eine Teilmenge der von Matroska offiziell unterstützten Tags⁴: TITLE (Filmtitel), DIRECTOR (Regisseur), DESCRIPTION (Inhaltsangabe) und einige andere.

Besonders hingewiesen sei auf die Möglichkeit der Wiederholung bestimmter Felder, wovon wir bei den Tags COMPOSER und ACTOR / CHARACTER Gebrauch machen. Das Feld ACTOR / CHARACTER ist überdies von der verschachtelten XML-Struktur her so angelegt, dass es eine eindeutige Zuordnung zwischen Darsteller und dargestellter Rolle ermöglicht. Die vollständige XML-Datei findet sich im Anhang.

Wenn Sie *MKVToolNix* installiert haben, rufen Sie das GUI auf. Das Objekt, das hier bearbeitet werden soll, ist die Datei *Katzen_MKV_ohne_Metadaten.mkv*, die Sie im heruntergeladenen Datenpaket im Ordner Daten finden. Durch einfaches Ziehen und Ablegen wird sie der Liste der Quelldateien hinzugefügt (Abb. 1).

¹ Siehe <https://mkvtoolnix.download>

² Permanent Hosting, Archiving and Indexing of Digital Resources and Assets, das Digital Asset Management System der Universität Wien zur dauerhaften Sicherung von digitalen Beständen.

³ Siehe <https://phaidra.univie.ac.at/o:996648>

⁴ Siehe <https://matroska.org/technical/specs/tagging/index.html> sowie die Beispiele dazu.

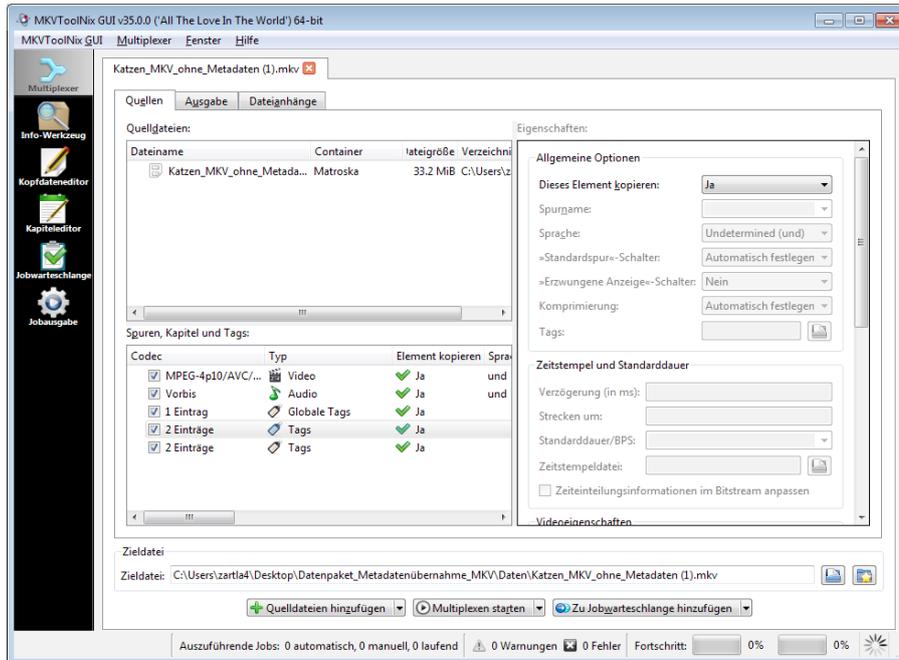


Abb. 1: Quelleneditor im MKVToolNix-GUI

Es fällt auf, dass die Quelldatei außer Video- und Audio-Datenstrom bereits drei Datenströme mit Tags enthält, von denen einer als „Globale Tags“ bezeichnet wird. Es handelt sich hier um technische Metadaten für Bild- und Tonspur (mit je zwei Einträgen: Spieldauer und verwendeter Encoder) und den gemeinsamen Datenstrom für jene Tags, die sich auf den Container als Ganzes beziehen (ein Eintrag: der verwendete Encoder). Diese globalen Tags sollen nun noch um weitere Angaben ergänzt werden.

Dazu gehen wir zum Reiter *Ausgabe* und ziehen die Datei *Metadaten_fuer_MKV_Import.xml* in das Feld *Globale Tags* (Abb. 2). Ein Klick auf die Schaltfläche *Multiplexen starten* bewirkt die Ausgabe der neuen Datei *Katzen_MKV_ohne_Metadaten (1).mkv*, in der nun alle gewünschten Informationen gespeichert sind.

