

Data transfer between Mync and DaVinci Resolve

with

Mync_nach_DVR.lua

Table of contents

System requirement

What can the script

Installation

Use Script

Tips

System requirement

Windows 10,
DaVinci Resolve 16,
Mync v 1.6.0xxxx Standard / Basic I'm not sure, but should work
Sqlite3.exe (Command-Line Shell)

The script is written in Lua and is executed directly in DaVinci Resolve.

I also got it to work with DaVinci Resolve 15 Studio, but it's a bit more cumbersome, because a script is only processed via the menu in the Fusion tab. But you can copy the whole script via the console. Unfortunately in DVR 15 Studio the SmartBins are not automatically connected to the keywords. Must then be done manually.

Since most of them will probably update to DVR 16 in the near future, I will only describe the handling with DRV 15 Studio in case of a larger demand.

If Mync Basic is also supported, I can't say, because I don't know if it is based on the same database. But I guess so, but I'm not sure. Just try it, nothing can happen, because the database is read only.

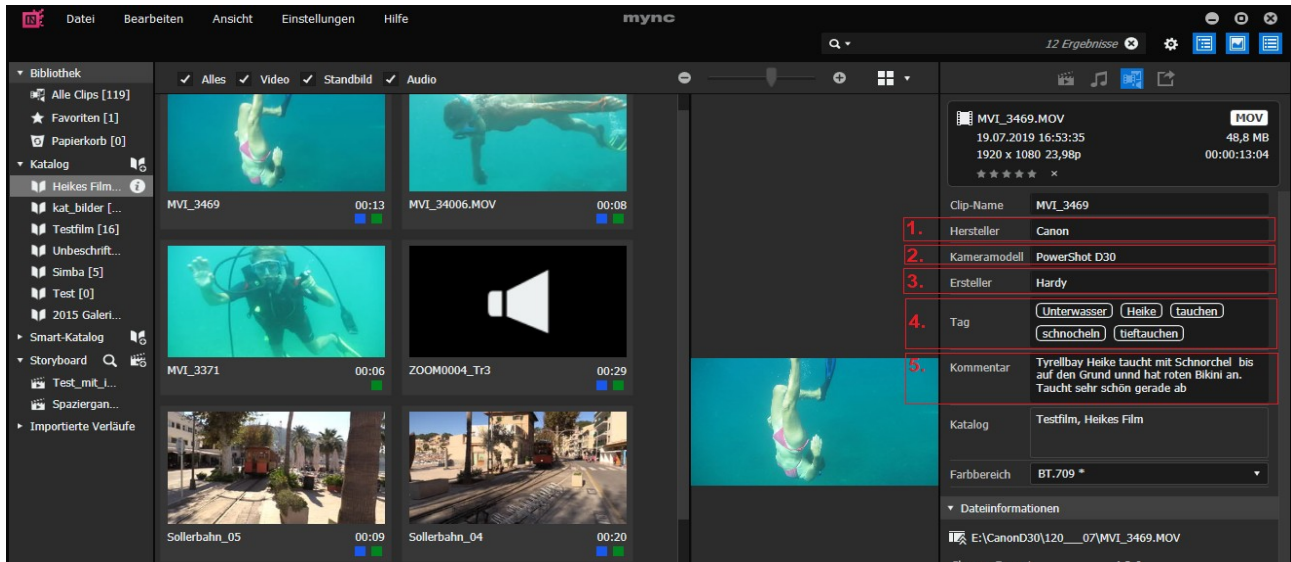
With slower computers, the background flickers during script execution, this is normal because a DOS window is opened briefly and closed again immediately during database queries.

What can the script

The script "Mync_nach_DVR.lua" reads the keywords "manufacturer", "camera model", "creator", "tag" as well as the "comment" from Mync and saves them in DaVinci Resolve (abbreviated as DVR below in the text). Prerequisite is that the video and audio clips are already loaded in DVR.

What I didn't implement (because for me unimportant) are pictures, which are not considered, because DVR always changed the name of these something to be able to represent picture frequencies.

