

SPIRTON

Convert videos to 60fps (Updated 21 May 2015)

May 19, 2010 / 1,241 Comments

Sections:

- [Introduction](#)
- [Purpose](#)
- [Instructions](#)
- [Final Thoughts](#)
- [FAQ](#)
- [Thanks](#)
- [Changelog](#)

Introduction:

Many modern TVs now include an extra frames feature, usually either 120Hz or 240Hz. This just means they take the source video, which is usually between 24-30FPS, and add new, interpolated frames between the original frames that make the motion look much more smooth.

High framerate (HFR at 48fps) was used in The Hobbit and will be used in upcoming films like the Avatar sequels, so why not watch all your movies at higher framerates?

A lot of people don't realize this, but 99% of TVs made before this feature was introduced are actually capable of displaying 50FPS and 60FPS, and of course most LCD monitors display 60FPS as well.

This tutorial will allow you to convert videos of any framerate to higher framerates; 25FPS becomes 50FPS and 24/30FPS becomes 60FPS by default, or you can specify whichever framerate you want.

This script offers greater quality than any televisions I have seen.

I'm sure many people will want to see how well this script works, so here's a comparison:

[Original File](#)

[Converted File](#)

Note: This can now be done in realtime using [Universal Media Server!](#)

Purpose:

This post provides instructions on how to easily convert any video to 60FPS, which allows you to take advantage of this feature of modern TVs on your computer monitor or TV without actually buying anything.

Step 1:

The files needed for this step are found in [this zip folder](#) (~3MB).

Run and install the file **AviSynth.exe**.

It is an early version of AviSynth 2.6 that is more stable and faster for multithreading than 2.5 releases.

Download and install **MeGUI**, run it, **let it update** all the things it wants to and import all of the presets.

Optional Tip: If you want MeGUI to update to the very latest stuff (at your own risk) you can push ctrl+s, go to the Extra Configuration tab, and in the Auto Update section select "Use development update server"

Close MeGUI and extract the files in the **tools** folder from our zip file into your MeGUI tools folder, which for most people will be either **C:\Program Files (x86)\MeGUI\tools** or **C:\Program Files\MeGUI\tools**.

Step 2:

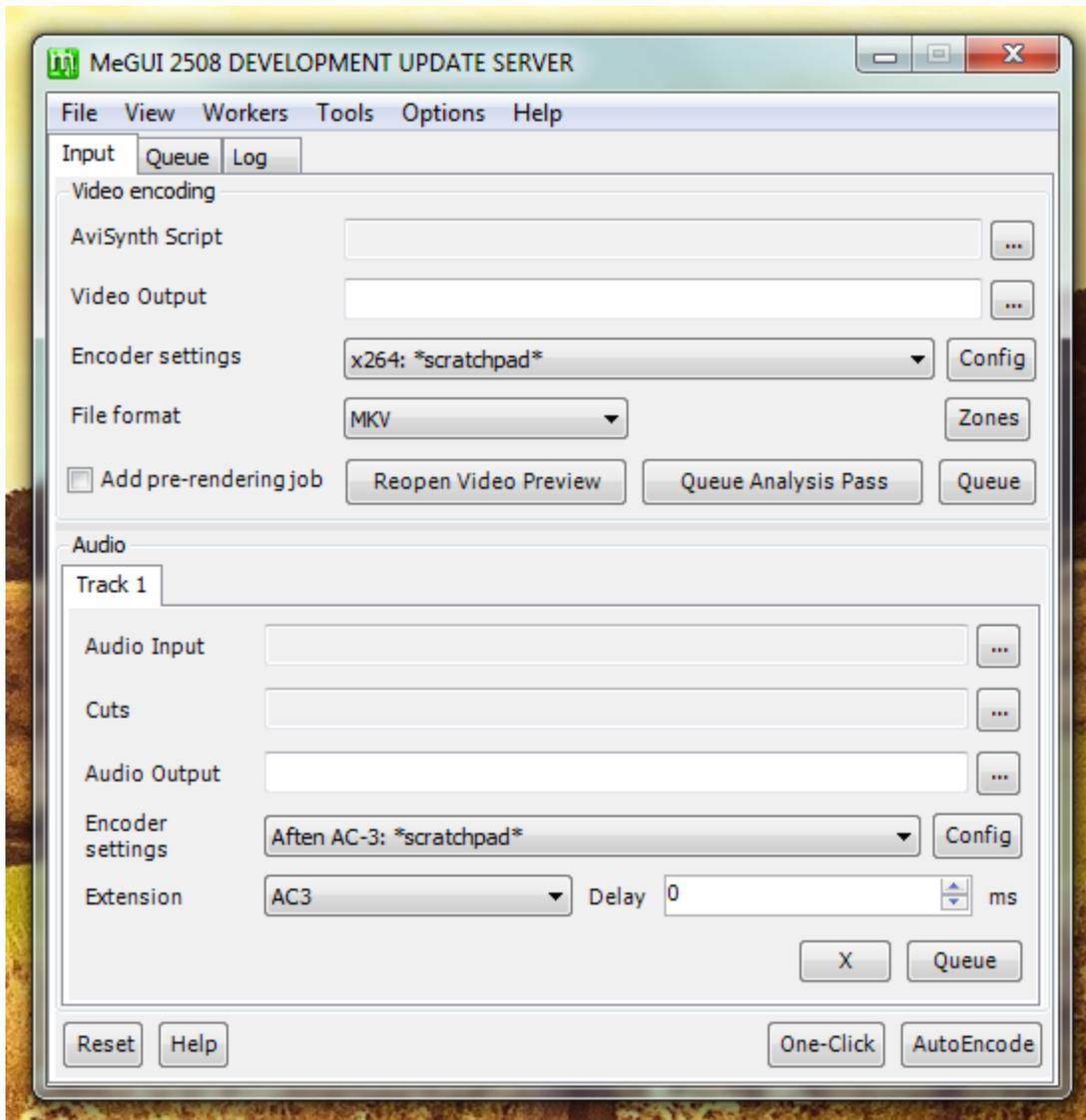
Download and install **MKVToolnix**.