Zum Vergleich enthält das begleitende Datenpaket auch die Datei *Katzen_MKV_mit_Metadaten.mkv*, für die der oben beschriebene Vorgang bereits durchgeführt wurde und die daher identisch mit *Katzen_MKV_ohne_Metadaten (1).mkv* sein sollte.

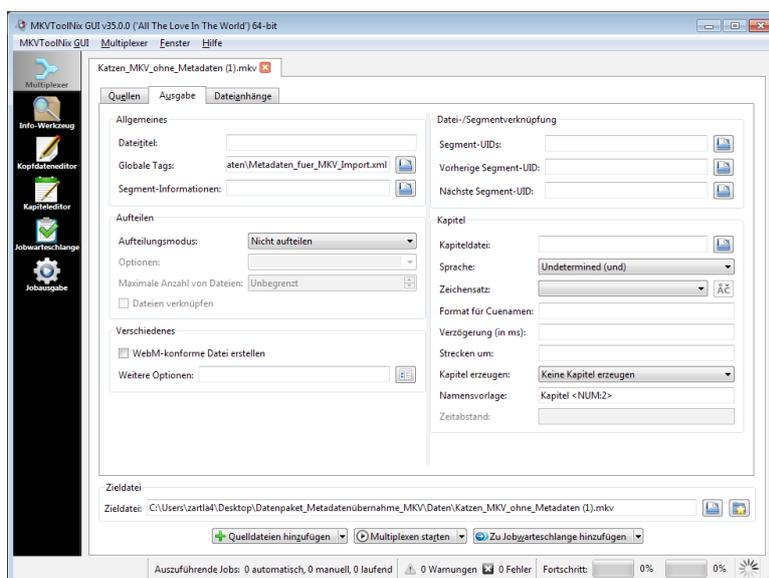


Abb. 2: Ausgabeditor im MKVToolNix-GUI

Zur Überprüfung sehen wir uns das erzeugte Video mit dem beliebten *VLC Media Player* an (Menüpunkt *Werkzeuge* → *Medieninformation*; Abb. 3). Leider ist der Support der MKV-Tags durch den VLC nicht perfekt, weshalb nicht alle Angaben richtig dargestellt werden, insbesondere dann, wenn es sich um mehrfach vorkommende Felder handelt. Aber es ist durchaus zu erkennen, dass die Abspielsoftware nun auf deskriptive Metadaten zugreifen und diese auswerten kann.

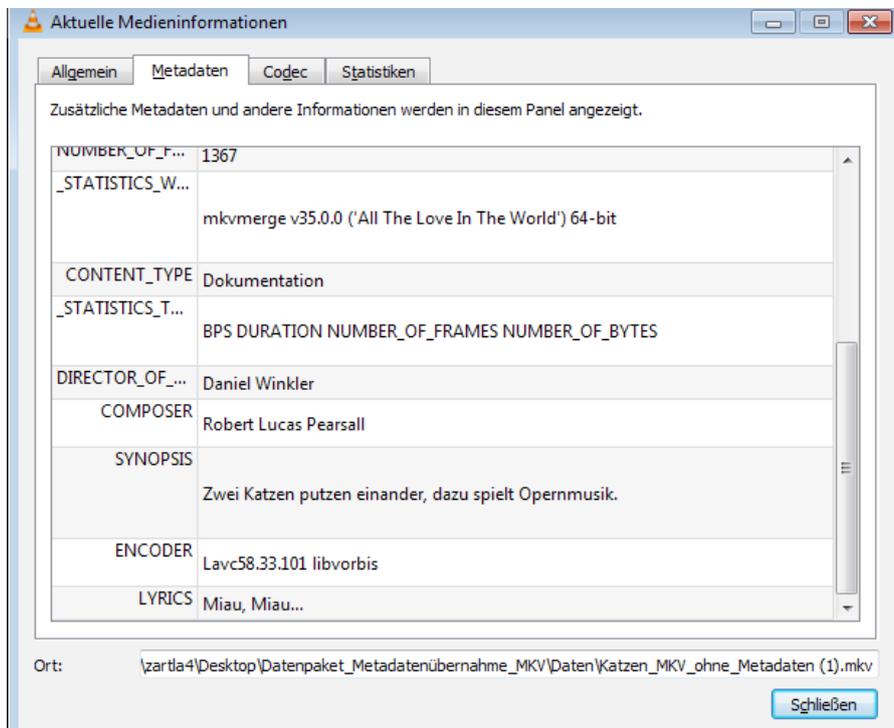


Abb. 3: Anzeige deskriptiver Metadaten im VLC Media Player

Wenn man die neue Datei im GUI als Quelle betrachtet, stellt man fest, dass die Anzahl der Einträge in den globalen Tags auf 18 angewachsen ist (Abb. 4).

