

Manual for batch-encoding subtitles into videos

Tessa Vermeir

Created as part of a project led by Frank Seifart, funded by the lab
'Dynamique du Langage', CNRS, Lyon (France)

January 2019



Table of contents

Introduction	3
1. The beginning: some definitions	3
1.1. Operating systems	3
1.2. Video and subtitle formats	4
1.3. Burning subtitles in batches	4
2. Getting started	4
2.1. MKVToolNix	5
2.2. MKVMerge	5
2.3. UnRar for Windows	6
3. Batch subtitling with a .bat code	6
3.1. Explanation of the code	6
3.2. Setting the user parameters of the code	7
3.3. Running the code	9
3.4. Solving errors	11
4. Some final tips and tricks	12
5. Converting videos to a TLA-accepted format	13
Links to the cited webpages	15

Introduction

This manual will guide you step by step and show you how you can burn subtitle sets into videos in batches. Please read the steps carefully and make sure you don't skip any.

In this manual, we will use a file written in code to encode the subtitles. There are many programs that also let you encode subtitles, among which a couple are freely accessible as well. One program that is often used, is HandBrake. While you can load multiple videos at once, you can't load more than one subtitle set at once, and you will have to select subtitles and set the parameters for each video separately. When you have a great number of videos you want to subtitle, this process is quite time-consuming. Another software program is MKVToolNix, but the same problem persists: it is not possible to encode multiple sets of subtitles into multiple videos at the same time. However, we will use MKVToolNix with our code; this manual will show you why and how.

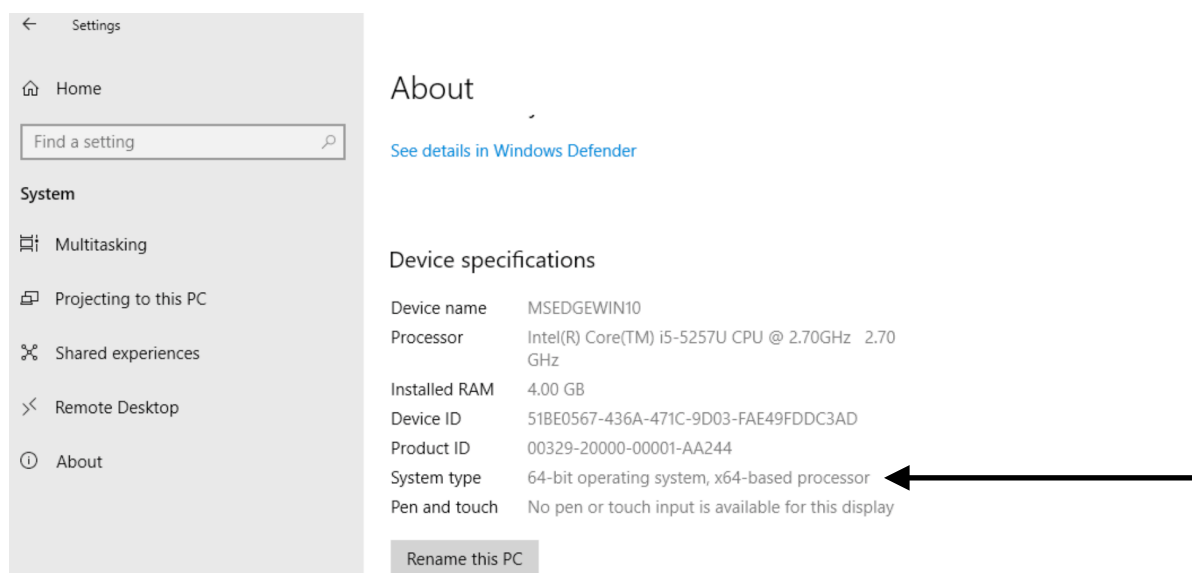
The first section explains what all the different types of files are that will be used for the encoding. The second section guides you through all the downloads and programs you need to have installed on your computer before starting. Section 3 walks you through different parts of the code and explains the most frequent errors and mistakes. Section 4 finishes with some tips and tricks.

1. The beginning: some definitions

1.1. Operating systems

First of all, the program used in this manual is only available for Windows. If you're working on a Linux or Mac OS system, the best solution is to try and find a Windows computer you can work with. Otherwise, there are solutions like VirtualBox that let you run Windows via another operating system. The operations presented in this manual also work in VirtualBox; however, we will not go into the details of its installation and use here. For more information on VirtualBox, see their website.

Since you'll have to install some programs, it is important to know which type of Windows operating system you have. Generally, downloads are available for Windows 64-bit and for 32-bit. To see which one your computer is using, click on the Windows-sign in the down left corner; go to "settings"; scroll all the way down in the menu to click on "about". Then you should see a page called "Device specifications" (or something similar). The system type should be noted there, like on the picture below. Here, the arrow shows that it is a 64-bit operating system.