When that is installed, run **MKVMerge** (Start Menu -> All Programs -> MKVToolnix -> mkvmerge GUI).

You will also need codecs installed. If you don't already have them I recommend [K-Lite Mega Codec Pack](#). If you install K-Lite, make sure to select "**Profile 9: Lots of Stuff**" during installation.

Step 3:

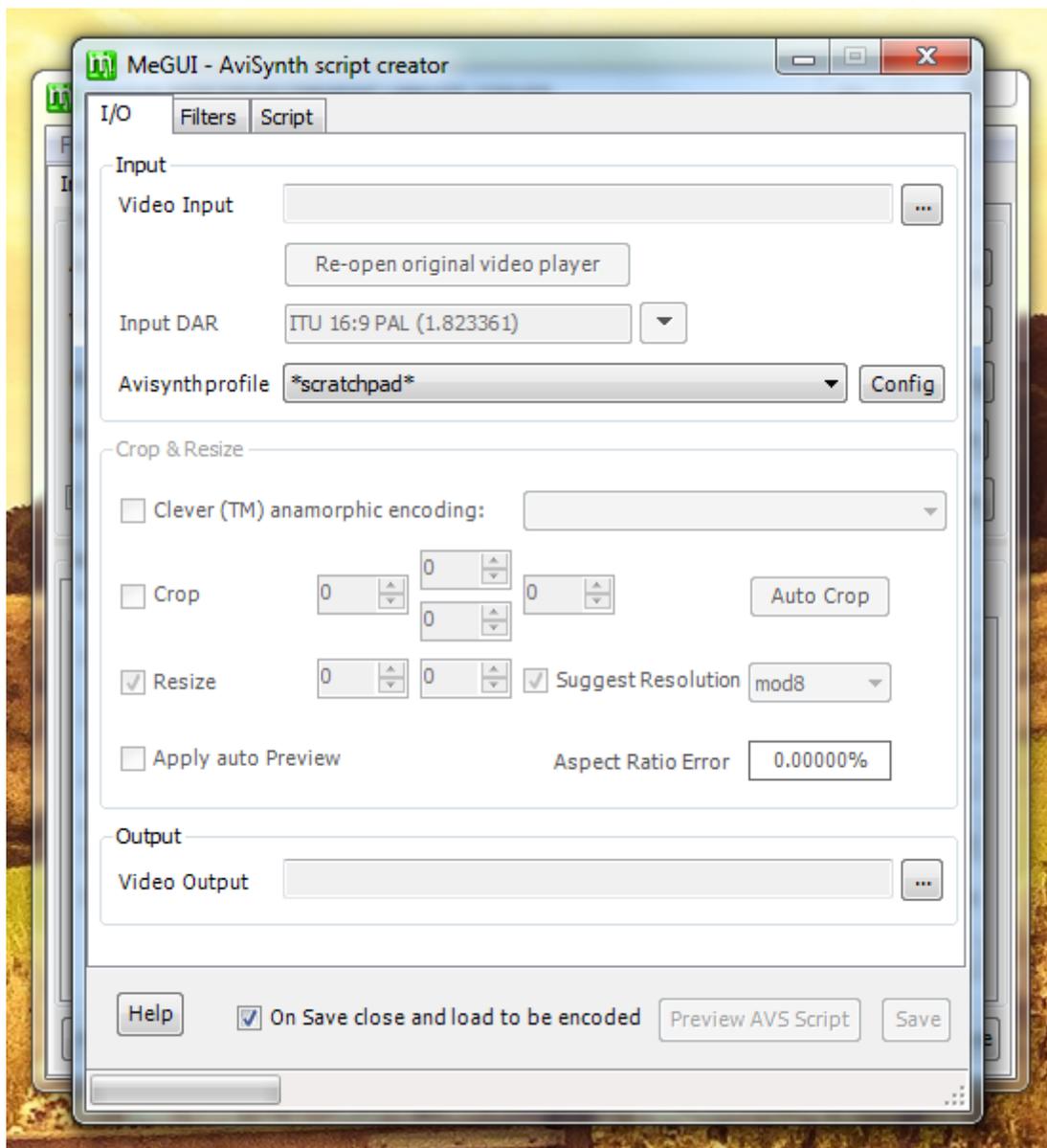
Start MeGUI again. You should be left with a window that looks similar to this:



Step 4:

Push **ctrl+r**, or go to **Tools -> AVS Script Creator**.