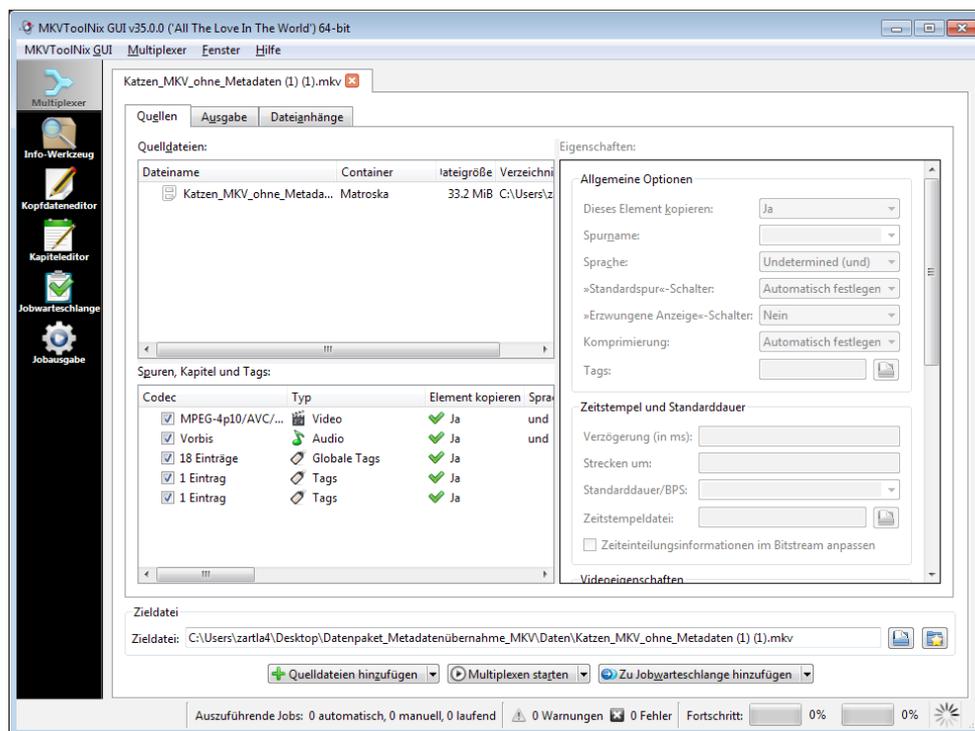


Abb. 4: Anzeige der globalen Tags im MKVToolNix-GUI

2.2. Metadatenexport

Das Auslesen der Tags erfolgt mittels des Programms `mkvextract`, das wir der methodischen Vielfalt halber über die Kommandozeile ansteuern. Die Syntax entspricht folgendem Schema:

```
mkvextract Quelldateiname.mkv tags [Optionen] Zieldateiname.xml
```

Unter der Voraussetzung, dass sich MKVToolNix im üblichen Programmordner befindet und wir das begleitende Datenpaket im Ordner `c:\video` entpackt haben, heißt das in unserem Fall:

```
„C:\Program Files\MKVToolNix\mkvextract.exe“ tags „C:\video\Datenpaket_
Metadatenübernahme_MKV\Daten\Katzen_MKV_ohne_Metadaten (1).mkv“ -r „C:\
video\Datenpaket_Metadatenübernahme_MKV\Daten\Metadaten.xml“
```

Die resultierende Datei *Metadaten.xml* enthält nicht nur die zuvor eingefügten globalen Tags, sondern auch die Tags für den Video- und den Audio-Datenstrom. Zum Vergleich wurde dem begleitenden Datenpaket die Datei *Katzen_MKV_mit_Metadaten.xml* beigefügt, die mit unserem Ergebnis identisch sein sollte.

3. Automatisierung durch Stapelverarbeitung

Ein automatisierter Arbeitsablauf zum Taggen einer größeren Anzahl von Videos lässt sich mit dem Zusatzprogramm *MKVCleaver* realisieren.⁵ Selbstverständlich ist es auch mit Windows-Hausmitteln (einfache Batch-Datei, PowerShell) ohne großen Aufwand möglich, etwas Vergleichbares herzustellen – zum Beispiel so:

```
cls
@echo off
echo *****
echo.
echo XML-Dateien mit den Tags aus den Matroska-Dateien extrahieren
echo.
for %%i in (*.mkv) do (
    echo %%i
    „C:\Program Files\MKVToolNix\mkvextract.exe“ „%%i“ tags „%~ni.xml“
)
echo.
echo *****
echo.
```

4. Automatische Metadatenextraktion beim Upload in ein Repository

Um die eingebetteten Metadaten eines Videos beim Upload in ein Repository direkt nutzen zu können, ist es erforderlich, die Dateien durch in die Repositoryssoftware integrierte Programmroutinen zu analysieren und die Tags zu extrahieren. Für auf PHP basierende Systeme gibt es dafür die Programmbibliothek *getID3()*⁶, die Metadaten aus einer Vielzahl von Multimediaformaten – darunter Matroska und MP4 – auslesen kann.

