FFMPEG

FFMPEG Install & Update

FFMPEG is the best, fastest transcode and swiss-army knife for video and audio files. It’s easy to install and use on a Mac (using homebrew). It *sucks* to install on a PC, but it can run faster - if you have a system that outperforms your Mac. All the cheat sheet things below can be modified for different codecs, and an infinite number of other ways.

Installation

On mac, I recommend installing FFMPEG through Homebrew. Very easy. Google it.

On windoze, I recommend installing FFMPEG through Scoop. You install Scoop via Powershell (non-admin), and then you run FFMPEG via Command Prompt. FFMPEG in a WSL environment, while easier to install in some cases, does not perform nearly as well without significant tweaking of the FFMPEG commands. Vanilla FFMPEG doesn’t work well in WSL(!)

Updating

If you used homebrew to install FFMPEG, it’s this easy:

```
brew update && brew upgrade ffmpeg
```

FFMPEG Cheat Sheet

Directory

You can easily CD your target directory so you don’t have those big old file strings.

Get Frame Count / Duration

(source)

```
ffprobe -v error -select_streams v:0 -count_packets \
    -show_entries stream=nb_read_packets -of csv=p=0 input.mp4
```

FFProbe - Audit Files and Create Log

Replace path with your directory path (both times). This audits for broken quicktime files and skips any that are encoded with NotchLC.

```
find /path -type f -name "*.mov" -exec sh -c 'dir="/path/"; ffprobe -v error -select_streams
v:0 -show_entries stream=codec_name -of default=noprint_wrappers=1:nokey=1 "$1" | grep -q
"NotchLC" && exit 0 || (ffprobe -v error -show_error -count_frames -select_streams v:0 -
read_intervals "%+#1" -skip_frame nokey -i "$1" 2>&1 | grep -v "Referenced QT chapter track
not found" && echo "$1" >> "$dir/incomplete_files.txt")' sh {} \;
```

CRF Low Bitrate MP4

“-crf 24” is bitrate / quality

```
ffmpeg -i input.mp4 -vcodec libx264 -crf 24 output.mp4
```

PNG+Alpha to WebM+ Alpha

```
ffmpeg -framerate 24 -i YourFile-%04d.png -c:v libvpx-vp9 -pix_fmt yuva420p -crf 24
YourOutPutFileNameHere.webm
```

%04d = number of #s in the image sequence file name

CONCAT to HAP

If Start Number is required

```
ffmpeg -framerate 30 -start_number 00126 -i
/Users/can/Desktop/SCRATCH/_somefolder/INTER/SQNCS/Grid-Render-PNG/Grid-Render_%05d.png -c:v
hap /Users/can/Desktop/SCRATCH/_somefolder/INTER/SQNCS/Grid-Render-PNG/Grid-Render_test.mov
```

If Start Number is not required

```
ffmpeg -framerate 30 -i /Users/can/Desktop/SCRATCH/_somefolder/INTER/SQNCS/Grid-Render-
PNG/Grid-Render_%05d.png -c:v hap /Users/can/Desktop/SCRATCH/_somefolder/INTER/SQNCS/Grid-
Render-PNG/Grid-Render_test.mov
```

HAP Alpha

Tiff+alpha to HAP+A

```
ffmpeg -framerate 30 -i /Users/can/Desktop/SCRATCH/_somefolder/INTER/SQNCS/Grid-Overlay/Grid-
Overlay_%05d.tif -c:v hap -format hap_alpha -chunks 6
/Users/can/Desktop/SCRATCH/_somefolder/INTER/SQNCS/Grid-Overlay/Grid-Overlay.mov
```

Transcode 422 to HAP

```
ffmpeg -i /Users/can/Desktop/7x4k-1800frms-APR422.mov -c:v hap -chunks 6
/Users/can/Desktop/7x4k-1800frms-HAP.mov
```

HEVC Straight Transcode

```
ffmpeg -i /Users/can/Desktop/220902_somename_hapq_60fps.mov -c:v hevc
/Users/can/Desktop/220902_somename_hevc.mp4
```

Simple Repackaging + Audio Add

This will take an existing video file and add an audio file to it.

```
ffmpeg -i /Volumes/SSD-01/01-Projects/somename/SOME-EXPORT/V3-HAP10.mov -i
"/Users/can/Desktop/matchv3.aif" -c:a copy -shortest -y -c:v copy /Volumes/SSD-01/01-
Projects/somename/SOME-EXPORT/V4-HAP10-AIF.mov
```

Repackaging 8-Channel Audio

```
ffmpeg -i /Volumes/SSD-01/01-Projects/somename/CRTV/V4-422.mov -i "/Volumes/SSD-01/01-
Projects/somename/CRTV/V5B.aif" -c:a copy -shortest -y -c:v hap -chunks 10 -map 0:v:0 -map
1:a:0 /Volumes/SSD-01/01-Projects/somename/SOME-EXPORT/VID_V4-AUD_V5B-HAP.mov
```

Combining Files AKA Getting Hacky

This example sets an in and out of file A and file B. Then it combines them referencing a simple text file that's just a list of the files. Then it adds audio. All files were already HAP so no transcoding needed to happen.

FILE-A

```
ffmpeg -ss 00:00:00.000 -i /Volumes/CT_FAST-02/somename_creative-HAPV7-chunk10.mov -t
00:08:00.000 -c copy /Users/can/Desktop/7A-StartPatch.mov
```

FILE-B

```
ffmpeg -ss 00:08:00.000 -i /Users/can/Desktop/somename_DELIVERY_V7B.mov -to 00:09:29.464 -c
copy /Users/can/Desktop/7B-EndPatch.mov
```

COMBO

```
ffmpeg -f concat -safe 0 -i /Users/can/Desktop/farts.txt -c copy
/Users/can/Desktop/7ABCOMBO.mov
```

Contents of fart.txt

```
file '/Users/can/Desktop/fileA.mov'
file '/Users/can/Desktop/fileBmov'
```

ADD AUDIO

```
ffmpeg -i /Users/can/Desktop/7ABCOMBO.mov -i "/Volumes/CT_FAST-02/Citizen_NYC_v10_8chan.wav" -
c:a copy -shortest -y -c:v copy -map 0:v:0 -map 1:a:0 /Volumes/CT_FAST-
02/somename_DELIVERY_V7B-HAP-6CHUNK.mov
```

Removing Audio

Remove Audio

```
ffmpeg -i /vidinput.mov -c:v copy -an /videonoaudio.mov
```

REMOVE AUDIO BATCH

cd (the directory)

then

```
for i in `ls *.MOV`; do ffmpeg -i $i -c:v copy -an /Volumes/SSD-01/_AudioRemove/$i; done
```