

Datenaustausch zwischen Mync und DaVinci Resolve

mit

Mync_nach_DVR.lua

Inhaltsverzeichnis

Systemvoraussetzung

Was kann das Script

Installation

Script benutzen

Tipps

Systemvoraussetzung

Windows 10,
DaVinci Resolve 16 Studio,
Mync v 1.6.0xxxx Standard
Sqlite3.exe (Command-Line Shell)

Das Script ist in Lua geschrieben und wird direkt in DaVinci Resolve ausgeführt

Ich habe es auch mit DaVinci Resolve 15 Studio zum laufen bekommen, aber da ist der Vorgang etwas umständlicher, da ein Script über das Menü nur im Fusion Tab abgearbeitet wird. Man kann aber über die Console das ganze Script rein kopieren.

Leider werden aber im DVR 15 Studio die SmartBins nicht automatisch mit den Keywords verbunden. Muss dann manuell gemacht werden.

Da die meisten sehr wahrscheinlich doch in absehbarer Zeit auf DVR 16 updaten werden, mache ich nur bei einer größeren Nachfrage eine Beschreibung für das Handling mit DRV 15 Studio.

Ob auch Mync Basic unterstützt wird, kann ich nicht sagen, da ich nicht weiß, ob da die selbe Datenbank zugrunde liegt. Ich vermute aber mal ja, bin mir aber nicht sicher. Einfach ausprobieren, passieren kann nichts, da nur lesend auf die Datenbank zugegriffen wird.

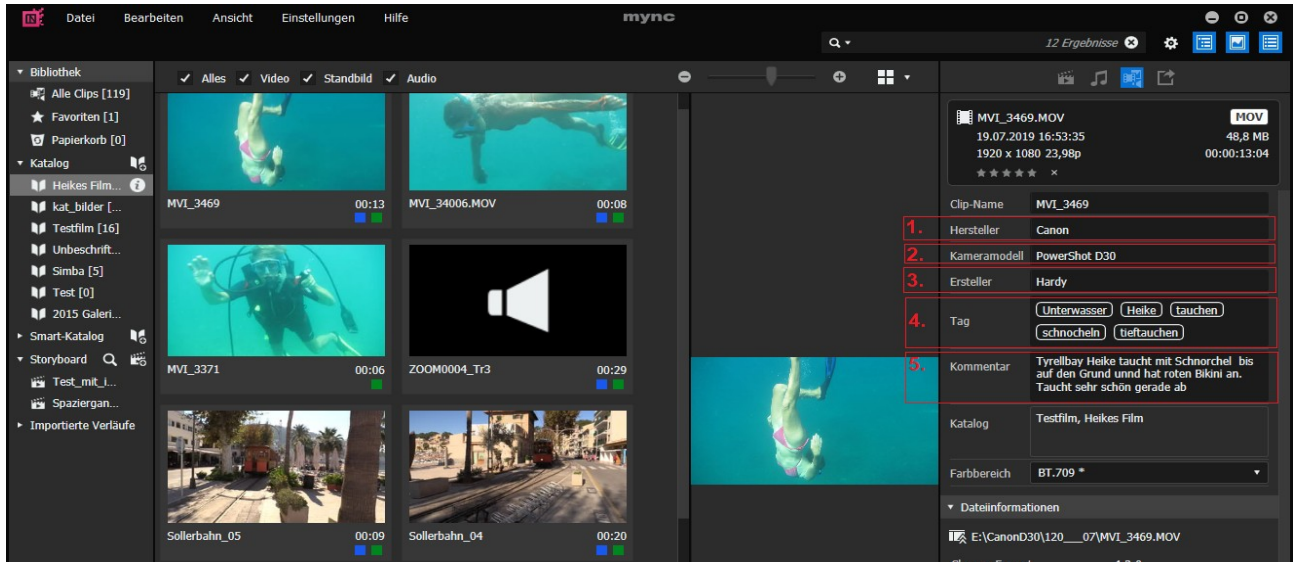
Bei langsameren Rechnern flackert beim Script abarbeiten der Hintergrund, das ist normal, weil beim Datenbank abfragen ein DOS Fenster kurz geöffnet und gleich wieder geschlossen wird.