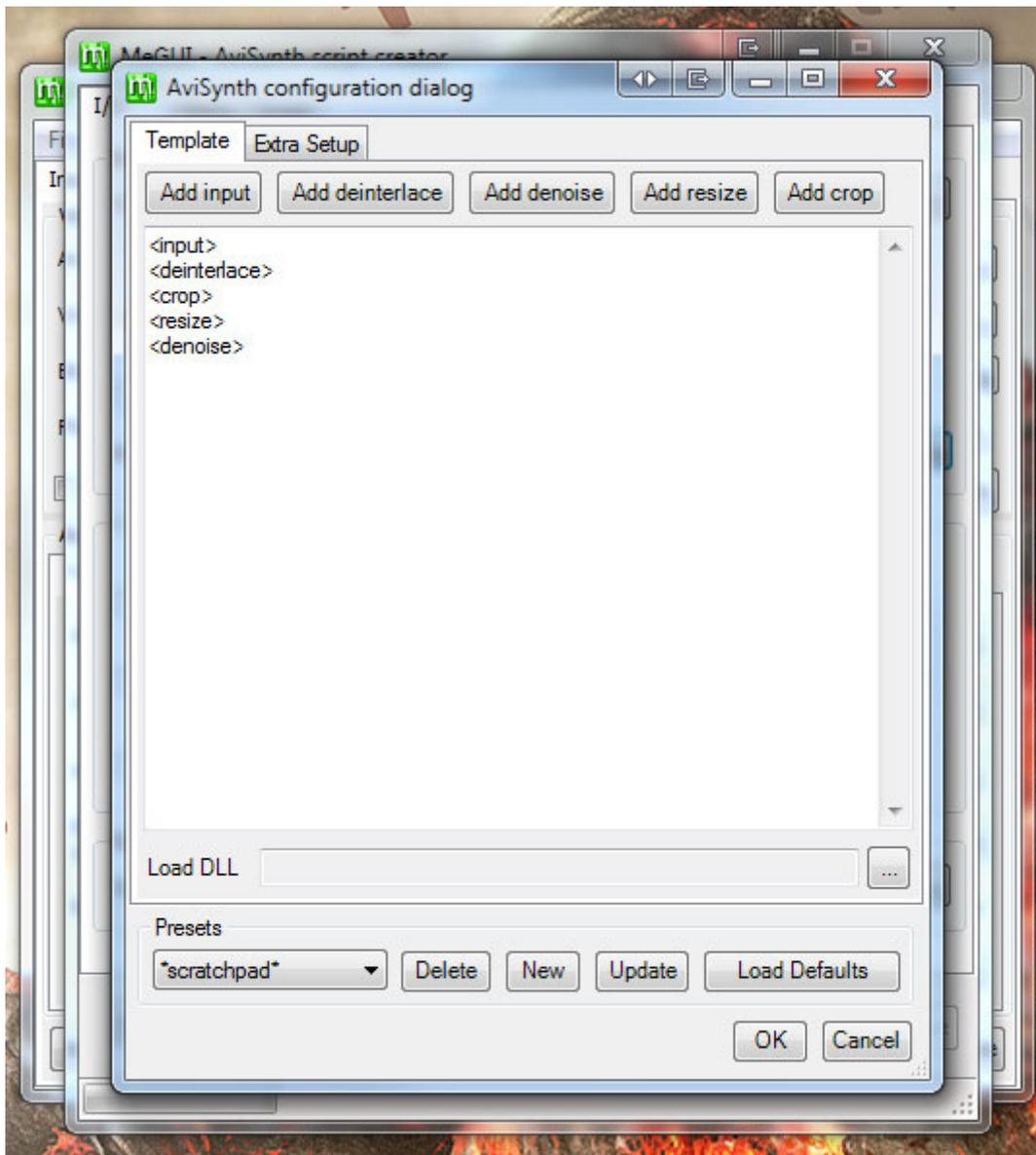
Now you should see something like this:



Step 5:

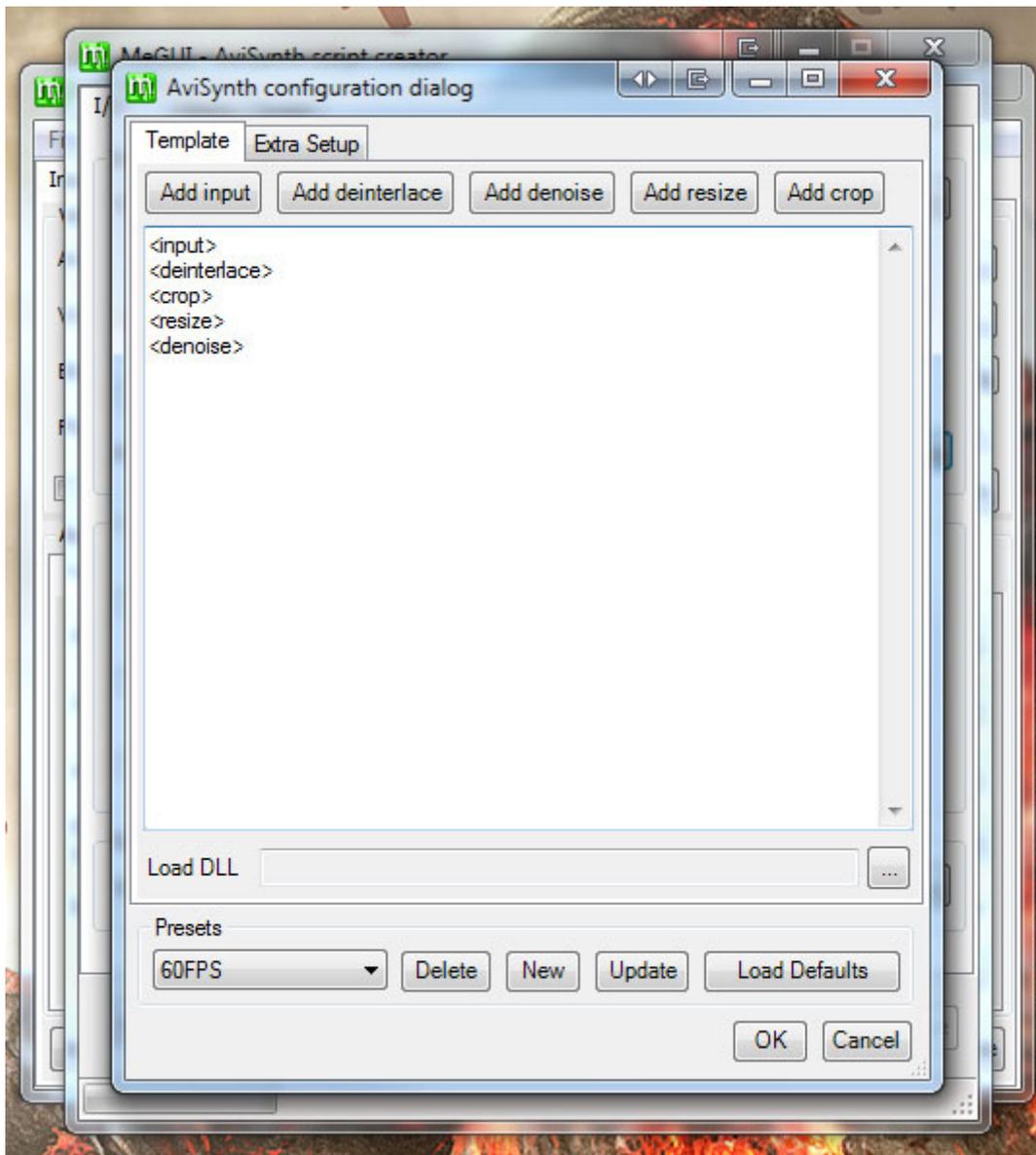
Click the **Config** button next to **Avisynth profile**.