I'll show you first what it can do and below I'll explain the installation and handling, which is really easy.



Above we see Mync and below DVR.

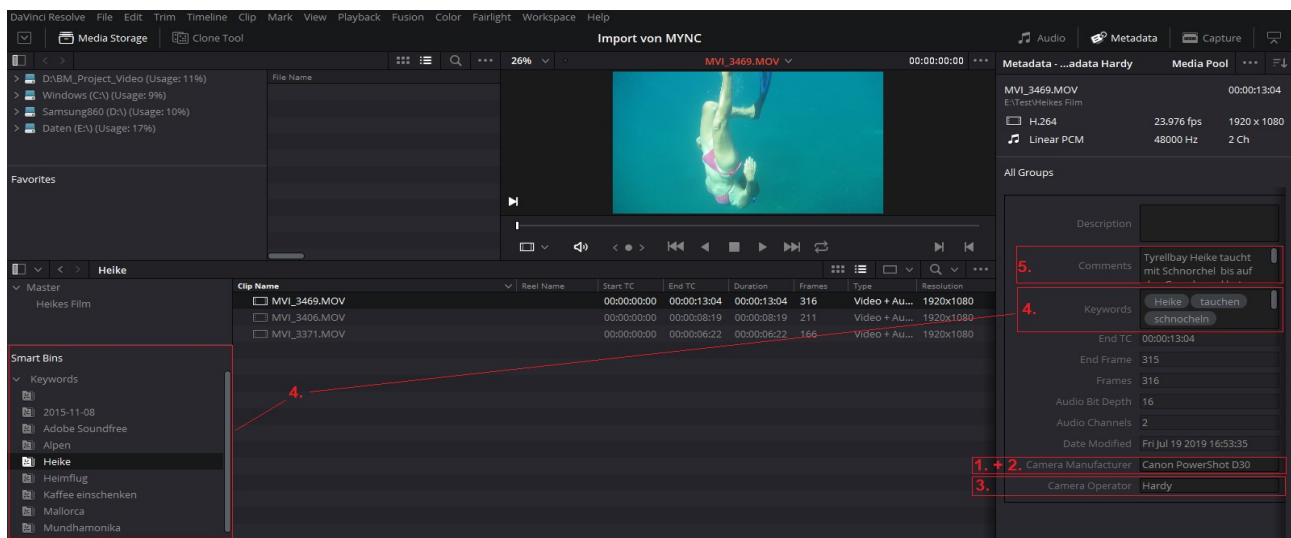
The fields 1. Manufacturer and 2. Camera model in Mync are combined in DVR and stored under "Camera Manufacturer". Order is first manufacturer and then camera model.

The field 3. Creator in Mync is saved in DVR under Camera Operator.

The field 4. Tag in Mync is stored in DVR under Keywords. Each tag is stored under a separate keyword. In DVR each keyword is also listed under Smart Bins.