⁵ Siehe <https://www.videohelp.com/software/MKVCleaver>

⁶ Siehe <http://getid3.sourceforge.net/> bzw. die Demo-Seite <https://www.getid3.org/demo/>

Damit sollte es im Prinzip möglich sein, die Datenfelder des Repositoriums automatisch zu befüllen und den Aufwand für die manuelle Metadatenerfassung beim Upload zu verringern oder im günstigsten Fall sogar zu eliminieren.

Auf die Frage, ob und wie dies in der Praxis geschehen kann, kann hier nicht näher eingegangen werden, da die Möglichkeiten zur Umsetzung vom jeweiligen Repositorium abhängen. In jedem Fall muss `getID3()` an die vorhandene Softwareumgebung angepasst werden, wozu aller Voraussicht nach professionelle Kenntnisse der PHP-Programmierung nötig sind.

MP4-Dateien

5. Überblick

Das Containerformat MP4 (übliche Dateiendung: *.mp4) ist vom Verbreitungsgrad her gesehen wesentlich bedeutender als MKV und kann zum gegenwärtigen Zeitpunkt als gängigstes Standardformat für Videos angesehen werden. Wie sich im Folgenden jedoch zeigen wird, gibt es bezüglich der Unterstützung von Metadaten Nachteile und MP4 ist daher nur zweite Wahl, wenn dieser Aspekt im Vordergrund stehen soll.

Für die Verarbeitung und das Taggen von MP4-Dateien gibt es eine Vielzahl von Programmen, vom simplen Einsteigertool bis zum komplexen Allroundwerkzeug. Nachdem der Zeitrahmen für die Fertigstellung der vorliegenden Arbeit umfangreiche Softwaretests nicht zuließ, fiel die Wahl auf das bekannte Videobearbeitungsprogramm *FFmpeg*⁷, da hier bereits vorhandene Vorkenntnisse genutzt werden konnten und es viele einschlägige Foren und Blogs gibt, die Antworten zu einem breiten Spektrum von Fragen bieten.

Auch zu diesem Abschnitt gibt es ein begleitendes Datenpaket, das aus PHAIDRA heruntergeladen werden kann⁸ und neben den verwendeten Beispieldateien den aktuellen Build von *FFmpeg* sowie den plattformübergreifenden Texteditor *MadEdit*⁹ enthält (mehr dazu weiter unten).

6. Import und Export von MP4-Metadaten mit FFmpeg

Ein einfaches Beispiel soll nun zeigen, wie ein MP4-Video ohne Metadaten mit solchen angereichert und diese wieder ausgelesen werden können. Die Metadaten müssen dazu als Textdatei mit UTF8-Codierung vorliegen.

Die zuvor zu beantwortende Frage, welche Metadaten in MP4-Dateien überhaupt eingebettet werden können, ist nicht ganz leicht zu beantworten. MP4 ist ein ISO-standardisiertes Format (offiziell: *MPEG-4 Part 12* und *14*), die zugehörigen Dokumente¹⁰ müssen käuflich erworben werden. Die Dokumentation ist überdies enorm umfangreich und ihre Lektüre setzt (hier leider nicht vorhandenes) Expertenwissen bzw. eine längere Einarbeitungszeit voraus¹¹. Für die vorliegende Arbeit wurde daher nach Beispielanwendungen gesucht, die konkrete Listen von erlaubten MP4-Tags bieten; gefunden wurden solche schließlich in der Dokumentation zu der

⁷ Siehe <https://ffmpeg.org/>. Der Funktionsumfang von *FFmpeg* geht weit über das Taggen hinaus; wenn man sich aber – wie in unserem Fall – darauf beschränkt, sind weitere Kenntnisse des Programms nicht unbedingt erforderlich.

⁸ Siehe <https://phaidra.univie.ac.at/o:999180>

⁹ Siehe <https://sourceforge.net/projects/madedit/>

¹⁰ ISO/IEC 14496-12 und -14

¹¹ Laut Wikipedia ist auch die Verwendung von Metadaten nach dem Standard Extensible Metadata Platform (XMP) möglich; vgl. https://en.wikipedia.org/wiki/MPEG-4_Part_14#Metadata

Mediaplayer-Software *Kodi*¹², der Dokumentation zur Linux-Videoschnittsoftware *Kdenlive*¹³ sowie im *MultimediaWiki*¹⁴. Es sei darauf hingewiesen, dass die Informationen in diesen drei Quellen voneinander abweichen und in einem Punkt sogar widersprüchlich sind (was hier noch eine Rolle spielen wird).