It should look like this:



Step 6:

Click the **New** button down the bottom and type **60FPS** in the name box, so it should look like this:



Step 7:

Now for the script that will do the actual 60FPS conversion.

Just **copy and paste** the following code into the text input area.

```
Cores=4
SetMemoryMax(512)
SetMTMode(3,Cores)
PluginPath = "C:\Program Files (x86)\MeGUI\tools\avisynth_plugin\"
LoadPlugin(PluginPath+"svpflow1.dll")
LoadPlugin(PluginPath+"svpflow2.dll")
Import(PluginPath+"InterFrame2.avsi")
<input>.ConvertToYV12()
SetMTMode(2)
```

```
<deinterlace>  
<crop>  
<denoise>  
<resize>  
InterFrame(Cores=Cores)
```

Important note #1: Make sure the path on the third line is correct.

For example, if you are running 32-bit Windows, you will need to modify the path to “Program Files” instead.

Important note #2: You should change the number “4” in the script (on line 1) to your number of CPU cores. If you don’t know how many cores your CPU has just Google your CPU and you should be able to find the specs somewhere, or 4 should be fine anyway.

Important note #3: If you have a video card, you can improve quality by adding “GPU=true” to the script. However, it may make encoding time slower, so take that into account.

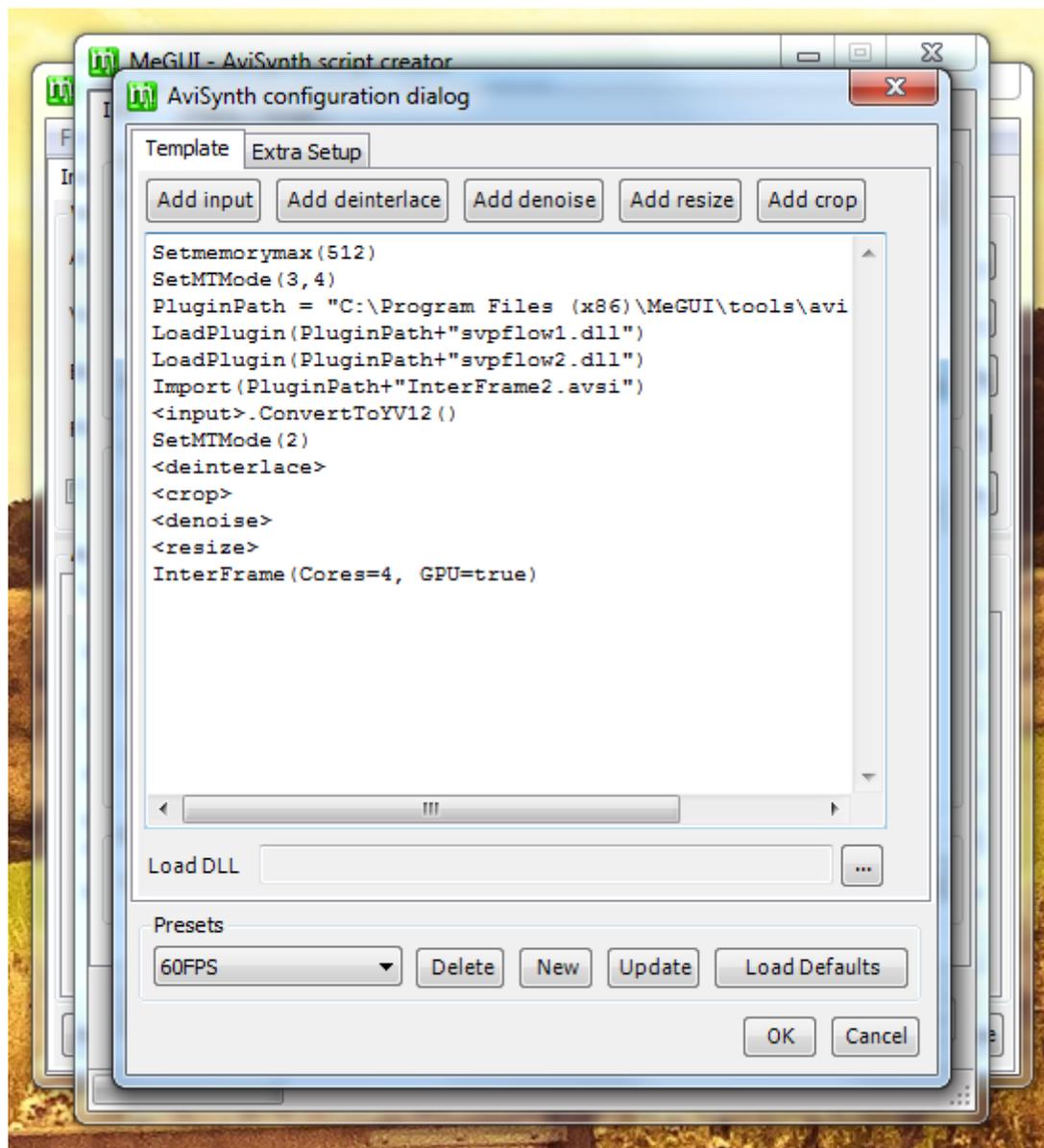
So *InterFrame(Cores=Cores)* would turn into *InterFrame(Cores=Cores, GPU=true)*

If you do this, you may also need to go into your MeGUI tools folder (usually *C:\Program Files (x86)\MeGUI\tools\x264*) and rename the file *openc1.dll* to *openc1.dll.bak*. This will let MeGUI use the version of *openc1.dll* that your video card drivers have installed, instead of the generic one from MeGUI.