Was kann das Script

Das Script „Mync_nachDVR.lua“ liest von Mync die Schlagworte „Hersteller“, „Kameramodell“, „Ersteller“, „Tag“ sowie den „Kommentar“ aus und speichert diese in DaVinci Resolve (unten im Text als DVR abgekürzt). Voraussetzung ist, dass die Video und Audioclips in DVR schon geladen sind.

Was ich nicht implementiert habe (da für mich unwichtig) sind Bilder, die werden nicht berücksichtigt, da DVR diese vom Namen her immer etwas umändert um Bildfrequenzen darstellen zu können.

Ich zeige zuerst auf, was es kann und weiter unten werde ich dann auf die Installation und Handhabung eingehen, wobei das wirklich einfach ist.



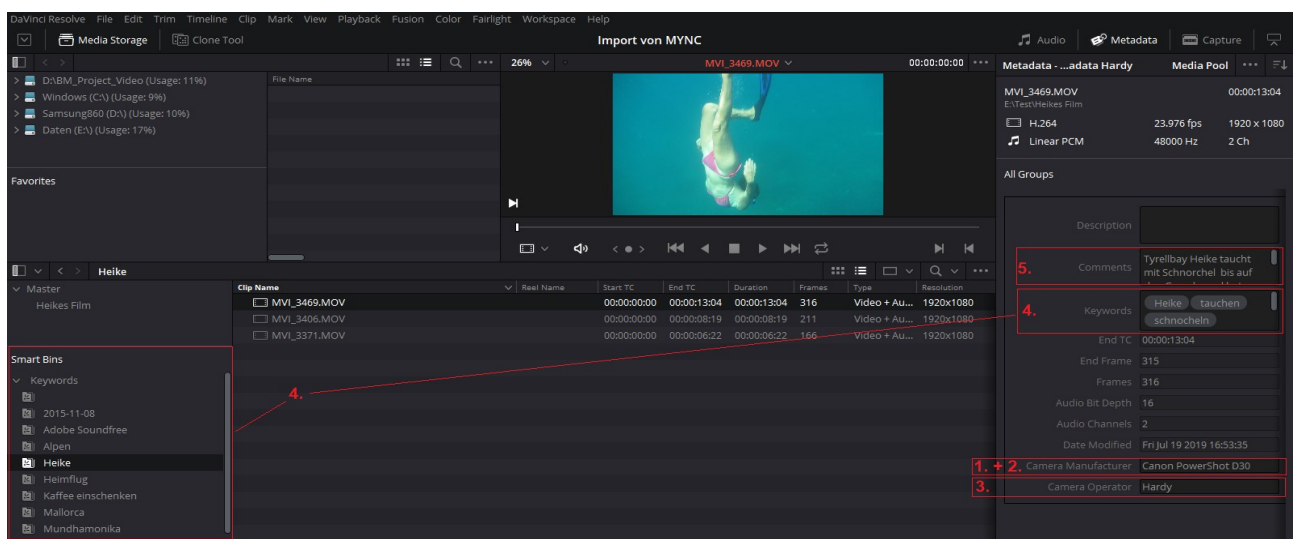
Wir sehen oben Mync und unten DVR.

Die Felder 1. Hersteller und 2. Kameramodell in Mync werden in DVR zusammengefasst und unter „Camera Manufacturer“ gespeichert. Reihenfolge ist zuerst Hersteller und dann Kameramodell.