1.2. Video and subtitle formats

As you probably know, videos come in different kinds of formats. The most common extension is .mp4: this format will work on any computer using any program (in the majority of cases). However, there also exist other types of extensions that are also more or less common. One of the bigger ones is the .mpg or .mpeg. Not all video programs recognise this extension. Curiously though, it is this extension that we are going to use for the batch encoding of subtitles. The reason for this is mostly practical: the software that we use for the encoding only accepts .mpeg files. Other programs do accept other extensions, but don't have the option of batch encoding. Even if you first have to transform .mp4-videos into .mpeg/.mpg-videos, when you have a big number of videos, this is still the easiest way. There are many sites that offer free online conversion from one format into the other. However, you will never really know what happens with your files if you upload them to a website; keep this in mind. Alternatively, some open-source (free) downloads are also available.

Subtitles also come in all different kinds. The most commonly used extension is .SRT. This format also contains information on time codes: when each subtitle will appear, and when it will disappear from the screen. If you are going to use subtitles that you created yourself, make sure to save them with this extension. When the encoding program "sees" this kind of file, it will automatically take over the time codes. You can choose the format of your subtitles when you export them from the program used to create them.

1.3. Burning subtitles in batches

The goal of this manual is to teach you how to burn subtitles into videos. When you burn subtitles, they become part of the video. Each time the video is opened, the subtitles will also appear and cannot be turned off. This process is called "burning" or "hard-encoding". In some cases, it is preferable to have the subtitles burned into the video; you now have only one file with both, and there is no need anymore to manually select the subtitles you want to have displayed on your screen.

When you burn in batches, this simply means that you burn a number of subtitles simultaneously. The trick is to tell the computer to burn certain subtitles with a certain video, and not with any other video. The goal of this manual is to show how to do this: burn subtitles automatically with the corresponding videos, in large numbers and simultaneously.

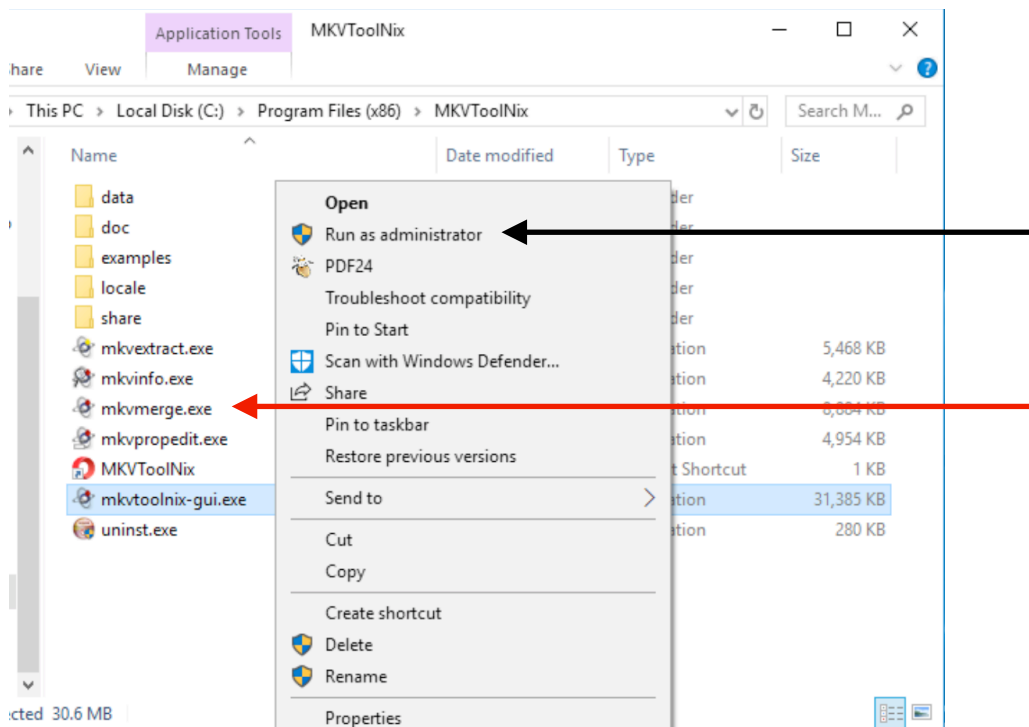
2. Getting started

When you want to encode subtitles in batches, you will need to have a number of programs installed on your computer before starting. This section will treat each of the programs and show how to install them. It is not necessary to know how to use the programs; they only need to be installed correctly.

Note When you finish downloading a program or file, make sure to save it on a location on your computer other than the Downloads file. A good place to store everything is on the local disk C. This will help you later when you need to find the programs again.

2.1. MKVToolNix

The first program you will need to install on your computer, is MKVToolNix. You can download the installer from [this webpage](#)¹. Search for the Windows version that correspond with your computer and install the program. After the installation is complete, run the program to check that it is installed correctly. If it opens and works, the installation is successful, and you can continue to the next step. If it doesn't work, verify that you installed the correct version, and that you don't have some kind of firewall activated that blocks programs downloaded from the Internet. Sometimes programs don't run correctly if you just open them; try rather the "run as administrator" option by right-clicking on the program icon (see picture below, black arrow).



