Automating MakeMKV with AppleScript

It's only days since I wrote about a custom script action upon inserting a DVD in OS X, allowing me have either run the DVD Player or MakeMKV to rip the DVDs.

Now I was digging around if I could do some automation on the GUI part, which indeed turned out to work ... at least to a certain degree.

Let's face it: Ripping DVDs with MakeMKV is simple and easy, however its scripting abilities s*** ;-)

There is actually some limited CLI support available. If you dive into the application package, you'll find the **/Applications/MakeMKV.app/Contents/MacOS/makemkvcon** program, which can be scripted somehow.

In the first place, I was perfectly able to rip off a complete DVD using the CLI command. But that was not what I had in mind, because I wanted to select only the main movies, and there only a limited set of audio streams and subtitles.
As I seemed to be unable to figure out, how that would be possible (some rare comments on the MakeMKV forums mentioned, that it would NOT be possible), I ended up using the GUI.

Well, after I wrote the auto-selector script, I started thinking on how to make this a bit more practical.
I know Apple released support for GUI (graphical user interface) scripting already years ago, but I had never really looked into until today ;-)

After some reading over there at macosxautomation.com I started on extending my script.
Please consider that you need to enable support for assistive devices, as mentioned on macosxautomation.com -- otherwise the script will simply fail.

With help of UI Browser, a utility to explore GUI elements of any application, I was able to identify the relevant items.

So with the latest additions, the script does the following:

- Runs whenever a DVD is inserted
- Automatically run DVD Player if no action is taken - or -
- run MakeMKV if requested to do so
- Wait for the DVD TOC (table of contents) to be read initially
- As soon as the TOC is read, open the DVD in MakeMKV
- Wait until the tracks selected were ripped and try to eject the DVD afterwards