Das Feld 3. Ersteller in Mync wird in DVR unter Camera Operator gespeichert

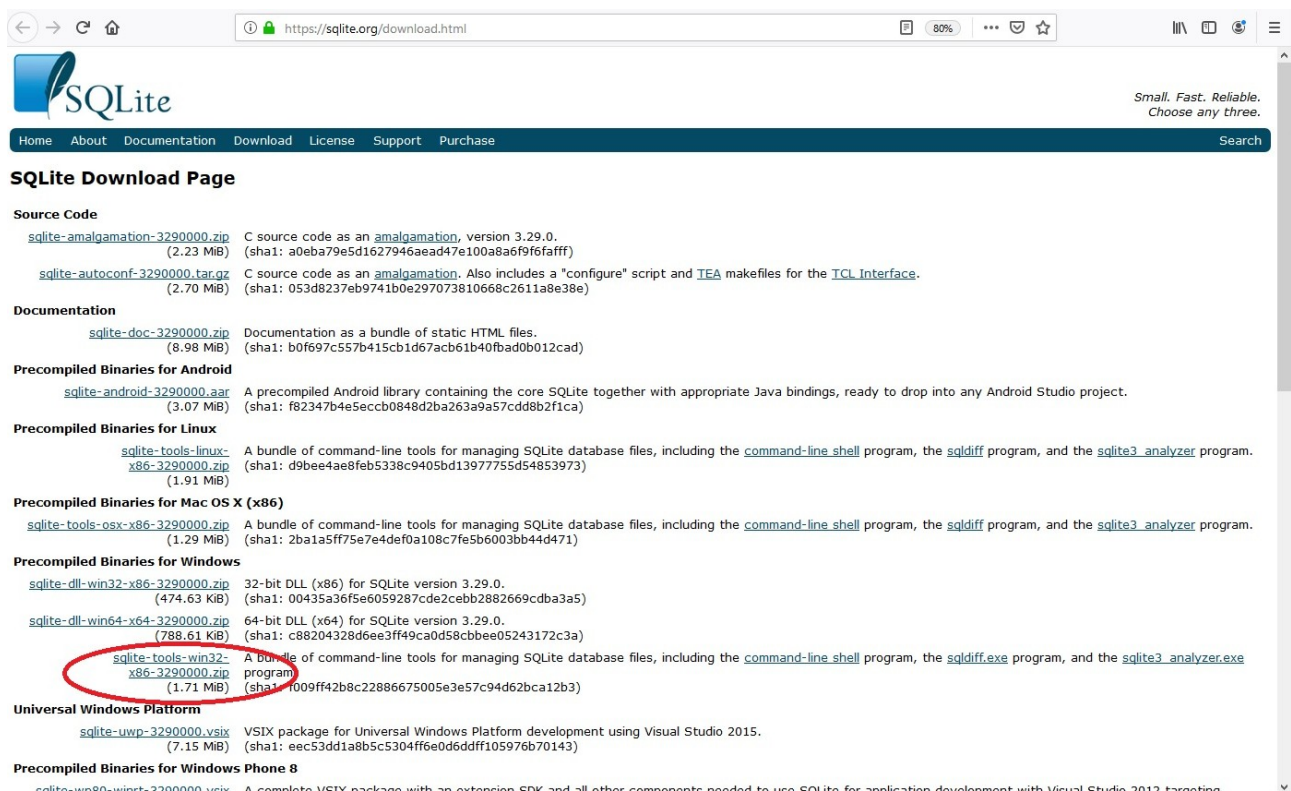
Das Feld 4. Tag in Mync wird in DVR unter Keywords gespeichert. Dabei wird jeder einzelne Tag unter einem separaten Keyword gespeichert. Ferner wird in DVR auch unter Smart Bins jedes Keyword aufgeführt. Zusätzlich wird ein leeres Keyword eingefügt, damit man sehr einfach auch alle Medien anzeigen kann. (siehe oberstes Keyword unten links.)