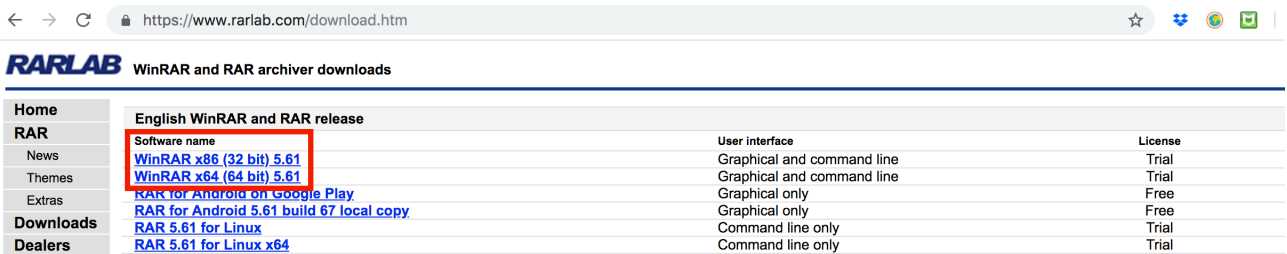
2.2. MKVMerge

The second thing you need is actually not a program, but a .exe file that should be included in the MKVToolNix download. Open MKVToolNix in your File Explorer and check if it's there. In the image above, the file is indicated with the red arrow.

¹ For those who have a Mac: there is indeed a Mac-version available of MKVToolNix. However, this version is not supported, and on top of MKVToolNix, we will need another program that does not exist for Mac. Windows really is the only way to go!

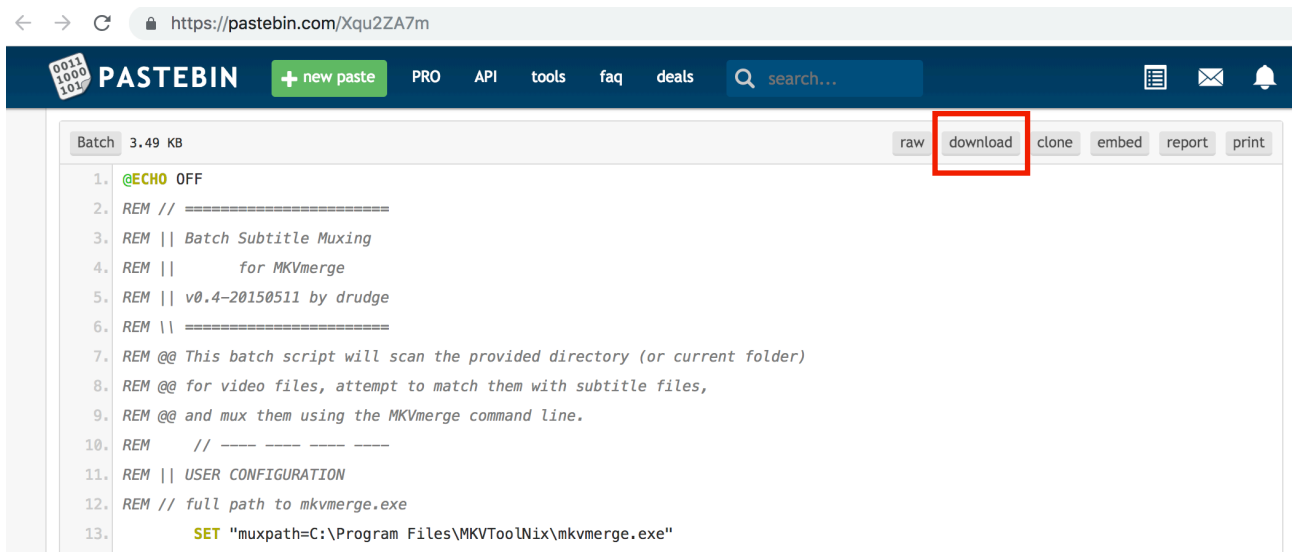
2.3. UnRAR for Windows

The last program you'll have to install is called "UnRAR for Windows". You can download it via [this link](#). From this download page, please select either WinRAR x86 (if you have a 32 bit Windows OS) or WinRAR x 64 (if your Windows is 64 bit). UnRAR will be used by our code to extract the text in the SRT subtitle sets.



3. Batch subtitling with a .bat code

The actual batch encoding will be done by a piece of code, a tiny program in a way. It is available online [here](#) (credits to "Drudge"). Download the code to your computer and verify that it is saved with a .bat extension. (This means "Windows batch file"; what it does exactly is not important for our purposes. However, if you're interested in learning more about them, [here's](#) a tutorial.) Give the file a name with which you will find it again easily.



The rest of this section will walk you through the set-up of the code, since you will need to set a number of parameters for the code to work. Everything you need to do is also explained in the code; if you understand everything, you can skip this section.

3.1. Explanation of the code

The code that does all the subtitling work actually uses all the programs that section 2 told you to install. It calls on the "merge"-function of MKVToolNix (for which you need the .exe-file) to mux videos and subtitles. This means that the subtitles are burned into the videos, called "muxing". You can do this by hand using MKVToolNix. However, this program won't allow you to mux in batches,

that's why we need this piece of code. You can ignore most of the lines in the code; we will only look at the lines in which you need to set the parameters in order to make the code work on your computer.

