Encoding video for the web

encoding-video.md

Encoding Video

Installing

Install FFmpeg with homebrew. You'll need to install it with a couple flags for webm and the AAC audio codec.

```
brew install ffmpeg --with-libvpx --with-libvorbis --with-fdk-aac --with-opus
```

If you already have ffmpeg installed, but not with the other libraries, use the `reinstall` command.

```
brew reinstall ffmpeg --with-opus
```

FFmpeg options. The `-c:v` option is an alias for `-vcodec` and `-c:a` is an alias for `-acodec`. `-crf` is Constant Rate Factor.

Constant Rate Factor

> This method allows the encoder to attempt to achieve a certain output quality for the whole file when output file size is of less importance. This provides maximum compression efficiency with a single pass. Each frame gets the bitrate it needs to keep the requested quality level. The downside is that you can't tell it to get a specific filesize or not go over a specific size or bitrate.

Convert to MP4

When converting to an MP4, you want to use the h264 video codec and the aac audio codec because IE11 and earlier only support this combination. The FFmpeg and H.264 Encoding Guide can walk you through some of the H.264 specific options.

```
ffmpeg -i input.mov -vcodec h264 -acodec aac -strict -2 output.mp4
```

For maximum compatibility, use the `profile` option. This may, however, increase the bit rate quite a bit. You can disable the audio stream with the `-an` option. `-pix_fmt yuv420p` is for Apple Quicktime support.

In this example, `input.mov` is converted to `output.mp4` with maximum compatibility, with Quicktime support, and without an audio stream.

```
ffmpeg -an -i input.mov -vcodec libx264 -pix_fmt yuv420p -profile:v baseline -level 3 output.mp4
```

Convert to WebM

VP8

`libvpx` is the VP8 video encoder for WebM. FFmpeg and WebM Encoding Guide will walk you through webm specifics.

In this example, `input.mov` is converted to `output.webm` with a constant rate factor of `10` (lower is higher quality) at a bitrate of `1M`. Changing the bitrate to something lower (e.g. `700K`) will result in lower file sizes and lower quality. If your video does not have audio, you may leave off the `-acodec libvorbis` part.

```
ffmpeg -i input.mov -vcodec libvpx -qmin 0 -qmax 50 -crf 10 -b:v 1M -acodec libvorbis output.webm
```

VP9

VP9 can encode videos at half the file size 😄👋 You can check out Google's VP9 encoding guide for their recommend settings or the FFmpeg VP9 guide.

Here's an example from the FFmpeg guide:

```
ffmpeg -i input.mov -vcodec libvpx-vp9 -b:v 1M -acodec libvorbis output.webm
```

And here's Google's "Best Quality (Slowest) Recommended Settings". You need to run the first line(s). It will create a log file (and warn you the out.webm is empty). On the second pass, the video will be output.

```
ffmpeg -i <source> -c:v libvpx-vp9 -pass 1 -b:v 1000K -threads 1 -speed 4 \
  -tile-columns 0 -frame-parallel 0 -auto-alt-ref 1 -lag-in-frames 25 \
  -g 9999 -aq-mode 0 -an -f webm /dev/null


ffmpeg -i <source> -c:v libvpx-vp9 -pass 2 -b:v 1000K -threads 1 -speed 0 \
  -tile-columns 0 -frame-parallel 0 -auto-alt-ref 1 -lag-in-frames 25 \
  -g 9999 -aq-mode 0 -c:a libopus -b:a 64k -f webm out.webm
```

Support

As of January 2015, all major browsers support MP4.