ACHTUNG: Die Erstellung einer Metadaten-Datei für FFmpeg funktioniert NICHT mit dem gewöhnlichen Texteditor in Windows, obwohl dieser UTF8-Unterstützung bietet. Verantwortlich dafür dürfte das Byte Order Mark (BOM) sein, das der Windows-Editor nicht unterdrücken kann und das das korrekte Einlesen der Metadaten-Datei durch FFmpeg offenbar verhindert. In MadEdit lässt sich das BOM unter dem Menüpunkt *Tools* entfernen.

6.1. Metadatenimport

Analog zu Abschnitt 2.1. soll eine MP4-Datei mit deskriptiven Metadaten angereichert werden. Folgende Felder sind in der im begleitenden Datenpaket enthaltenen Datei *Metadaten_fuer_MP4_Import.txt* enthalten: *title; artist; author; album_artist; composer; album; date; network; comment; genre; copyright; lyrics; synopsis* (vgl. Abschnitt 10. im Anhang – Änderungen für Testzwecke nur mit MadEdit!).

Der Aufruf von FFmpeg erfolgt wieder über die Kommandozeile¹⁵. Die Syntax lautet hier:

```
ffmpeg -i Quelldateiname.mp4 -i Tagdateiname.txt -map_metadata 1 -c copy
Zieldateiname.mp4
```

Unter der Voraussetzung, dass sich FFmpeg im üblichen Programmordner befindet und wir das begleitende Datenpaket im Ordner *c:\video* entpackt haben, heißt das:

```
„C:\Program Files\ffmpeg\bin\ffmpeg.exe“ -i
„C:\video\Datenpaket_Metadatenübernahme_MP4\Daten\Katzen_MP4_ohne_Metadaten.
mp4“ -i
„C:\video\Datenpaket_Metadatenübernahme_MP4\Daten\Metadaten_fuer_MP4_Import.
txt“ -map_metadata 1 -c copy
„C:\video\Datenpaket_Metadatenübernahme_MP4\Daten\Katzen_MP4_ohne_Metadaten
(1) .mp4“
```

¹² Siehe https://kodi.wiki/view/Video_file_tagging#MP4_tag_options

¹³ Siehe <https://kdenlive.org/en/project/adding-meta-data-to-mp4-video/>

¹⁴ Siehe https://wiki.multimedia.cx/index.php/FFmpeg_Metadata

¹⁵ Auf Wunsch sind mehrere Programmpakete mit grafischen Benutzeroberflächen, die auf FFmpeg zurückgreifen, verfügbar – siehe z.B. [https://github.com/amiaopensource/ffmpeg-amia-wiki/wiki/3\)-Graphical-User-Interface-Applications-using-FFmpeg](https://github.com/amiaopensource/ffmpeg-amia-wiki/wiki/3)-Graphical-User-Interface-Applications-using-FFmpeg)