Important note #4: This script always outputs either 50FPS or 59.94FPS (60FPS) for compatibility reasons. If you want to always double the framerate instead, so that for example 23.976FPS turns into 47.952FPS (48FPS), you can use the “FrameDouble” parameter.

So *InterFrame(Cores=Cores)* would turn into *InterFrame(Cores=Cores, FrameDouble=true)*

So now your window should look similar to this (with different text):



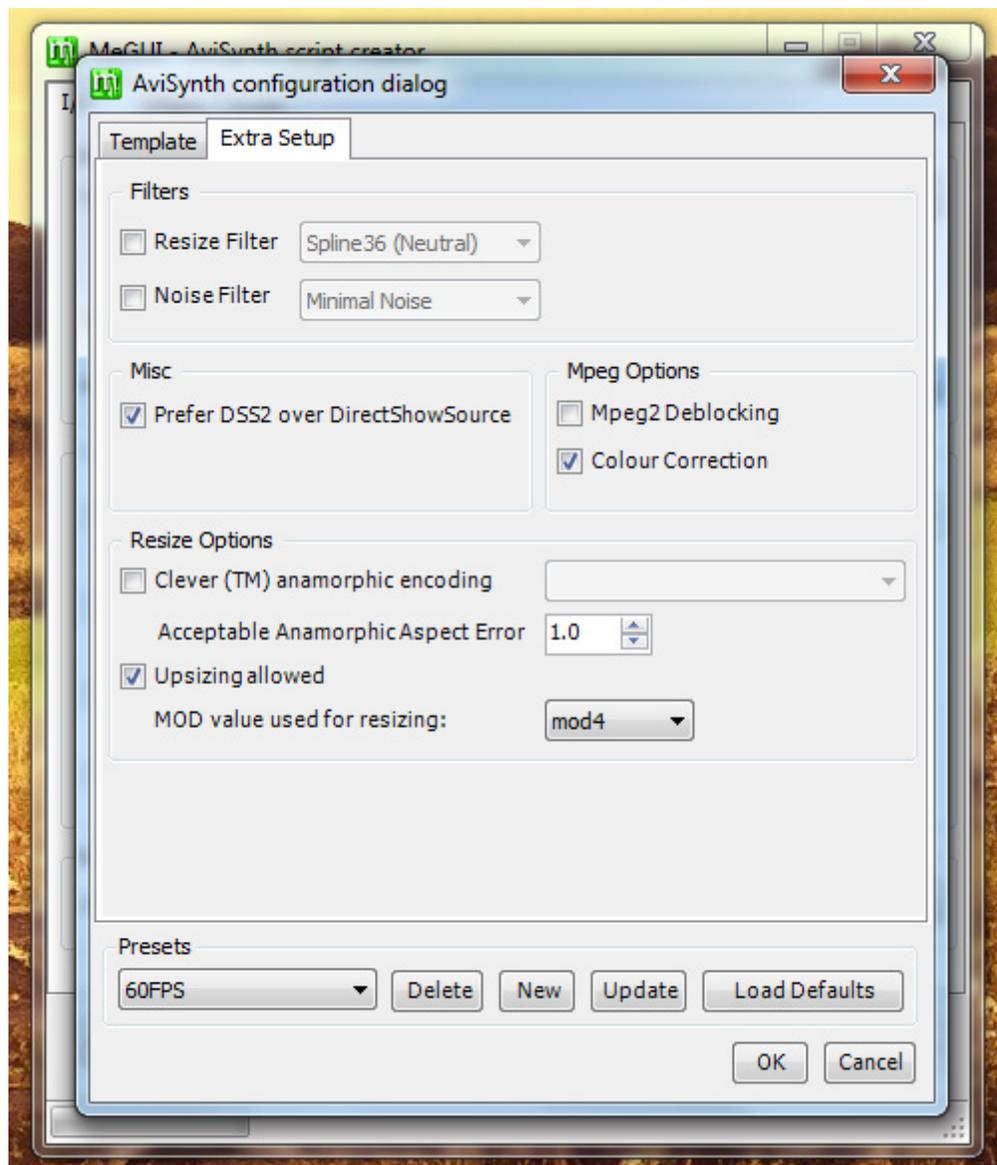
Step 8:

Click the **Extra Setup** tab and make sure “Prefer DSS2 over DirectShowSource” is checked.

Optional tip #1: You might also want to choose your preferred resizer. For aspect-ratio changes (like from 720×576 to 720×480) a neutral filter like Spline36 will be nice. For actual resizing (like from 1920×1080 to 1280×720) a sharp filter like Spline64 will be best. I usually just leave it on Spline36. Make sure to untick the checkbox on the option after you change the default resizer, otherwise it will always be used by default which is often unnecessary.

Optional tip #2: I recommend changing the “MOD value used for resizing” to mod4. This gives the best balance between compatibility (a lot of devices like PS3 don’t have good mod2 support) and aspect ratio accuracy (the higher the MOD value, the more likely the aspect ratio is to be skewed more heavily).