Das Feld 5. Kommentar in Mync wird in DVR unter Comments gespeichert



Installation

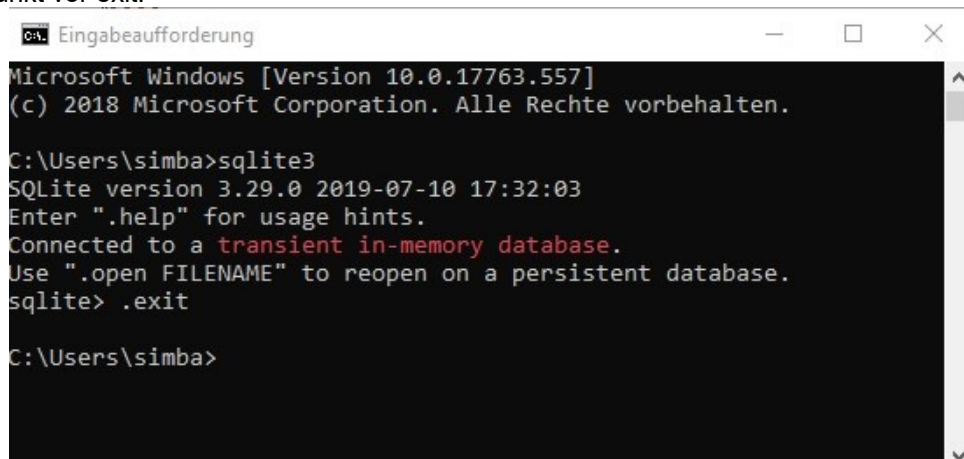
1. Nachdem das Script „Mync_nachDVR.lua“ heruntergeladen ist, wird es in das Verzeichnis (ich gehe hier von einer Standard-Installation von DVR aus) „C:\ProgramData\Blackmagic Design\DaVinci Resolve\Fusion\Scripts\Comp“ eingefügt. Sollte das Verzeichnis „C:\ProgramData“ nicht sichtbar sein, dann im Windows Explorer unter der Kategorie Ansicht ein Häkchen bei „Ausgeblendete Elemente“ setzen
2. Um zu überprüfen ob das richtige Verzeichnis ausgewählt wurde kurz in DVR im Menü „Workspace -> Scripts“ nachschauen. Da muss es nach dem kopieren zu sehen sein
3. „sqlite3-tools-win32-x86-3290000.zip“ herunterladen (siehe unten) und entpacken. Die Datei sqlite.exe in ein Verzeichnis kopieren und die Systemvariable „Path“ so setzen, damit Sqlite3 gefunden wird.



The screenshot shows the SQLite Download Page with the following content:

- Source Code**
 - [sqlite-amalgamation-3290000.zip](#) (2.23 MiB): C source code as an amalgamation, version 3.29.0. (sha1: a0eba79e5d1627946ae47e100a8a6f9f6afff)
 - [sqlite-autoconf-3290000.tar.gz](#) (2.70 MiB): C source code as an amalgamation. Also includes a "configure" script and TEA makefiles for the TCL Interface. (sha1: 053d8237eb9741b0e297073810668c2611a8e38e)
- Documentation**
 - [sqlite-doc-3290000.zip](#) (8.98 MiB): Documentation as a bundle of static HTML files. (sha1: b0f697c557b415cb1d67acb61b40fbad0b012cad)
- Precompiled Binaries for Android**
 - [sqlite-android-3290000.aar](#) (3.07 MiB): A precompiled Android library containing the core SQLite together with appropriate Java bindings, ready to drop into any Android Studio project. (sha1: f82347b4e5eccb0848d2ba263a9a57cdd8b2f1ca)
- Precompiled Binaries for Linux**
 - [sqlite-tools-linux-x86-3290000.zip](#) (1.91 MiB): A bundle of command-line tools for managing SQLite database files, including the `command-line_shell` program, the `sqldiff` program, and the `sqlite3_analyzer` program. (sha1: d9bee4ae8feb5338c9405bd13977755d54853973)
- Precompiled Binaries for Mac OS X (x86)**
 - [sqlite-tools-osx-x86-3290000.zip](#) (1.29 MiB): A bundle of command-line tools for managing SQLite database files, including the `command-line_shell` program, the `sqldiff` program, and the `sqlite3_analyzer` program. (sha1: 2ba1a5ff75e7e4def0a108c7fe5b6003bb44d471)
- Precompiled Binaries for Windows**
 - [sqlite-dll-win32-x86-3290000.zip](#) (474.63 KiB): 32-bit DLL (x86) for SQLite version 3.29.0. (sha1: 00435a36f5e6059287cde2cebb2882669cdba3a5)
 - [sqlite-dll-win64-x64-3290000.zip](#) (788.61 KiB): 64-bit DLL (x64) for SQLite version 3.29.0. (sha1: c88204328d6ee3ff49ca0d58cbee05243172c3a)
 - [sqlite-tools-win32-x86-3290000.zip](#) (1.71 MiB): A bundle of command-line tools for managing SQLite database files, including the `command-line_shell` program, the `sqldiff.exe` program, and the `sqlite3_analyzer.exe` program. (sha1: f009ff42b8c22886675005e3e57c94d62bca12b3)
- Universal Windows Platform**
 - [sqlite-uwp-3290000.vsix](#) (7.15 MiB): VSIX package for Universal Windows Platform development using Visual Studio 2015. (sha1: eec53dd1a8b5c5304ffe0d6dfff105976b70143)
- Precompiled Binaries for Windows Phone 8**
 - [sqlite-wp8-winnt-3290000.vsix](#): A complete VSIX package with an extension SDK and all other components needed to use SQLite for application development with Visual Studio 2012 targeting