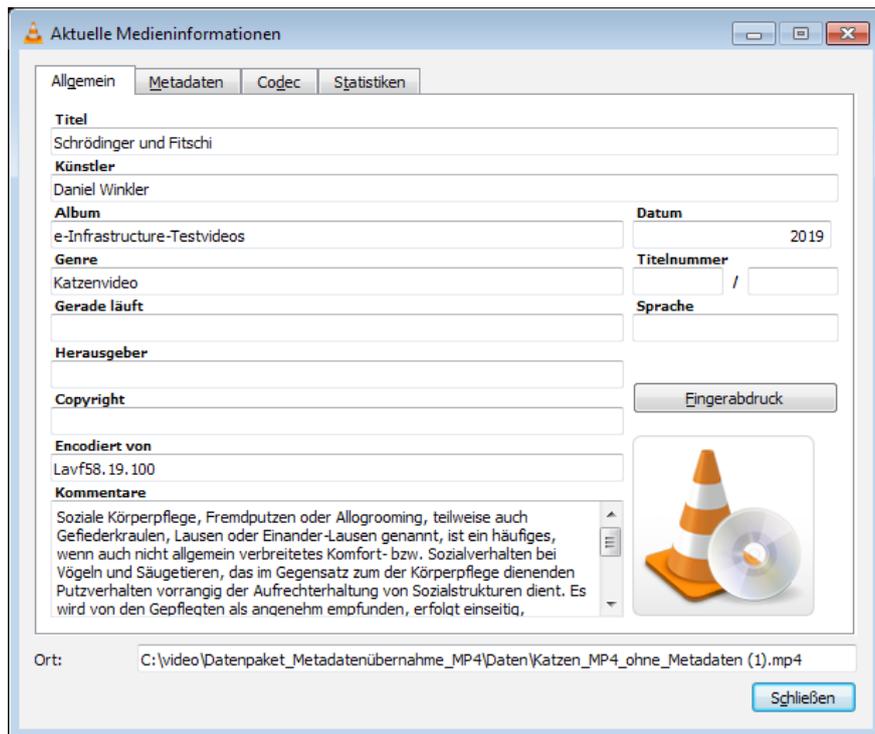


Abb. 5: Anzeige deskriptiver Metadaten im VLC Media Player

Abb. 5 zeigt, dass das Einfügen zumindest teilweise erfolgreich war – inwieweit, wird sich im folgenden Abschnitt zeigen.

6.2. Metadatenexport

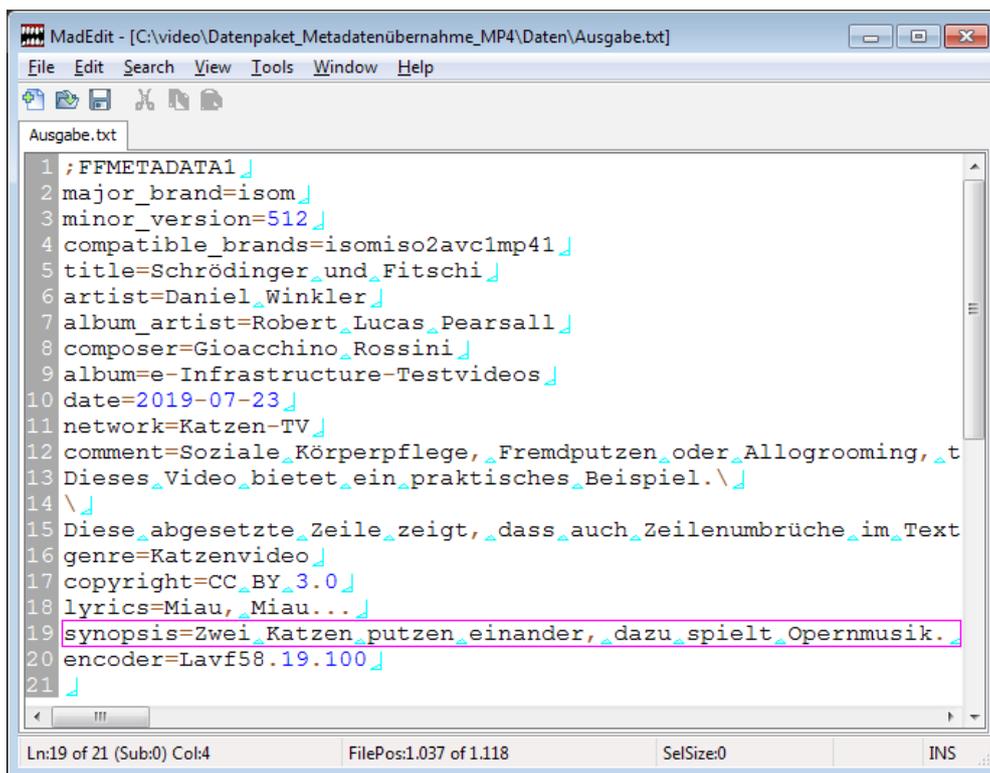
FFmpeg kann die Tags nicht nur einlesen, sondern auch auslesen und in einer Textdatei speichern. Die Syntax dazu lautet:

```
ffmpeg -i Quelldateiname.mp4 -f ffmetadata Tagdateiname.txt
```

Wir wenden dies auf die in 6.1. erzeugte Datei an:

```
„C:\Program Files\ffmpeg\bin\ffmpeg.exe“ -i
„C:\video\Datenpaket_Metadatenübernahme_MP4\Daten\Katzen_MP4_ohne_Metadaten
(1).mp4“ -f ffmetadata
„C:\video\Datenpaket_Metadatenübernahme_MP4\Daten\Ausgabe.txt“
```

Mit MadEdit geöffnet sieht das Ergebnis wie folgt aus:



```
1 ;FFMETADATA1
2 major_brand=isom
3 minor_version=512
4 compatible_brands=isomiso2avc1mp41
5 title=Schrödinger und Fitschi
6 artist=Daniel Winkler
7 album_artist=Robert Lucas Pearsall
8 composer=Gioacchino Rossini
9 album=e-Infrastructure-Testvideos
10 date=2019-07-23
11 network=Katzen-TV
12 comment=Soziale Körperpflege, Fremdputzen oder Allogrooming, t
13 Dieses Video bietet ein praktisches Beispiel.\
14 \
15 Diese abgesetzte Zeile zeigt, dass auch Zeilenumbrüche im Text
16 genre=Katzenvideo
17 copyright=CC BY 3.0
18 lyrics=Miau, Miau...
19 synopsis=Zwei Katzen putzen einander, dazu spielt Opernmusik.
20 encoder=Lavf58.19.100
21
```