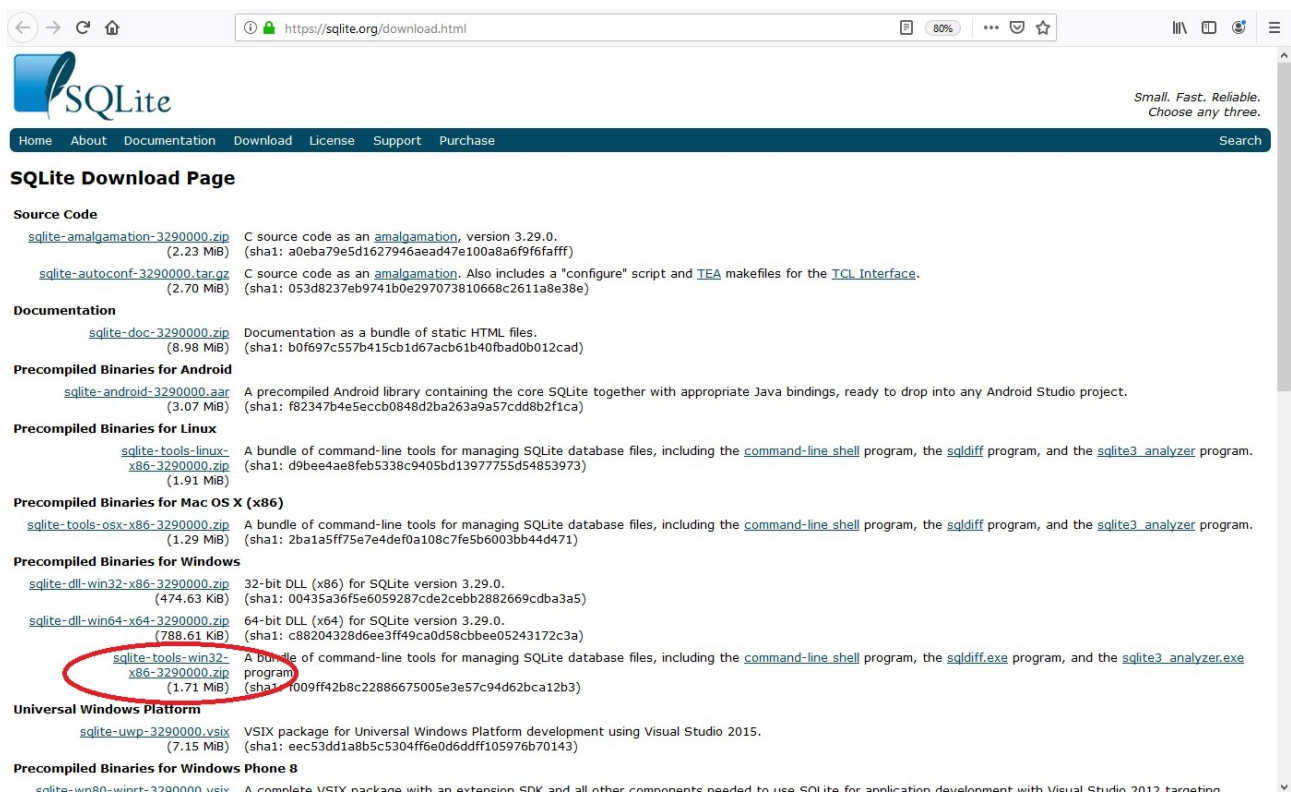
In addition, an empty keyword is inserted so that all media can be displayed very easily. (see top keyword below left.)

The field 5. Comment in Mync is stored in DVR under Comments.



Installation

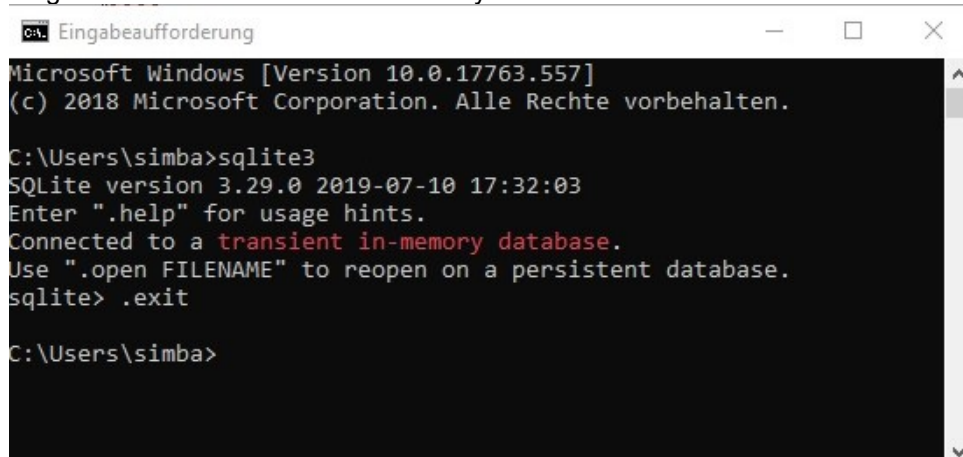
1. After the script "Mync_nach_DVR.lua" is downloaded, it will be inserted into the directory (I'm assuming a standard installation of DVR here) "C:\ProgramData\Blackmagic Design\DaVinci Resolve\Fusion\Scripts\Comp". If the directory "C:\ProgramData" is not visible, check the box "Hidden elements" under the category "View" in Windows Explorer.
2. To check if the correct directory has been selected, briefly look in DVR in the menu "Workspace -> Scripts". There it must be to be seen after copying
3. download "sqlite3-tools-win32-x86-3290000.zip" (see below) and unzip. Copy the file sqlite.exe into a directory and set the system variable "Path" to find Sqlite3.