3.2. Setting the user parameters of the code

When you download and open the code (using Notepad for example; right-click on the file and choose "open with Notepad"), you will see a couple of different sections. We will first look at the first section (lines 11 to 31) in which the code allows you to set the parameters of your choice. This is indicated with "user configuration" in line 11 (see also image below). First, you will need to "tell" the code where it can find the programs and files it needs to work. These are indicated in lines 13 and 16 (see also the picture).

```

11. REM || USER CONFIGURATION
12. REM // full path to mkvmerge.exe
13.     SET "muxpath=C:\Program Files\MKVToolNix\mkvmerge.exe"
14. REM \ \ -- -- -- -- --
15. REM // full path to unrar program, make sure to include switches
16.     SET "rarpath=C:\Program Files\7-Zip\7z.exe"
17.     SET "rarcmd=e"
18. REM \ \ -- -- -- -- --
19. REM // output path (with trailing slash)
20. REM ++ leave blank for working directory
21.     SET "outputdir="
22. REM @@ default: SET "outputdir="
23. REM \ \ -- -- -- -- --
24. REM // prepend text to output filename
25. REM ++ REQUIRED if outputdir is left blank
26.     SET "fileprefix=subtitled-"
27. REM @@ blank: SET "fileprefix="
28. REM \ \ -- -- -- -- --
29. REM || END USER CONFIG
30. REM \ \ ---- -- -- -- --
31. REM -- editing below this line should be done precisely. (here thar be dragons)
32. REM =====

```

You will only need to modify after the " = " signs. Here, you can paste the path to where you saved the programs and files that the code needs to use. To get this path, follow these steps: open a File Explorer; navigate to where you saved the downloads; CTRL + Shift + click on the icon of the program (see image below); choose "copy as path". Navigate back to the Notepad file with the code in it, and paste your path after the " = " sign. For example, the path on my computer looks like this:

"C:\Program Files (x86)\MKVToolNix\mkvmerge.exe"

Copy the paths asked for in lines 13 and 16. Ignore line 17; you don't have to change anything here.

The next step concerns the output. The code will save the subtitled videos separately onto your computer – i.e. the originals will not be overwritten. You'll just have to tell it where to store the output and what to call each video. These parameters are set in lines 21 and 26.

```

11. REM || USER CONFIGURATION
12. REM // full path to mkvmerge.exe
13.     SET "muxpath=C:\Program Files\MKVToolNix\mkvmerge.exe"
14. REM \ \ - - - - -
15. REM // full path to unrar program, make sure to include switches
16.     SET "rarpath=C:\Program Files\7-Zip\7z.exe"
17.     SET "rarcmd=e"
18. REM \ \ - - - - -
19. REM // output path (with trailing slash)
20. REM ++ leave blank for working directory
21.     SET "outputdir="
22. REM @@ default: SET "outputdir="
23. REM \ \ - - - - -
24. REM // prepend text to output filename
25. REM ++ REQUIRED if outputdir is left blank
26.     SET "fileprefix=subtitled-"
27. REM @@ blank: SET "fileprefix="
28. REM \ \ - - - - -
29. REM || END USER CONFIG

```

Here, you have a couple of options.

1) In line 21, you can choose to save the output videos in a specific file folder. For example, I created a folder called “mkvmerge_out”. The name of the folder is not important; you only need to make sure that you copy the path correctly after the = sign.

IMPORTANT If you choose to copy the path of a folder where the output will be stored, be sure to put a backslash (or trailing slash; “\”) at the end of the name – even if the path to this folder does not finish with a backslash. If you don’t do this, the code will not recognise the path as a folder, and will return an error message if you try to run the code.

2) Alternatively, you can choose to not specify an output directory. This means that your subtitled (muxed) videos will be saved in the same folder as the originals. This is not necessarily good or bad; but beware that the non-subtitled videos will be in the same location as the muxed ones.

IMPORTANT If you decide to save the output in the same location as the input videos, you are obliged to fill out the parameter in line 26. This line will add a prefix to the titles of your muxed videos. For example, I chose the prefix “sub_”. If you don’t do this, the code will not be able to save the muxed video, since there already exists a video with the same name in the same folder. The code is not able to overwrite the original video in this case, and you will get an error message.

Of course, if you choose to save the output in another folder, you can still use a prefix. Doing so may help you distinguish more clearly between muxed videos, and videos that don’t have burned-in subtitles.

Finally, the last line where you need to copy a directory path is line 39 (see picture on next page). In this line, you will tell the code where you stored the videos and subtitle files that it will

need to mux: here it gets its input. You don't have to change anything in line 42; for our purposes, it is enough to set the input path in line 39.

```

38. REM @@ default working path
39. SET "wp=." ←
40.
41. REM @@ optional working path via argument
42. IF EXIST "%~1" SET "wp=%~1"
43.
44. REM @@ ready steady go!