Data current as of May 2019. Sources:

- jwplayer's research
- caniuse for AV1
- caniuse for MPEG-4/H.264
- caniuse for HEVC/H.265
- caniuse for WebM

Browser	AV1	H264	H265	VP8	VP9	AAC	MP3	VORBIS	OPUS
Chrome for Desktop	70	30	-	30	30	30	30	30	33
Chrome for Android	-	30	-	30	30	30	30	30	-
IE	-	9	10[1]	-	-	9	9	-	-
IE Mobile	-	10	-	-	-	10	10	-	-
Edge	75	12	12[1]	-	14[2]	12	12	-	14
Firefox for Desktop	67	22	-	20	28	22	22	20	20
Firefox for Android	-	20	-	20	28	20	20	20	20
Safari for Mac	-	3	11[3]	-	-	3	3	-	-
Safari for iOS	-	3	11[3]	-	-	3	3	-	-
Opera for Desktop	57	25	-	11	16	-	-	11	12
Android Stock Browser	-	2.3	-	4.0	5	2.3	2.3	4.0	-

Notes

1. Supported only for devices with hardware support.
2. Edge 14+ has partial support for VP9
3. Supported only on macOS High Sierra and onwards.

Recommended markup

Since all browsers support MP4, we can use WebM's VP9 codec for modern browsers and fall back to MP4s for the rest.

```
<video>
  <source src="path/to/video.webm" type="video/webm; codecs=vp9,vorbis">
  <source src="path/to/video.mp4" type="video/mp4">
</video>
```

Creating thumbnail images from the video

Here's their guide. Output a single frame from the video.

```
ffmpeg -i input.mp4 -ss 00:00:14.435 -vframes 1 out.png
```

Output one image every second as a jpg.

```
ffmpeg -i input.mp4 -vf fps=1 out%3d.jpg
```

Reversing a video

FFmpeg now has a reverse filter. Usage: (source from this video.stackexchange answer)

For video only:

```
ffmpeg -i input.mp4 -vf reverse reversed.mp4
```

For audio and video:

```
ffmpeg -i input.mp4 -vf reverse -af areverse reversed.mp4
```

This filter buffers the entire clip. For larger files, segment the file, reverse each segment and then concat the reversed segments.

AKin28 commented on Aug 2, 2016

How to do this with ffmpeg? mp4 to images. 60FPS

szepeviktor commented on Sep 16, 2016

@AKin28 http://stackoverflow.com/questions/10957412/fastest-way-to-extract-frames-using-ffmpeg

kn00tcn commented on Jan 23, 2017

'ie for windows' also includes edge?

jschertz commented on Feb 12, 2017

This is an excellent resource. Thanks for throwing it together!

cenobitedk commented on Mar 11, 2017

This is a great guide, thank you!
However, I'm not impressed with the results compared to a h264 encoding. I'm encoding a 480p file for a website and the h264 encoding results in 20MB. My VP9 file looks awful when the filesize is less than 18MB. All I can read is that VP9 encoding should give me a filesize of down to 50% of the h264 file. Why can't I get near that at all?

Vestride commented on Mar 16, 2017 Author Owner

I updated the support table to include Edge.

TrevorSundberg commented on Apr 5, 2017

The -an command line argument has to go at the end right before the output file, otherwise it does nothing: -an output.mp4

tvler commented on May 8, 2017

Thanks for writing this up :)

BryceLee commented on May 12, 2017

How to convert mp4 to webp?

resool commented on May 25, 2017 • edited by Vestride

nice tutorial!
just fyi: I didn't know what the `-strict` flag meant so I checked the docs and find out that:

```
Note: -strict experimental (or -strict -2) was previously required for this encoder, but it is no longer experimental
and these options are unnecessary since 5 December 2015.
```

source: https://trac.ffmpeg.org/wiki/Encode/AAC#NativeFFmpegAACencoder

ODonnellM commented on Jul 24, 2017

Hey all, I made a little python wrapper to do exactly this in one easy command. I'd love for it to get some use and some feedback if anyone was looking for an open source solution to converting video files for the web.

https://github.com/ODonnellM/webencode

jeffpamer commented on Aug 10, 2017

I've probably revisited this gist 20 times over the past year, so I thought I'd just say thanks for writing it up. It's incredibly useful.