And here's the code:

```
# dvd-action
# ---
# an AppleScript utility to select DVD Player or MakeMKV
# upon inserting a DVD
# ---
# released "as is" under the terms of GPL v2.
# Copyright © 2011 Gianpaolo Del Matto
#
```

```
# r0.1 initial release 2011/12/09
# - simple selector with default action after timeout
#
# r0.2 2011/12/15
# - added limited GUI scripting functionality for MakeMKV
#
# r0.3 2011/12/16
# - added support to check for assistive device access
# (code borrowed from http://www.macosxautomation.com)
# - added simple progress indicator while waiting for drive
# do become available for access
#



# initialization
#
# true : show progress dialogs, false: hide progress dialogs
set verbose to true




# kindly borrowed from
# http://www.macosxautomation.com/applescript/uiscripting/index.html
# make sure that support for assistive devices is enabled
#
tell application "System Events"
 if UI elements enabled is false then
  tell application "System Preferences"
   activate
   set current pane to pane id "com.apple.preference.universalaccess"
   display dialog "This script requires access for assistive devices be enabled." & return & return & "To continue, click the OK button and enter an administrative password in the forthcoming security dialog." with icon 1
  end tell
  set UI elements enabled to true
  if UI elements enabled is false then
   display dialog "This script cannot run while access for assistive devices is disabled." & return & "Exiting now." buttons {"OK"} with icon 2
   return "user cancelled"
  end if
 end if
end tell



# check if DVD Player or MakeMKV are already running
# if not, present a little menu, that allows to select
# which application should be run
#
if not appIsRunning("DVD Player") and not appIsRunning("MakeMKV") then

 with timeout of 3600 seconds
```

```
  display dialog "Please choose your desired action:" with title "Action for DVD" buttons {"Cancel", "play DVD", "rip DVD"}
cancel button "Cancel" giving up after (5)
 end timeout
 if button returned of result is "rip DVD" then
  if verbose then display dialog "Starting MakeMKV" buttons {"OK"} giving up after 2
  tell application "MakeMKV" to activate
  repeat
   if application "MakeMKV" is running then exit repeat
   delay 5
  end repeat

 else if button returned of result is "Cancel" then
  return
 else
  if verbose then display dialog "Starting DVD Player" buttons {"OK"} giving up after 2
  tell application "DVD Player" to activate
  return
 end if

end if



# check again if MakeMKV is running
# this time, attempt to automate these steps:
# - automatically open the DVD as soon as the drive is ready for access
#
if appIsRunning("MakeMKV") then
 # now tell MakeMKV what to do
 tell application "System Events"
  tell process "MakeMKV"
   # in some rare cases, the menu bar items will stay "greyed out"
   # even if a DVD was inserted
   # the culprit: "AXEnabled" property of the menu will be TRUE even then,
   # although the menu items are not clickable
   # There seems to be no way to check, if they're actually active or inactive
   # However, as a workaround, sending CMD+O to fire up the "File - Open" dialog
   # seems to work, and yet: just opening and closing it will enable the menu bar
   #
   set _retval to false
   repeat until (_retval) is true
    try
     tell application "MakeMKV" to activate
     keystroke "o" using {command down}
     delay 1

     if (exists window 1) and (exists window 2) and (value of attribute "AXRoleDescription" of window 1) = "dialog" then
      click button "Cancel" of window 1
      set _retval to true
     end if
    end try
    delay 1
   end repeat


   # note on the dvd drive selection and menu handling:
   #
   # because we don't know the name of the DVD device,
   # we simply address it by the id of 'menu item 1'
   # so we're just selecting the first device found there.
   # we might have more than one DVD drive, but this is out of scope for now
   #
   # this script was initially triggered by inserting a new disc
   # thus we need to wait until the disc is initially read
   # while this is in progress, our 'Open disc' menu item will be disabled
   #
   set _retval to false
   repeat until (_retval) is true
    try
     repeat while (value of attribute "AXEnabled" of menu "Open disc" of menu item "Open disc" of menu "File" of menu bar item "File" of menu bar 1) is false
      get value of attribute "AXEnabled" of menu "Open disc" of menu item "Open disc" of menu "File" of menu bar item "File" of menu bar 1
      set counter to 5
      repeat while counter is greater than 0
       if verbose then display dialog "Waiting for dvd drive to become ready for use ..." with title "Please wait ?" buttons {"" & counter & ""} giving up after 1
       set counter to counter - 1
      end repeat
     end repeat
     set _retval to true
    on error
     display dialog "error"
    end try
    delay 1
   end repeat

   # seems that the dvd drive is now ready for use, now try to load the disc
   #
   if verbose then display dialog "Trying to load DVD in MakeMKV now ?" giving up after 1

   # now load the drive
   #
   tell application "MakeMKV" to activate
   click menu item 1 of menu 1 of menu item "Open disc" of menu 1 of menu bar item "File" of menu bar 1

   # now stay and wait until another window appears
   # this should be "completed" dialog at the end
   #
   repeat until ((exists window 1) and (exists window 2))
```

```
    delay 15
   end repeat

   # check the type of window 1, which should be a dialog
   # if so, let's close it and try to eject the disk
   #
   if (exists window 1) and (exists window 2) and ((value of attribute "AXRoleDescription" of window 1) = "dialog" or (value of attribute "AXRoleDescription" of window 1) = "standard window") then
    click button 1 of window 1

    click menu item "Eject disc" of menu 1 of menu bar item "File" of menu bar 1
   end if

  end tell
 end tell
end if



# func:appIsRunning
# appName : string : name of application (e.g. "DVD Player")
#
# helper to check if application is running or not
#
on appIsRunning(appName)
 tell application "System Events" to (name of processes) contains appName
end appIsRunning
```

This, as far as it goes, saves at least the additional button click to load the DVD in MakeMKV.
However, there is one drawback after all ...

While examining the MakeMKV GUI, I figured that the main panel was not recognized by UI Browser. It was just reported as **unknown (UI element #4)**. Unfortunately there was no obvious way to get access to the elements within that panel.
This - as of today - effectively limits the use of GUI scripting, as it is not possible to interact with the elements within that panel.
Otherwise, I'd be happy to sit down and write the part, which does auto-selection of the track elements according to some given criteria.