```

One last change to be made is in lines 69-71 (see image below). Here, you can tell the code which kind of video extensions it has to search. There are already two extensions: .mkv and .mp4. The last one doesn't work optimally, and the first is a less common extension. If you do have videos in one of these formats, you can still try them of course. Otherwise, if you want to use a .mpg or .mpeg extension, you'll have to add these manually to the code. The easiest way to do this is to copy line 70, then press Enter, and paste everything on line 71. Modify the ".mkv"-part into ".mpg" or ".mpeg". If you want to include both, make two new lines, one for each of them. Make sure not to change anything else, and that the markup is the same as on line 70. Tabulation and spaces are important in code 'language', just like final dots and other signs, and if there is a mistake somewhere, this will likely lead to a fail – there won't be any muxed videos in the output.

```

67. :getfiles
68. FOR %%I IN ("%wp%\*.avi",
69. REM "%wp%\*.customVideoExtension",
70. "%wp%\*.mkv",
71. "%wp%\*.mp4") DO (
72. REM @@ found a video file, now check for subtitles
73. CALL:getsubs "%%~I"
74. )

```

3.3. Running the code

When you've set all the user parameters, it's almost time to try and run the code. First, however, it is important to verify a couple of things.







When you run the code, it will go to the input directory you directed it to (line 39) and look for videos and subtitles. It is very **important** that videos and their associated subtitles have **the exact same name**. If this is not the case, the code will not be able to match subtitles to videos, and nothing will be muxed. You can give a video and a subtitle set the exact same name without problem; they don't have the same extension, so you will not get an error. For example, in my video and subtitle database I have these names²:

² In general, it is not very important if you have spaces in your document titles. However, when working with code like the one in this manual, spaces can make a big difference. Code is like a language: a space means the start of a new word or entity, just like when you're writing a text. Therefore, when using code, it is better to use a sign other than spaces to separate words. I used the underscore "_", but you can also use "-" for example, or nothing at all of course (i.e. glue all the words together).

“The_Ice_Princess.mpg”

“The_Ice_Princess.srt”

If you want to have two or more subtitled versions of the same video, e.g. because you have subtitles in different languages, there are two options. Either you duplicate the video, and rename each version according to the language of the subtitles that will be burned in; or you mux the subtitle sets one by one. The first option would look something like the image below.

Name	Date	Type	Size	Length
 mkvmerge_out	1/9/2019 6:37 AM	File folder		
 mkvmerge_subtitle_muxing.bat	1/9/2019 8:16 AM	Windows Batch File	4 KB	
 The_Ice_Princess_EN.mpg	10/25/2012 6:57 AM	MPG File	30,966 KB	00:02:33
 The_Ice_Princess_EN.srt	4/28/2011 8:07 AM	Subtitle File	3 KB	
 The_Ice_Princess_FR.mpg	10/25/2012 6:57 AM	MPG File	30,966 KB	00:02:33
 The_Ice_Princess_FR.srt	4/28/2011 8:07 AM	Subtitle File	3 KB	

When you run the code and it finds these files, it will match the French .srt with one copy of the associated video file (i.e. the _FR.mpg), and the English .srt with the other copy (i.e. the _EN.mpg). If you decide to do each version separately, it is better to store the subtitle set you don't use somewhere else. That way, the code will only find one video file and one subtitle set with the same name. When you're done with set one – say, the French one, you can then (re)move the French subtitles and put in the English ones, before running the code a second time.

ATTENTION If you choose option two (i.e. muxing one after the other different languages), beware that you add a prefix to the muxed video of the output, especially if you keep the original video files and the output video files in the same folder. This will avoid getting errors, and also makes it easier for you to recognise originals versus muxed videos. In any case, if you run the code on the same video more than once using different subtitle sets, you will have to add some kind of distinctive feature to each output; if there already exists a video file with the exact same name, the code will not be able to save the second muxed version.

If you have a lot of different videos to mux with subtitles, the easiest solution is probably to create a special output folder. That way, you know for sure which videos have the burned-in subtitles, and you will never get the error that there already is a video with the same name (i.e. the original input video).

When everything is set, run the code by clicking right on the file icon and choosing “Run as administrator”. A pop-up will likely appear, asking if you want to allow the program to make changes to your computer. Click yes – after all, we need to give the code permission to create new videos with our subtitles burned into them, and to save those on our computer. Another pop-up will then appear, that looks like the command line screen (for those of you who are familiar with that interface). It's a black background with square, white letters. If all goes well, you don't have anything more to do than wait and let the code work its magic! – otherwise, continue to the next section, where the possible errors are explained and solved.

3.4. Solving errors

Sometimes, even when having carefully set the parameters, something can go wrong, and you will get an error message. Luckily, the code will tell you if something goes wrong, and more importantly, why. This last section will show which errors you could get, and how to solve them.

1) ERROR: empty [fileprefix] setting requires [outputdir] to be set

If you get this message, you didn't select an output folder nor added a prefix to the output video files. This means that the code has to save the muxed video under the same name as the original in the same folder. As said before, this is not possible: the code cannot overwrite a file with another file having the exact same name. To solve this error, either change the output directory, or add a prefix to the output videos (or both). See also page 8.