4. Überprüfen ob die sqlite3 aus der „Eingabeaufforderung“ in einem beliebigen Verzeichnis gestartet werden kann. Mit „exit“ schließt man das Programm wieder und kommt auf die DOS – Ebene zurück. Man achte auf den Punkt vor exit.



```
Microsoft Windows [Version 10.0.17763.557]
(c) 2018 Microsoft Corporation. Alle Rechte vorbehalten.

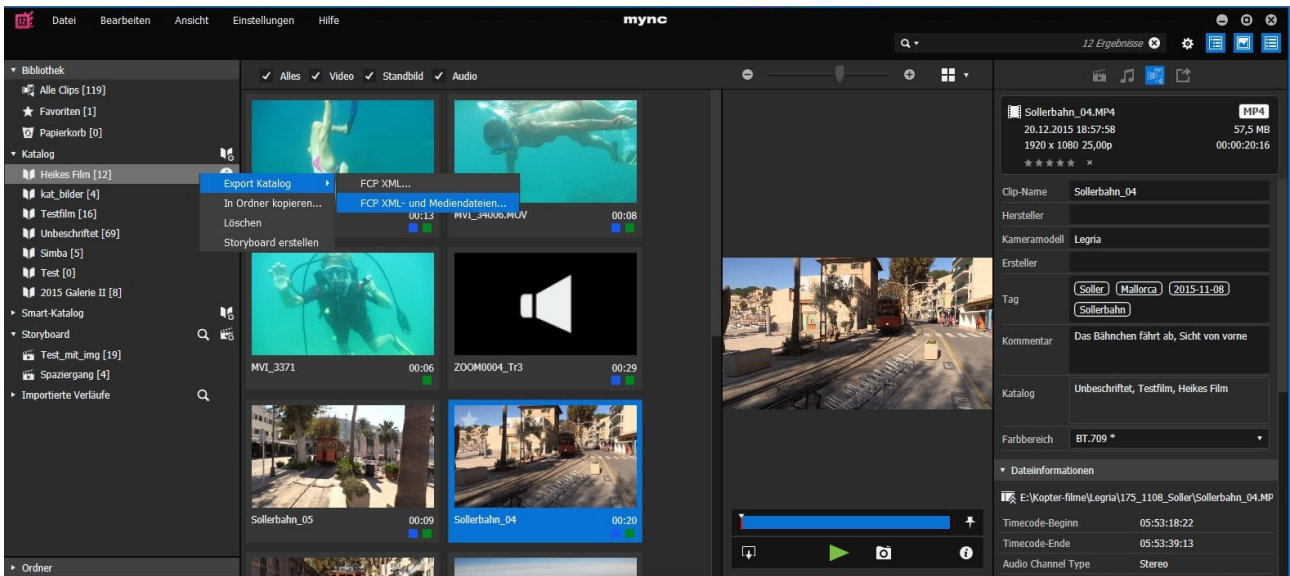
C:\Users\simba>sqlite3
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .exit

C:\Users\simba>
```