The screenshot shows the SQLite Download Page. The page title is "SQLite Download Page". The navigation menu includes Home, About, Documentation, Download, License, Support, and Purchase. The page is divided into several sections:

- 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: 1009ff42b8c22886675005e3e57c94d62bca12b3)
- Universal Windows Platform**
 - [sqlite-uwp-3290000.vsix](#) (7.15 MiB): VSIX package for Universal Windows Platform development using Visual Studio 2015. (sha1: eec53dd1a8b5c5304ffe0d6dff105976b70143)
- Precompiled Binaries for Windows Phone 8**
 - [sqlite-wp8-wint-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 Windows Phone 8.

4. Check if the sqlite3 can be started from the "command prompt" in any directory. With ".exit" you close the program again and return to the DOS level. Pay attention to the dot before exit.



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

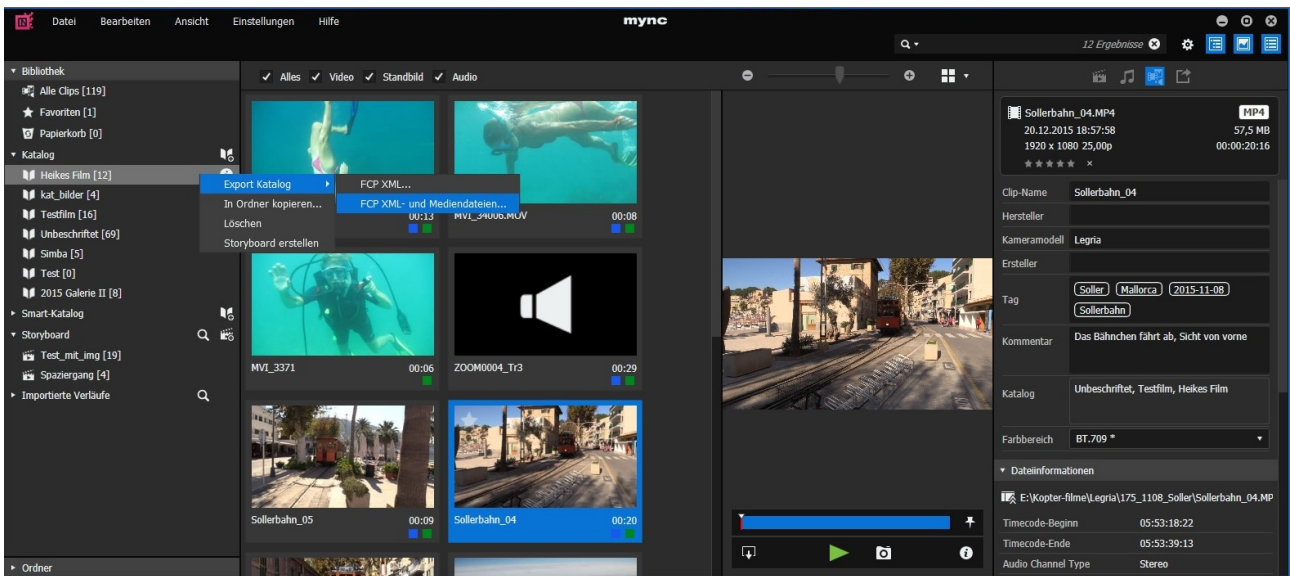
C:\Users\simba>sqlite3
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .exit