2) ERROR: [%%~n].idx] -- Missing .sub file

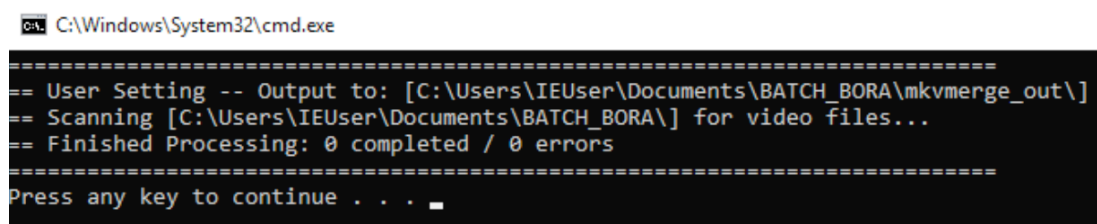
This type of error is a little bit harder to solve quickly. It means that something is wrong with the UnRAR program, which is supposed to extract the text from the .srt-file you want to use. When this error comes up, check if UnRAR is really installed correctly on your computer. You can do this by running the program. If you're able to run it correctly, try copying the directory to the program again into the code. It may be that something went wrong the first time. Alternatively, you can uninstall UnRAR from your computer, and re-install it again. If you do this, check again to see which operator version of Windows your computer has, and make sure that you download the corresponding version of UnRAR (see also pages 3 and 6 respectively).

3) ERROR: [%~n1%~x1] -- Existing output file

This message comes up when you already have a file with the same name in the output folder. For example, you didn't add a prefix to the output video and tried to save it in the same folder as the original. Alternatively, maybe you accidentally tried to mux the same video twice, and you already have a muxed version; or you want to subtitle the same video a second time but with a different set, and forgot to add distinctive features to each of the versions. To solve this error, make sure to make changes in the names of the files in such a way as to be able to distinguish them properly.

4) Finished Processing: 0 completed / 0 errors

It may happen that this message will pop up:



```

C:\Windows\System32\cmd.exe
=====  

== User Setting -- Output to: [C:\Users\IEUser\Documents\BATCH_BORA\mkvmerge_out\  

== Scanning [C:\Users\IEUser\Documents\BATCH_BORA\] for video files...  

== Finished Processing: 0 completed / 0 errors  

=====  

Press any key to continue . . . █

```

If you did deposit videos and subtitles in the correct input folder, it can still be that the code won't mux anything. Multiple causes can lead to this error, and unfortunately, the script doesn't say exactly where the error comes from in these cases.

The most common mistake is probably a difference in the name of the video and its associated subtitle set. When the names are not exactly the same, the muxing code will not be able to

match them together, and will think there is no job to be done – hence the “0 error”-line. Make sure the names are the same, and try again.

This same message will also pop up when there are no video files in the input directory (the working path). You can change this directory in line 39 of the code. Copy the path carefully, and make sure that your files are indeed in that folder.

It can also happen that the code doesn't recognise the extension of your video files. You can change this as well. In lines 70 and 71, there are two extensions made available, and we added one as well; if you wish to add yet another one, follow the same steps (see page 9) and then try again. If the type of video file is not supported by MKV Merge, you will not get an error message specifying this, but you may want to try in that case to first convert your videos to a supported format, and then try again. Formats that will surely work with MKV Merge are .mpg, .mpeg, and .mkv.

If you tried all of the above solutions, and still nothing is being muxed, there is one last option you can try: put everything together in the same folder. Well, everything except maybe the program installers; if those are saved on the local disk C: and the paths are copied correctly, the problem shouldn't arise there. However, when no videos are muxed, a solution can be to put the .bat file in the same folder as the input videos and their subtitles. Change the working path setting in the code (line 39) by copying and pasting the directory to the correct file. Alternatively, you can also leave the original coding: “wp=” and then try and run the code again.

4. Some final tips and tricks

The trickiest part of this way of batch encoding, is setting the parameters correctly in the .bat file. Many things can go wrong, and you don't always know why. Above, we discussed a number of frequent errors and (equally frequent) mistakes. One other very important thing to remember, is to be careful when you modify the code. Like it was said before, code is like written language, but with very strict rules; if you add or remove anything, even spaces and quotation marks, this can make the code crash.

For example, every time you see the word “SET” in a line, what follows has to be within quotation marks. If you get an error that this or that directory or file is not found, it could very well be that there is something wrong with the syntax in the code. Also very important is the backslash “\” after you define the output directory. You have to add this sign so that the code knows you refer to a folder, rather than a single file. The backslash indicates that the directory you refer it to, can accommodate the output videos you tell it to make. Look again carefully if you like to e.g. line 13 in the code: every time there is a \, this means that what is to the left of this \ is the name of a folder, and what is to the right of the \ is some content within that folder.

Even though the code should go to the folder you indicated as input folder to look for videos, sometimes it does not work very well if the .bat file with the code is not in the same folder as the videos. If there is no \ at the end of the input directory, it is never going to work; but the error may persist even if you typed in everything correctly. The easiest solution in that case is to make sure they are all in the same folder. The same goes for the programs that the code calls upon to search and mux videos and subtitles; sometimes it is not enough to download them, and in general it's better to save them on e.g. your local disk C:, and even run them a first time to check that everything works.

Remember also that this code will not make any changes to the format of the video. If you want to change the output format, you will have to use another program. There are many free options to do this. The best is not to do it online on a website, but with a program that you can down-

load and install on your computer. That way, you can be certain that the videos you transform will not be used for other purposes by the website.