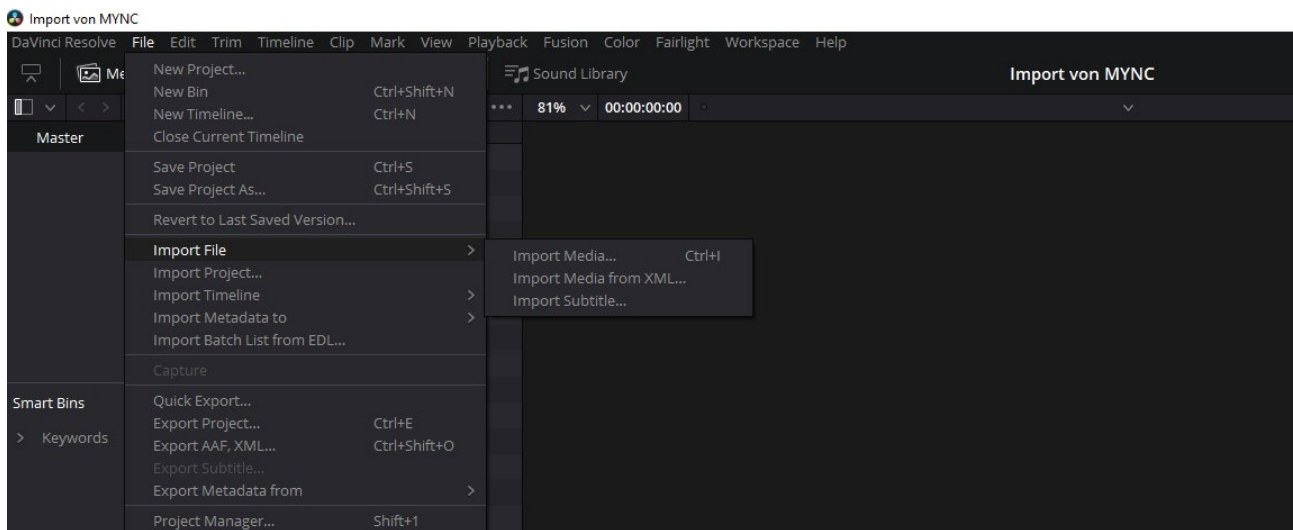
Ist das erledigt und funktioniert, dann ist die Installation abgeschlossen.

Script benutzen

1. Der einfachste Weg ist ein Katalog oder ein Storyboard als „FCP XML- und Mediendateien...“ aus Mync zu exportieren.



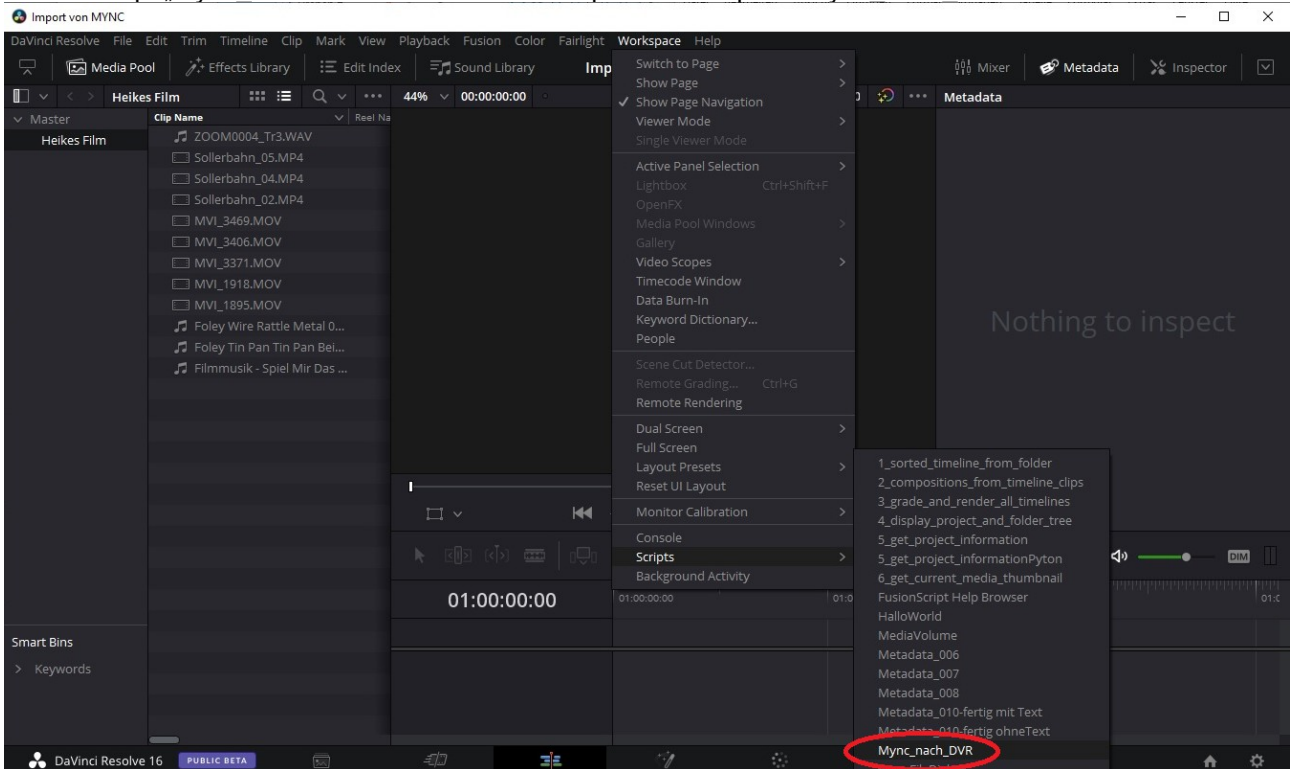
2. In DVR den Mediapool oder das Edifenster öffnen und im Menü „File→ ImportFile →Import Media from XML“ den Katalog oder das Storyboard importieren.