C:\Users\simba>
```

Once this is done and works, the installation is complete.

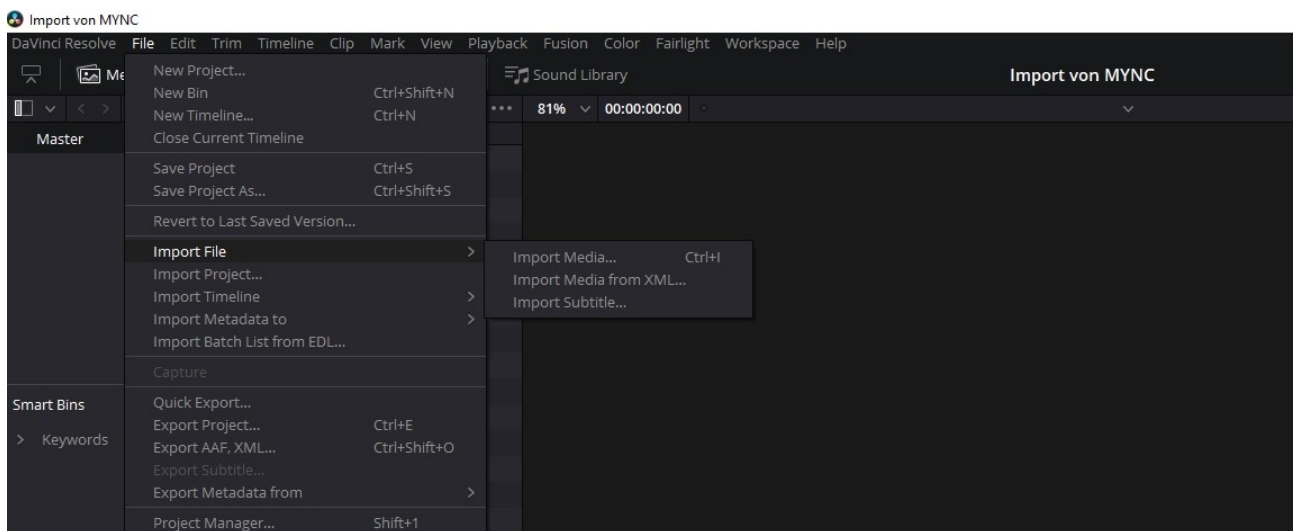
Use the Script

1. You can simply start DVR 16 Studio and import any video clips or audio clips that are already tagged in Mync. If you like, you can also simply export a catalog or storyboard as "FCP XML and Media Files..." from Mync.

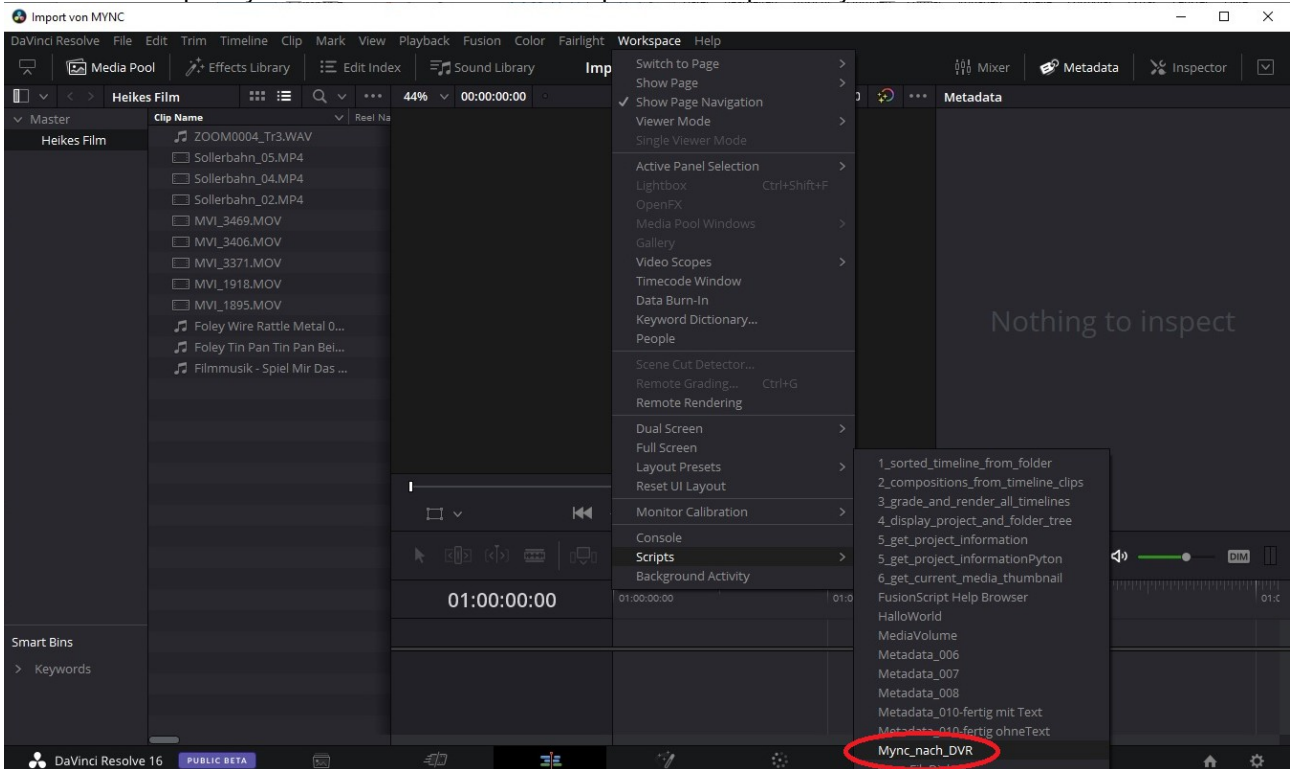


2. Once you have gone the catalog export way, open In DVR the Mediapool or the Edit window and select in the menu "File→ ImportFile →Import Media from XML" to import the catalog or the Storyboard.