Despite all the hassle, once everything is in place and up and working, you'll be happy you don't have to encode everything by hand!

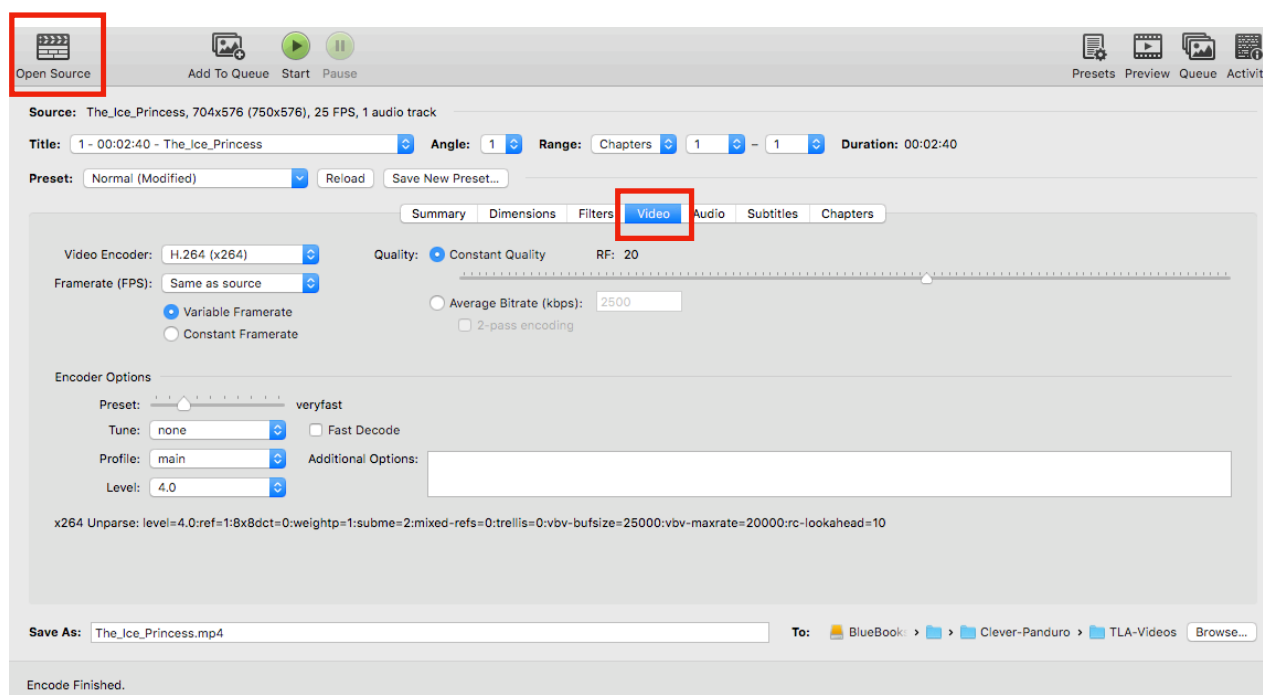
5. Converting videos to a TLA-accepted format

If you created your subtitled videos as part of a language project and you want to make them available on the website of The Language Archive (TLA)³, there is one last step to be undertaken. Luckily, it is not a big one.

In the introduction, the software HandBrake was mentioned. While this program doesn't have a batch subtitle encoding function, we will use it for this final step to transform the MKV Merge output into an mp4-format with a H.264 encoder. It is not really important to know what this means exactly; just that using this encoder, it will be possible to play your video in an online browser. This is necessary when you want to make your material available on TLA⁴.

First, go to the HandBrake [download page](#) (here with .fr extension; if the link doesn't work, search for "HandBrake" using your favorite search engine. Make sure however that you really use HandBrake's own website, as there are other sites as well who offer the software). Download and install the version corresponding to your computer. HandBrake, contrary to other software used in this manual, is also compatible with Mac OS and Linux.

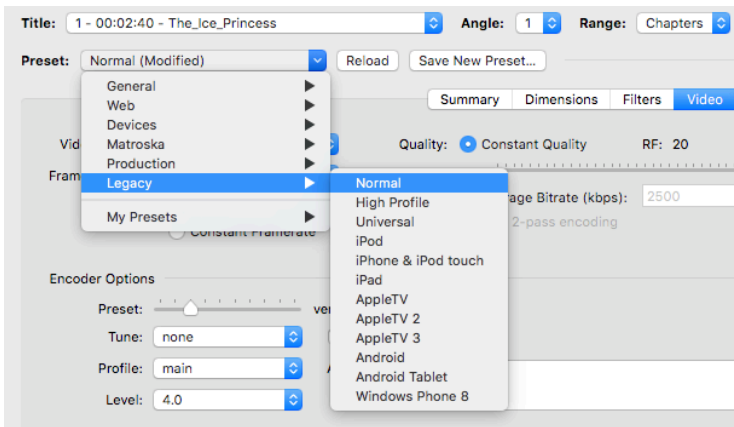
Run the program. Open a video using the clapper board icon in the top left corner saying "Open Source" (see image below). You can also choose to load an entire folder of videos at once. Now, you will have to set the correct parameters. Click on the tab "Video" (see image below). Note that this manual was written on a Mac using the Mac version of HandBrake; the interface might look different on a Windows computer.