Now it should look something like this:



Then just click the **Update** button then the **OK** button

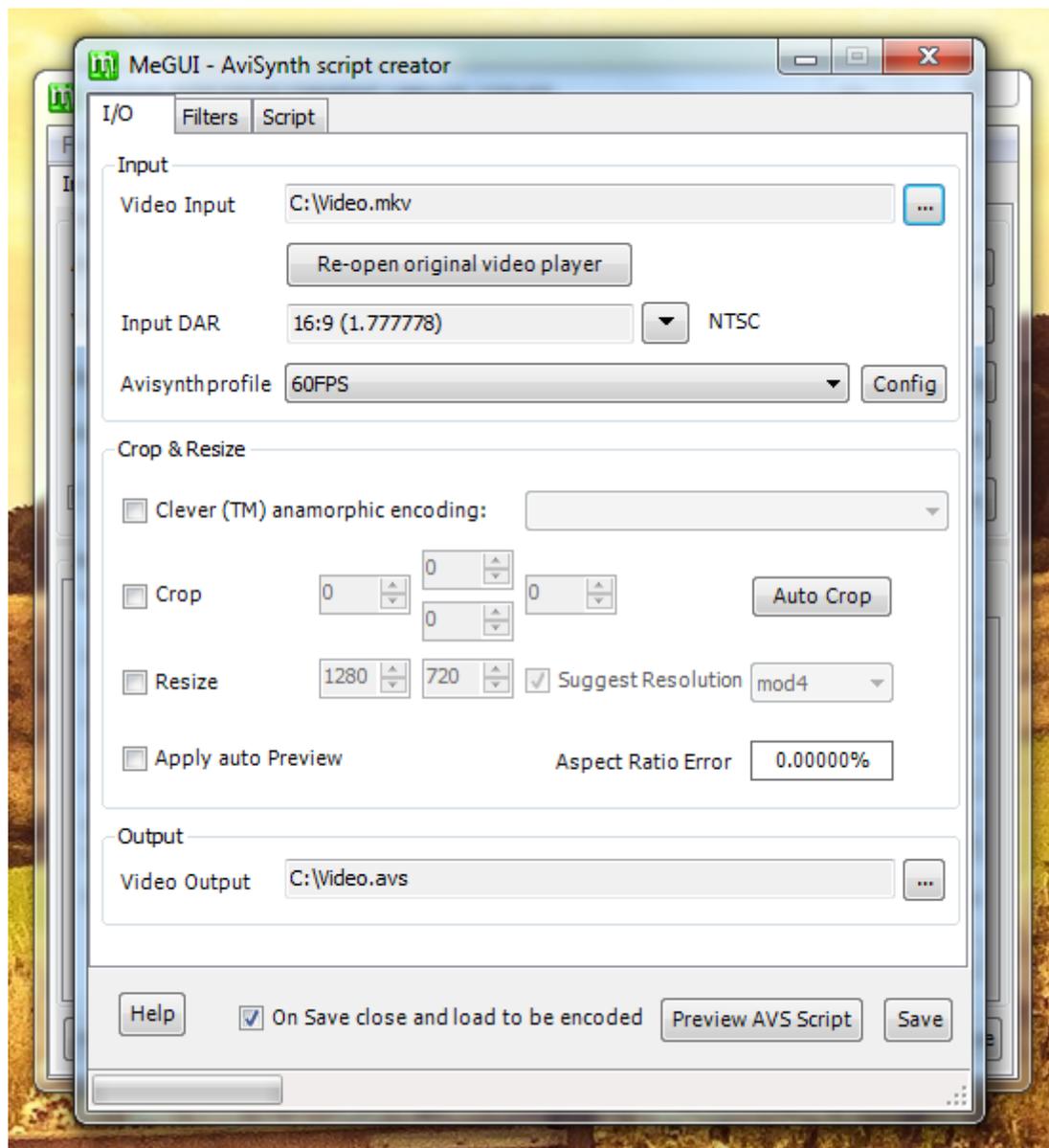
Step 9:

Now we're back to the **AviSynth script creator** window and now that we have our 60FPS AviSynth template made and selected, it's time to choose which video we want. Just click the ... button next to **Video Input** up the top of the window and find your video, then click **Open**.

A window will pop up with buttons, click the option on the right, usually either "DirectShowSource" or "AviSource".

A window will pop up to show you the video, you can just close that.

Now you should see something like this:



Step 10:

Click the **Save** button. The video preview window might pop up again, you can close it again.

Step 11:

Now is the time when you need to choose the quality of the video and audio.