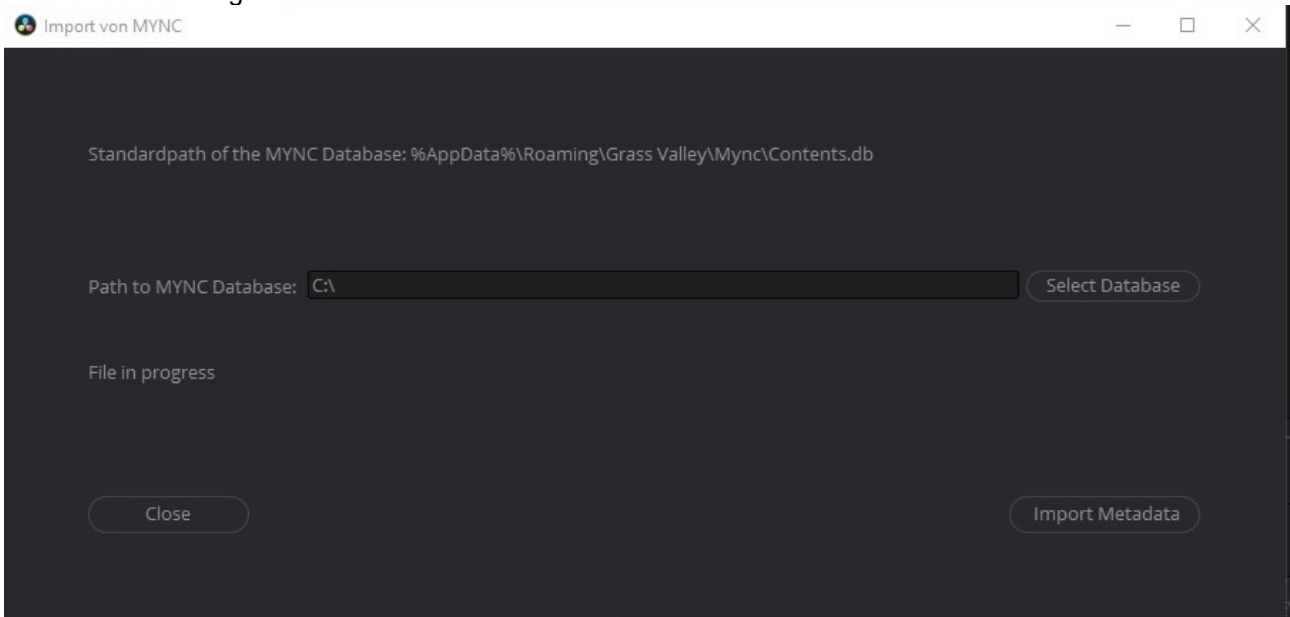
Man kann aber auch einfach DVR 16 Studio starten und beliebige Videoclips oder Audioclips, die in Mync schon mit Schlagwörtern versehen sind, importieren.