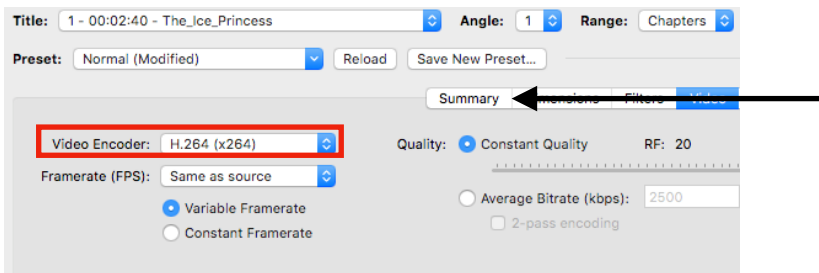
³ See their [guide](#) for instructions on how to upload material.

⁴ For more details on supported formats, see their [webpage](#) listing supported formats.

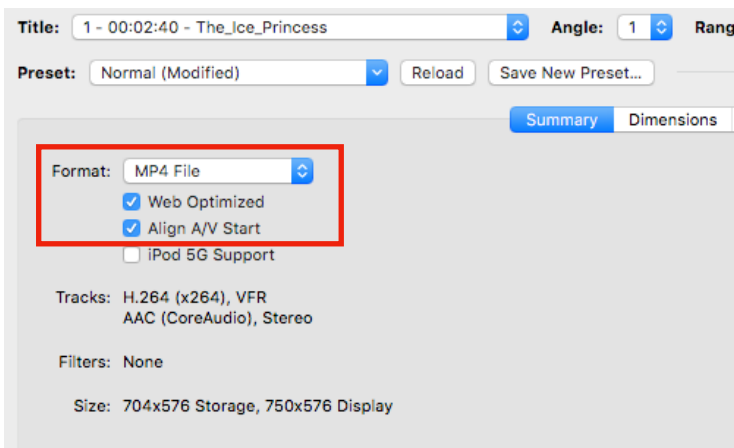
In the top left corner, you'll see the titles of your videos (one at a time; to see the others, click on the title and a menu will appear). Right beneath, you see "Preset:" followed by a drop-down menu. Choose "Legacy", then "Normal" (see image below).



A little bit further down, within the slightly darker grey square, you can see the parameter "Video Encoder". Here is where the important stuff happens: Check if this is set to "H.264 (x264)". If it's not, click on the drop-down menu, and select this encoder (see image below).



The last thing you'll have to check, is in another tab. Click on "Summary" (indicated in the image above with the black arrow). Here, make sure that after "Format" it says "MP4 File", and that below, the boxes "Web Optimized" and "Align A/V Start" are both checked (see image below).



One last thing to indicate is the place where you want the output video(s) to be stored. All the way at the bottom, you see "Save As:". Here, you can choose a title (if you don't want the same as the original). Further to the right you can choose the directory on your computer you want the output to be saved to. Click on "Browse..." and select the folder you want.

When all is in place, you can start the converting. Click on the green button “Start”, or “Add To Queue” if you want to add more videos before starting the conversion. If you have loaded a folder of videos, you can also add them simultaneously to the queue. On a Mac, you can do this by going to “File” and then choosing the option “Add Titles To Queue...”. The advantage of the queue is that you can prepare a couple of videos, store them there, and then start working on others. You can also let the queue run in the background while you prepare the next set of videos. If you want to see how the queue is doing, go to the top-right corner and click on the (black) icon “Queue”. A new window will pop up, showing all the titles you added to the queue and if the converting is done or some videos are still pending.

ATTENTION If you have loaded multiple videos at once, check if the parameter changes you made are applied to all of them. Even though you can load more than one video at a time, you can set the parameters for each of them separately.

With these settings, your video should be compatible with the TLA website. If you have many videos, you can choose to save the parameters you set as a personal preset. On the right side of the “Preset” menu, there is a button saying “Save New Preset...”. Here, you can save your parameters with a name of your choice. Also note that when you look at the queue, it may say “Title 1, Chapter 1” etc. if you added more than video at the same time. Note that this information is not added to your videos. It is mostly practical for yourself: the first group of videos you add will be called ‘chapter 1’, the second ‘chapter 2’ etc; now you can easily see which group of videos has already been finished!

Well, that was about everything you’ll need to know about batch encoding and converting video files. Good luck!

Links to the cited webpages

Drudge’s code deposit

<https://pastebin.com/Xqu2ZA7m>

HandBrake

<https://handbrake.fr/downloads.php>

MKVToolNix

<https://mkvtoolnix.download/>

TLA accepted formats

<https://archive.mpi.nl/accepted-file-formats>

TLA deposit manual

<https://archive.mpi.nl/deposit-manual#Introduction>

UnRar for Windows

<https://www.rarlab.com/download.htm>

Windows Batch files tutorial

<https://www.howtogeek.com/263177/how-to-write-a-batch-script-on-windows/>