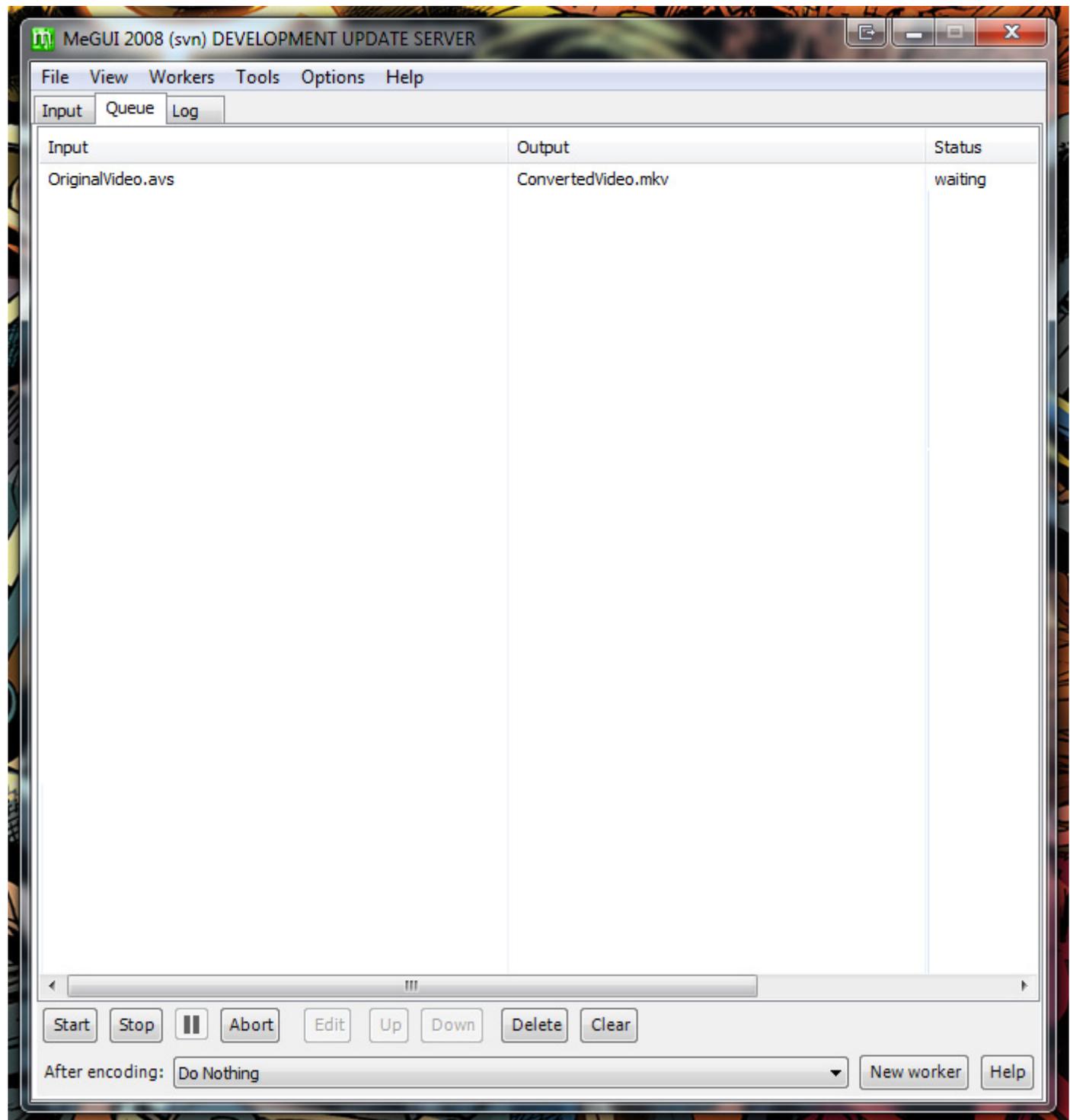
If you don't know what to do here, something quick and easy is to just use CRF, which means you let the codec decide on the bitrates itself. So you can click the "Config" button next to the "Encoder settings" dropdown when you have a x264 option selected, then just enter a number between 16-20 in the "Quality" input box. 16 means your video will be almost exactly the same quality as the source, while 20 will look good too while taking up less space. Personally I use 16 for everything because why not? ☐

So when you have your quality, click **Queue** in the Video encoding section.

Step 12:

Now your video is ready for conversion.

Now just go to the **Queue** tab and click **Start** down the bottom, then wait for it to finish.



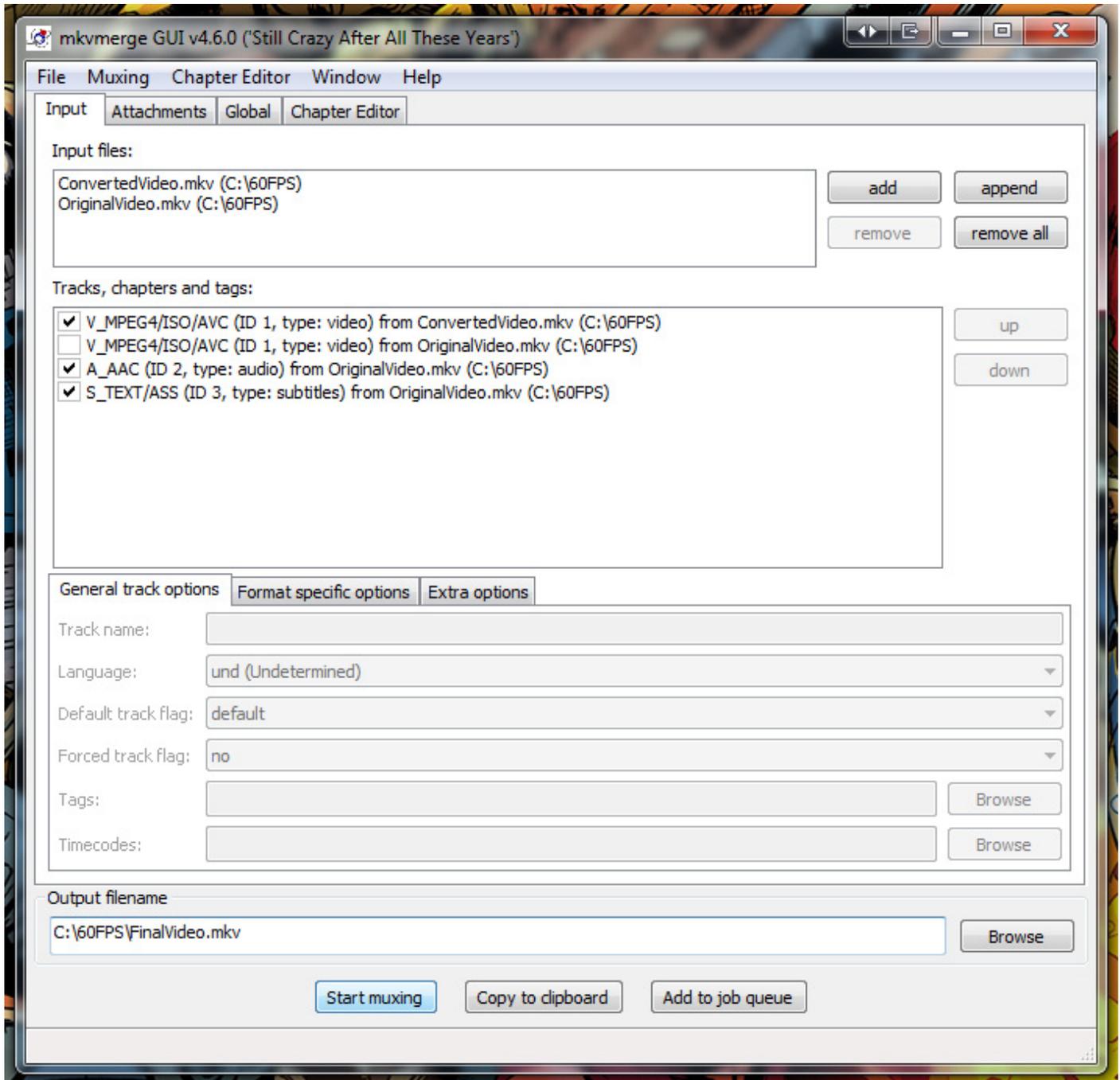
Step 13:

The final step is to add your audio and/or subtitles from the original file to the converted file. This process uses muxing, not conversion, so there is no quality loss.