Es können auch Bilder importiert werden, diese werden aber vom Script nicht berücksichtigt.

3. Das Script „Mync nachDVR“ über Menü Workspace → Scripts →Mync nach DVR starten.



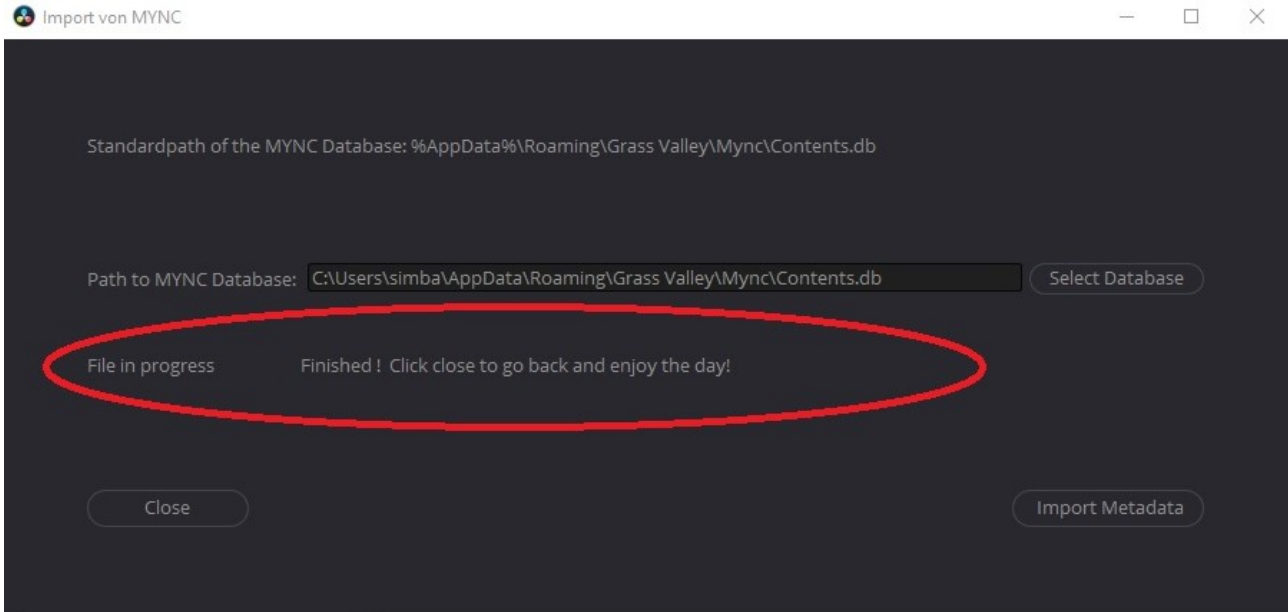
4. Es öffnet sich folgendes Fenster.



Über den Button „Select Database“ muss die Mync Datenbank „Contents.db“ ausgewählt werden. Standardmäßig liegt diese im „C:\Users\username\AppData\Roaming\Grass Valley\Mync\Contents.db“

5. Nun nur noch den „Import Metadata“ Button drücken und dann geht der Import los.

6. Wenn alles erledigt ist, kommt unter „File in progress“ die Meldung „Finished! Click close to go back...“. Das machen wir jetzt und der Import ist abgeschlossen.



Tipps

Um die Ansicht der Metadaten übersichtlicher zu gestalten, habe ich mir ein eigenes „Metadata Presets“ angelegt.

Das geht ganz einfach unter Menü „DaVinci Resolve ->Preferences.. -> User ->Metadata.

Das Ganze sieht bei mir so aus.

Leider habe ich keinen Weg gefunden um die Anzeige der Metadaten von Default auf das eigens erstellte „Metadata Presets“ dauerhaft einzustellen. So muss ich jedes Mal bei einem Neustart von Resolve zuerst von Default auf meine Metadata Ansicht umstellen. Nicht so schlimm, aber etwas lästig.