andreyrd commented on Sep 7, 2017

Hey, I believe the `brew install` command should have the flag `--with-fdk-aac`, not `--with-fdk-aacc`. :)

shshaw commented on Sep 15, 2017

FYI: FFMPEG does have a reverse filter now: https://video.stackexchange.com/a/17739

```
ffmpeg -i <source> -vf reverse reversed.mp4
```

shshaw commented on Sep 15, 2017

I keep having issues when converting to webm and ogv, the color of the video drastically changes. Any suggestions?

joeyhoer commented on Sep 29, 2017

Will this post be updated with H265 documentation? https://caniuse.com/#feat=hevc

Vestride commented on Oct 20, 2017 Author Owner

Thanks for the notes, I've updated the install flag, updated the reverse a video section, and added H.265 for IE/Edge/Safari.

jasonm commented on Dec 12, 2017 • edited

For h.264, should this post also advise using `-movflags faststart` per this article about including the `moov` atom at the beginning of the file for efficient seeking?

superhawk610 commented on Jan 16, 2018

@jasonm shouldn't it be `-movflags +faststart`?

AntonTrollback commented on Feb 12, 2018 • edited

Thank you **@Vestride**

Themanwithoutaplan commented on Feb 20, 2018

Has anyone got a snippet for MP4 (AVC, AAC), VP9 and MP4 (HEVC)? I can't seem to come up with a combination that works: Safari will ignore the HEVC if the AVC is there.

Baswazz commented on Mar 20, 2018

Thank you for this great guide **@Vestride**

denji commented on Jul 7, 2018

Removed via `ffmpeg` metadata `-map_metadata -1`.

MatthewZaso commented on Oct 4, 2018

This guide is D O P E

One additional tip for anyone stuck:
I was trying to encode a video with an odd width value and was ending up with the following error:

> width not divisible by 2 (61x58)

After some searching, I found this short command that when appended to the video adds a 1px padding to the video:
(Padding is preferable to scaling in this context to avoid distortion)

`-vf pad="width=ceil(iw/2)*2:height=ceil(ih/2)*2"`

from this StackOverflow post

MatthewZaso commented on Nov 15, 2018 • edited

If you have ffmpeg and the dependencies above installed you can create an executable bash script or alias by doing the below:

```
echo -n "Enter the output filename"; read OUTPUT
echo 'Encoding file '$FILENAME' to mp4 (using profile)...'
ffmpeg -an -i $FILENAME -vcodec libx264 -pix_fmt yuv420p -profile:v baseline -level 3 $OUTPUT'.mp4'
echo 'Encoding file '$FILENAME' to webm (vp9)...'
ffmpeg -i $FILENAME -vcodec libvpx-vp9 -b:v 1M -acodec libvorbis $OUTPUT'.webm'
```

I added this to my `/bin` to cut down on time. Feel free to use and modify to your needs!

mrcoles commented on Feb 6, 2019

With the latest homebrew (`brew update`), it seems that these flags no longer exist for ffmpeg and all the "--with-FOO" options listed here are installed by default. I just tried it out and all the builds appear to work!

> brew install ffmpeg
> ==> Installing dependencies for ffmpeg: aom, frei0r, fribidi, pcre, glib, pixman, cairo, graphite2, harfbuzz, libass, libbluray, libogg, libvorbis, libvpx, opencore-amr, opus, rtmpdump, flac, libsndfile, libsamplerate, rubberband, sdl2, snappy, speex, giflib, webp, leptonica, tesseract, theora, x264 and x265
> …

ai commented on Mar 16, 2019

Firefox now supports AV1. Here is a guide on how to use it on the web:

https://evilmartians.com/chronicles/better-web-video-with-av1-codec

king724 commented on Mar 27, 2019 • edited

> With version 1.7.0, libvpx added support for row based multithreading which greatly enhances the number of threads the encoder can utilise. This improves encoding speed significantly on systems that are otherwise underutilised when encoding VP9. Since the amount of additional threads depends on the number of tiles, which itself depends on the video resolution, encoding higher resolution videos will see a larger performance improvement.
>
> FFmpeg added support for row based multithreading in version 3.4, released on January 25th, 2018. As of libvpx version 1.7.0 this multithreading enhancement is not enabled by default and needs be manually activated with the `-row-mt 1` switch.

source

I'm seeing a large improvement in speed (35 mins vs 21 mins) by adding this (`-row-mt 1`) to the VP9 conversions.