For this we will use the program we installed earlier, MKVMerge.

So run that and drag the converted file into that program, then do the same with the original file. If you can't drag them on, you can use the **add** button instead.

Now you have several items in the **Tracks, chapters and tags** section. You just need to untick the ones you don't want. Usually you will want to leave them all selected except for the original video track, as I've done in the following example:



Finally, set the **Output filename** down the bottom of the window and click **Start muxing**.
When that is done, you have your final video!

Final Thoughts:

This might seem like a long process, but actually it is only long once. Since we have now made templates, the process for the videos from now on will be much faster; instead of entering the script each time, now you can just make sure your 60FPS AviSynth script is selected.

This file can now be played just like any other video. I personally watch videos on my TV which is linked up to my computer using the awesome, free program called **Universal Media Server**, and that handles these 60FPS videos perfectly with no configuration.

Any comments are appreciated. If you can think of an improvement to the script, or to the process, or if you just found it useful and want to say thanks, please do leave a comment. I would love to know that it's being used.

If you want to donate to help me with server costs and time spent keeping this article updated, you can do so here with my thanks ☐



FAQ:

- **How can I fix random crashing of vfw4x264.exe?**

This probably means you are encoding a file using an input mode other than DSS2, which can happen if the clip is .m2ts or .dga (blu-ray) or another uncommon format.

You can fix it by changing the 2 in the second line of the script to either 3, 4, 5 or just removing that line (try them in that order, since the options get progressively slower)

- **The option “Prefer DSS2 over DirectShowSource” is disabled. or I get an error like “can’t load avss.dll”.**

This means your computer is missing Haali Media Splitter. You can download it [here](#) or as part of a **K-Lite Mega Codec Pack** and the error should go away after you install it. If you install K-Lite, make sure to select “Profile 9: Lots of Stuff” during installation.

- **The file stops encoding after 99%.**

This usually indicates a codec configuration issue. Try installing/reinstalling K-Lite Mega Codec Pack, and make sure to let it select its preferred defaults (not “quick upgrade”)

- **Can I customize the output framerate to make it 48FPS, 75FPS, 120FPS, or anything else?**

Yes, you can choose any framerate you want. It is best to choose a framerate that is divisible in some way with the original video, but even that is your choice. You just need to add the options “NewNum” and “NewDen”, like this:

```
InterFrame(Cores=Cores, NewNum=48000, NewDen=1001)
```

NewNum stands for “New Numerator” and NewDen stands for “New Denominator”. In general, with NTSC material the NewDen should be 1001 and the NewNum should be 1000x your desired framerate (so for 48FPS you want 48000 and 1001) while for PAL content you might just want something like NewNum=75,NewDen=1 to output 75FPS content.

- **The output seems choppy, can I make it smoother?**

Yes, by default we output a video that sacrifices a little smoothness in order to reduce errors, but different people’s eyes pick up on different things so it can all be a matter of taste. You can try the Smooth tuning, like this:

```
InterFrame(Cores=Cores, Tuning="Smooth")
```

Also, using GPU mode produces more accurate results and therefore should appear smoother, though it can be slightly slower overall. You can try it like this:

```
InterFrame(Cores=Cores, GPU=true)
```

Or try both options at once like this:

```
InterFrame(Cores=Cores, Tuning="Smooth", GPU=true)
```

- **Why isn’t the GPU being used much in GPU mode?**

Only some things are done with the GPU in GPU mode, it’s still mostly done by CPU but helped by GPU.

- **Why is the video still the original framerate after running it through MKVMerge?**

You probably selected the wrong tracks in MKVMerge so you’re getting the original video track instead of the new one; make sure you untick the original video stream before muxing.

- **Why does MeGUI give an error “SVSmoothFPS: unable to load library given in ‘compose’ [... svpflow2.dll]”**

Try re-installing or updating your video card drivers, and if that doesn’t work then maybe your video card doesn’t support OpenCL so remove GPU=true from your script.

- **Why am I getting the error “SVSmoothFps: unable to load library given in ‘compose’” when I try to run the encoding process, but not when I preview?**

Go into your MeGUI tools folder (usually `C:\Program Files (x86)\MeGUI\tools\x264`) and rename the file `opencl.dll` to `opencl.dll.bak`. This will let MeGUI use the version of `opencl.dll` that your video card drivers have installed, instead of the generic one from MeGUI.

- **Why am I getting the error “Error message for your reference: Can not read the video stream”?**
MeGUI can't read from folders or files with certain characters, so try moving the video to a different folder and/or renaming the file.

Thanks:

Widezu69 for providing the original AviSynth script that I used as a base.

Fizick for MVTools, the library that does all the work, and **SVP** for the modifications to MVTools that make it even better for our purposes.

Everyone on various forums for their feedback, this is really a community effort.

Changelog:

21/05/2015:

- Updated InterFrame to 2.8.2, which improves speed

16/03/2015:

- Corrected and improved the instructions and FAQ

05/03/2015:

- Updated InterFrame to 2.8.0, which improves speed and lowers memory usage
- Updated conversion script

24/02/2015:

- Updated the instructions and FAQ

27/09/2014:

- Updated InterFrame to 2.6.0
- Added a tip about using framedoubling mode
- Updated most images to reflect the minor changes to MeGUI
- Minor improvements to instructions