Abb. 6: Die extrahierte Metadaten-Datei in MadEdit

Vier neue Felder sind hinzugekommen (*major_brand*; *minor_version*; *compatible_brands*; *encoder*) – es handelt sich um technische Angaben, wobei auffällt, dass Angaben zu Spielzeit, Auflösung und andere wichtige Werte nicht erfasst wurden. Außerdem ist das Feld *author* verlorengegangen, was darauf zurückzuführen ist, dass MultimediaWiki einen Fehler enthält: *author* wird von MP4 nicht unterstützt, wie die Dokumentation zu Kdenlive richtig anmerkt.

Betrachtet man *Katzen_MP4_ohne_Metadaten (1).mp4* mit dem Metadaten-Betrachter MediaInfo¹⁶, so stellt sich heraus, dass die fehlenden technischen Angaben sehr wohl darin enthalten sind, aber von FFmpeg nicht ausgelesen werden.

7. Automatisierung durch Stapelverarbeitung

Die Automatisierung der Aufgaben in den Abschnitten 6.1. und 6.2. lässt sich analog zu Abschnitt 3 auf einfache Weise durch Windows-Batchdateien oder PowerShell realisieren.

8. Automatische Metadatenextraktion beim Upload in ein Repository

Siehe dazu Abschnitt 4.

¹⁶ Siehe <https://mediaarea.net/de/MediaInfo>

Fazit

Für die automatisierte Metadatenübernahme aus Videodateien erweist sich das Matroska-Format (MKV) als bessere Lösung im Vergleich zu MP4. Die Vorteile sind im Einzelnen:

- MKV ist ein vollständig freies und offenes Format, das nicht nur die üblichen proprietären Codecs unterstützt, sondern auch freie wie Theora oder Dirac (Video) und Vorbis (Audio).
- Für die Bearbeitung von MKV-Dateien gibt es das Programmpaket MKVToolNix, das alle Funktionen – auch das Einfügen und Extrahieren von Metadaten – in einer Hand vereinigt.
- Die Dokumentation zu MKV und MKVToolNix ist übersichtlich und (soweit es im Rahmen der vorliegenden Arbeit überprüft werden konnte) vollständig, speziell auch was die Metadaten betrifft.

MP4 ist zwar deutlich weiter verbreitet, hat aber einige wesentliche Nachteile, die sich vor allem auch im Hinblick auf die Integration von Metadaten bemerkbar machen:

- MP4 ist zumindest teilweise patengeschützt und bezüglich der verwendbaren Codecs weniger flexibel als MKV.
- Zur Verarbeitung von MP4-Dateien gibt es eine Vielzahl von Programmpaketen, doch ohne weitere Untersuchungen ist im Vorhinein nicht klar, welches für welchen speziellen Zweck besonders geeignet ist.
- MP4 ist ein ISO-Standard, was bedeutet, dass die offizielle Dokumentation (die ISO-Norm) im Regelfall nicht frei verfügbar ist. Einschlägige Informationen sind zwar im WWW zu finden, aber nur verstreut, bruchstückhaft und in Einzelfällen auch fehlerhaft.

Für beide Formate besteht im Prinzip die Möglichkeit, die in den Tags enthaltenen Metadaten mit der Programmbibliothek `getID3()` direkt beim Upload in ein Repositorium auszulesen und weiter zu verarbeiten. Insgesamt ergibt sich aber auf Grund der oben aufgelisteten Vorteile die Empfehlung, in digitalen Repositorien das Matroska-Format für Videos bevorzugt zu unterstützen.