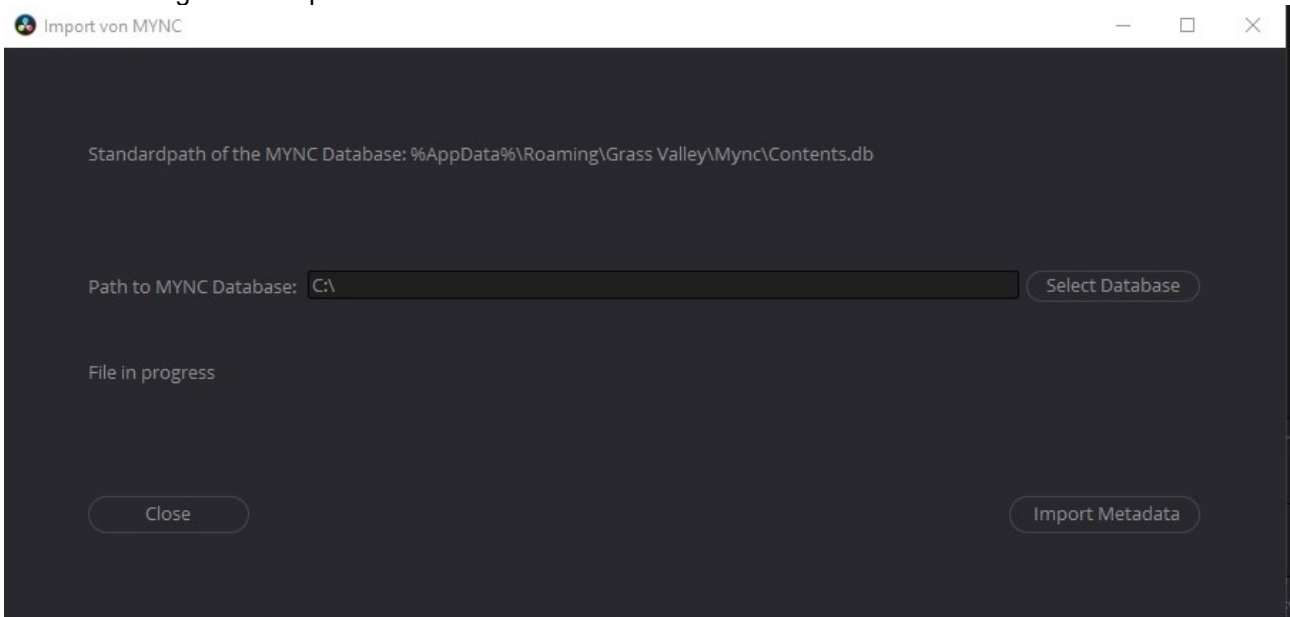
Otherwise you can skip this point



3. Start the script "Mync to DVR" via menu Workspace → Scripts → Mync to DVR.



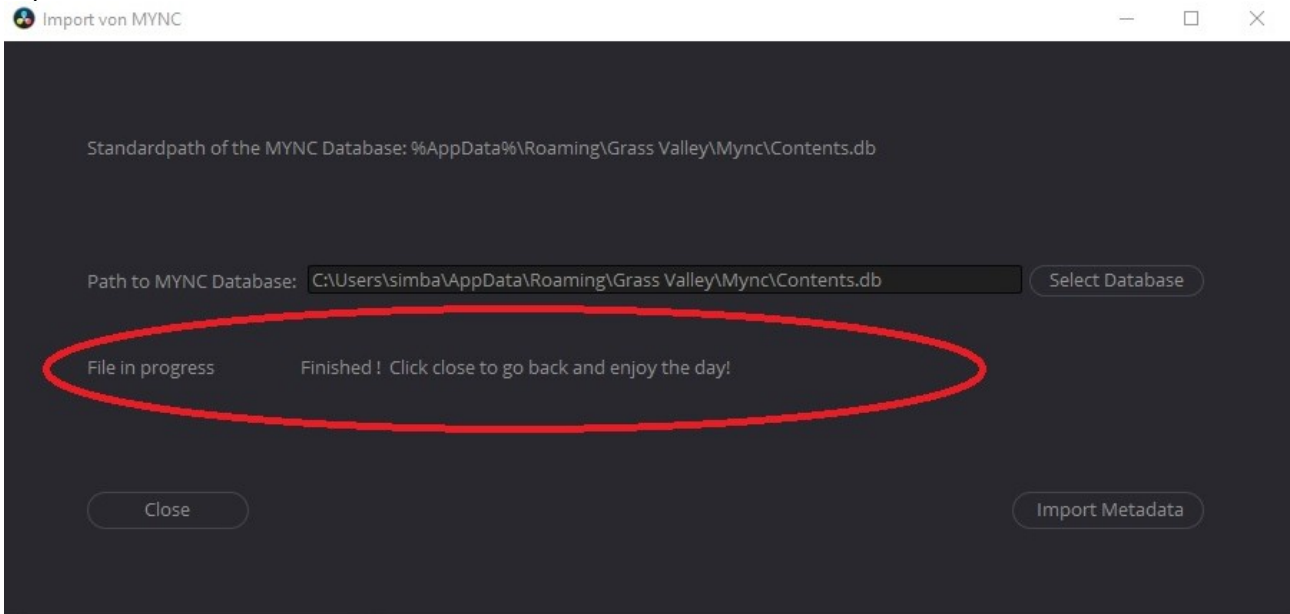
4. The following window opens.



The Mync database "Contents.db" must be selected via the button "Select Database". By default it is located in the "C:\Users\username\AppData\Roaming\Grass Valley\Mync\Contents.db".

5. Now just press the "Import Metadata" button and the import will start.

6. When everything is finished, the message "Finished! Click close to go back...". We do this now and the import is finished.



Tips

In order to make the view of the metadata more visible, I have created my own "Metadata Presets".

This is very easy under Menu „DaVinci Resolve ->Preferences.. → User →Metadata.

The whole thing looks like this to me.

Unfortunately I couldn't find a way to permanently set the display of the metadata from Default to the specially created "Metadata Presets". So every time I restart Resolve I first have to switch from Default to my Metadata view. Not so bad, but a bit annoying.

Maybe someone of you has a hint for me