MatthewZaso commented on Apr 12, 2019

If for some reason you find yourself in a situation where you can't use video and *need* to use a gif (I'm so sorry), giphy engineering has a great article about how to create the lowest possible gif sizes with ffmpeg. I just found it really helpful for an HTML email I had to make

https://engineering.giphy.com/how-to-make-gifs-with-ffmpeg/

zajca commented on May 28, 2019

Funny results I have to say. I have 1m 19s animated video. Original MP4 1080p file has 108MiB.

MP4 takes few seconds and resulting file is 14.2MiB
VP8 takes around 1m and file is 10MiB
VP9 takes around 2m and file is 15.7MiB
VP9 (google recomended) takes around hour and file is 13.9MiB

Can't see any benefits of VP9 or it's animated video same special case?

jitendravyas commented on Aug 12, 2019 • edited

My MP4 video (no audio) not working in Microsoft Edge. What setting should I choose?

thetwopct commented on Aug 19, 2019

Note that the method specified to install FFmpeg with options no longer works; example error "Error: invalid option: --with-libvpx"

supun19 commented on Aug 23, 2019

ffmpeg -ss 2.96 -i /opt/online-screen-recorder-server-side-record/lib/../videos/6c1daf61-19da-405c-9e08-cd2c74e0ee86.webm -y -vcodec copy -t 5.63 /opt/online-screen-recorder-server-side-record/lib/../videos/trimVideos/6c1daf61-19da-405c-9e08-cd2c74e0ee86.webm

this command run correctly and give me original file not cut is there any mistake?

amoshydra commented on Sep 11, 2019 • edited

> Funny results I have to say. I have 1m 19s animated video. Original MP4 1080p file has 108MiB.
>
> MP4 takes few seconds and resulting file is 14.2MiB
> VP8 takes around 1m and file is 10MiB
> VP9 takes around 2m and file is 15.7MiB
> VP9 (google recomended) takes around hour and file is 13.9MiB
>
> Can't see any benefits of VP9 or it's animated video same special case?
> -- **@zajca**

I shared the same sentiment when I started pasting these command snippets into my terminal. Then I realise, these command snippets vary in output options, in terms of their encoding speed, output resolution, frame rate, bitrate and quality.

So I have decided to make a few things constant.

- preset medium, unrestricted threads, same speed
- resolution 720p
- frame rate 24
- fixed average bitrate 500 / 1000

And here is my result and my opinion

Codec	Average bitrate	Size	Time taken (s)	My quality rating
x264	1000k	3.705 mb	Video duration (s) / `5`	6.5 / 10
x264	500k	1.857 mb	Video duration (s) / `5`	4 / 10

Codec	Average bitrate	Size	Time taken (s)	My quality rating
vp8	500k	1.909 mb	Video duration (s) / `1.5`	6.5 / 10
vp9	500k	1.661 mb	Video duration (s) / `0.07`	7 / 10

In my context,

- given the same compression settings, vp9 seems to have a better visual quality compared to vp8.
- vp9 / vp8 output seems to output same visual quality with half the file size compared to x264

▶ Expand for commands

svale commented on Nov 8, 2019

Thank you **@Vestride** for a very nice writeup!

MaxChri commented on Feb 18 • edited

Thank you so much for the tutorial. I'm building a streaming plattform and I was wondering if you can convert uploaded videos to mp4 (for browser support) using ffmpeg. It works fine but one big problem is the speed and propably quality (not tested). But if you convert to webm, you'll loose all the quality.