Anhang

9. Die Datei *Metadaten_fuer_MKV_Import.xml*

Dies ist der Inhalt der Datei *Metadaten_fuer_MKV_Import.xml*, die in Abschnitt 1.2.1. als Datenstrom in den MKV-Container eingefügt wird. Sie enthält sechzehn Datenfelder und dazu zwei Unterfelder.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Tags>
  <Tag>
    <Simple>
      <Name>TITLE</Name>
      <String>Schrödinger und Fitschi</String>
    <Simple/>
  </Simple>
  <Simple>
    <Name>DIRECTOR</Name>
    <String>Alexander Zartl</String>
  <Simple/>
</Simple>
  <Simple>
    <Name>DATE_RELEASED</Name>
    <String>2018-04-16</String>
  <Simple/>
</Simple>
  <Simple>
    <Name>DESCRIPTION</Name>
    <String>Dieses Video zeigt zwei Katzen bei der sozialen Fellpflege.</String>
  <Simple/>
</Simple>
  <Simple>
    <Name>SYNOPSIS</Name>
    <String>Zwei Katzen putzen einander, dazu spielt Opernmusik.</String>
  <Simple/>

```

</Simple>

<Simple>

<Name>DIRECTOR_OF_PHOTOGRAPHY</Name>

<String>Daniel Winkler</String>

<Simple/>

</Simple>

<Simple>

<Name>LYRICS</Name>

<String>Miau, Miau...</String>

<Simple/>

</Simple>

<Simple>

<Name>COMMENT</Name>

<String>Soziale Körperpflege, Fremdputzen oder Allogrooming, teilweise auch Gefiederkraulen, Lausen oder Einander-Lausen genannt, ist ein häufiges, wenn auch nicht allgemein verbreitetes Komfort- bzw. Sozialverhalten bei Vögeln und Säugetieren, das im Gegensatz zum der Körperpflege dienenden Putzverhalten vorrangig der Aufrechterhaltung von Sozialstrukturen dient. Es wird von den Gepflegten als angenehm empfunden, erfolgt einseitig, wechselseitig oder gleichzeitig und ist auf die für sie selbst schwer zugänglichen Körperstellen von Artgenossen gerichtet.

Dieses Video bietet ein praktisches Beispiel.

Diese abgesetzte Zeile zeigt, dass auch Zeilenumbrüche im Text kein Problem darstellen.</String>

<Simple/>

</Simple>

<Simple>

<Name>GENRE</Name>

<String>Katzenvideo</String>

<Simple/>

</Simple>

<Simple>

```
<Name>COPYRIGHT</Name>

<String>CC BY 3.0</String>

<Simple/>

</Simple>

<Simple>

  <Name>CONTENT_TYPE</Name>

  <String>Dokumentation</String>

  <Simple/>

</Simple>

<Simple>

  <Name>PRODUCER</Name>

  <String>e-Infrastructure-Testvideogemeinschaft</String>

  <Simple/>

</Simple>

<Simple>

  <Name>COMPOSER</Name>

  <String>Gioacchino Rossini</String>

  <Simple/>

</Simple>

<Simple>

  <Name>COMPOSER</Name>

  <String>Robert Lucas Pearsall</String>

  <Simple/>

</Simple>

<Simple>

  <Name>PRODUCTION_STUDIO</Name>

  <String>Österreichische Zentralbibliothek für Physik</String>

  <Simple/>

</Simple>

<Simple>

  <Name>ACTOR</Name>
```

```

    <String>Schrödinger</String>

    <Simple>
        <Name>CHARACTER</Name>
        <String>Linke Katze</String>
    </Simple>
</Simple>
<Simple>
    <Name>ACTOR</Name>
    <String>Fitschi</String>
    <Simple>
        <Name>CHARACTER</Name>
        <String>Rechte Katze</String>
    </Simple>
</Simple>
</Tag>
</Tags>

```

10. Die Datei *Metadaten_fuer_MP4_Import.txt*

Dies ist der Inhalt der Datei *Metadaten_fuer_MP4_Import.txt*, die in Abschnitt 2.2.1. in den MP4-Container eingefügt wird. Sie enthält dreizehn Datenfelder.

```

;FFMETADATA1
title=Schrödinger und Fitschi
artist=Daniel Winkler
author=Alexander Zartl
album_artist=Robert Lucas Pearsall
composer=Gioacchino Rossini
album=e-Infrastructure-Testvideos
date=2019-07-23
network=Katzen-TV
comment=Soziale Körperpflege, Fremdputzen oder Allogrooming, teilweise auch Gefiederkraulen, Lausen oder Einander-Lausen genannt, ist ein häufiges, wenn auch nicht allgemein verbreitetes Komfort- bzw. Sozialverhalten bei Vögeln und Säugetieren, das im Gegensatz zum der Körperpflege dienenden Putzverhalten vorrangig der Aufrechterhaltung von Sozialstrukturen dient. Es wird von den

```

Gepflegten als angenehm empfunden, erfolgt einseitig, wechselseitig oder gleichzeitig und ist auf die für sie selbst schwer zugänglichen Körperstellen von Artgenossen gerichtet.\

Dieses Video bietet ein praktisches Beispiel.\

\

Diese abgesetzte Zeile zeigt, dass auch Zeilenumbrüche im Text kein Problem darstellen.

genre=Katzenvideo

copyright=CC BY 3.0

lyrics=Miau, Miau...

synopsis=Zwei Katzen putzen einander, dazu spielt Opernmusik.