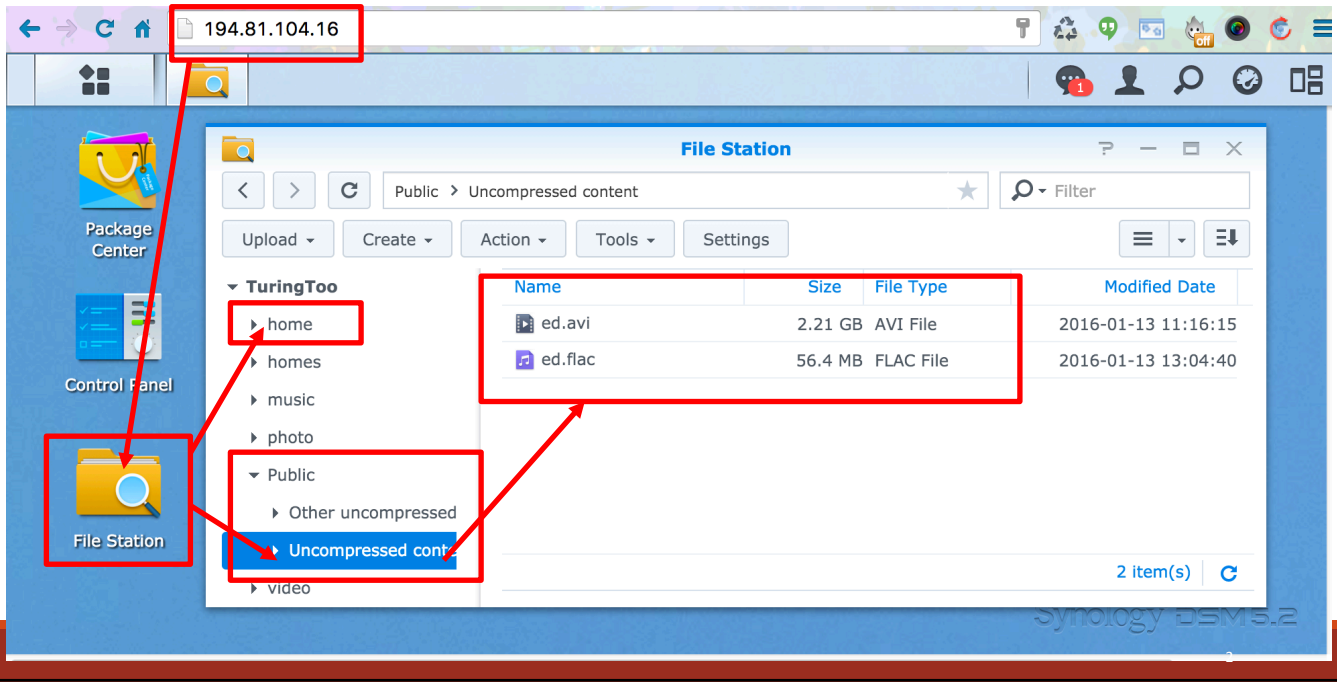
FFmpeg – the swiss army knife of Internet Streaming

references:

- Fabio Sonnati, https://sonnati.wordpress.com/2011/07/11/ffmpeg-the-swiss-army-knife-of-internet-streaming-part-i/

Your new online storage



Download the uncompressed video file

Elephants Dream is a computer-generated short film that was produced almost completely using the free software 3D suite Blender. The name refers to a Dutch tradition used by parents to abruptly end children's bedtime stories with introducing a sneezing elephant.

Create a folder such as “mediatech” in C:\,

Download:
ed.avi (uncompressed video)
ed.flac (uncompressed audio)

3

FFMPEG - part 1

- H.264 can encode video with approximately **3 times fewer bits** than comparable MPEG-2 encoders.
- H.264 is up to **twice** as efficient as MPEG-4 Part 2 (natural video) encoding

Figure.1A MPEG-2

Figure.1B H.264

In figure 1, you can see the difference of encoding via MPEG-2 and H.264 at 100 kpbs.

What is FFMPEG

- FFmpeg is a **free** software project that produces libraries and programs for handling multimedia data.
- FFmpeg includes *libavcodec*, an audio/video codec library used by several other projects, *libavformat*, an audio/video container mux and de-mux library, and the *ffmpeg command line program* for transcoding multimedia files.
- FFMPEG can be compiled under Linux, Mac OS X, Microsoft Windows. Most computing platforms and microprocessor instruction set architectures are also supported, like x86 (IA-32 and x86-64), PPC (PowerPC),ARM, DEC Alpha, SPARC, and MIPS.

5

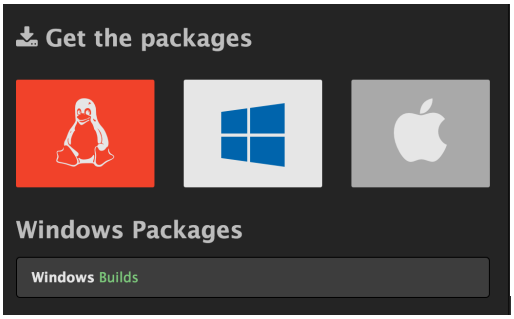
What is FFMPEG

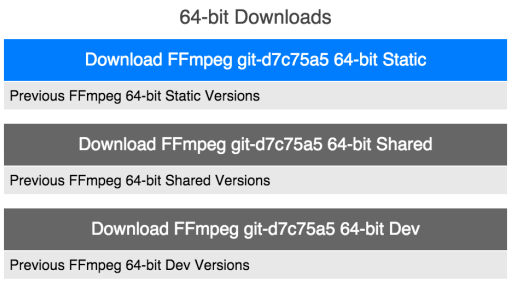
- So ffmpeg is the **command line program** that uses several other **open source programs** (notably x264 for H.264 encoding) to offer exceptional transcoding capabilities especially for **server side batch transcoding**, **live encoding** and **audio/video files manipulation** (muxing, demuxing, slicing, splitting and so on).
- http://www.ffmpeg.org/ (About, download, documentation, news, bug report, wiki, mailing list, forum, etc.)

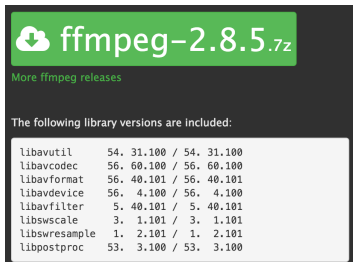
6

Download

- http://www.ffmpeg.org/download.html
- Option 1: Download the source code and compile it.
- **Option 2: Download the pre-compiled packages:**





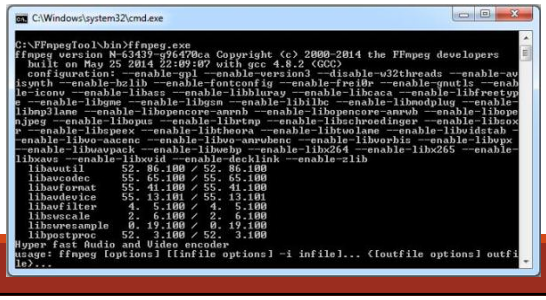


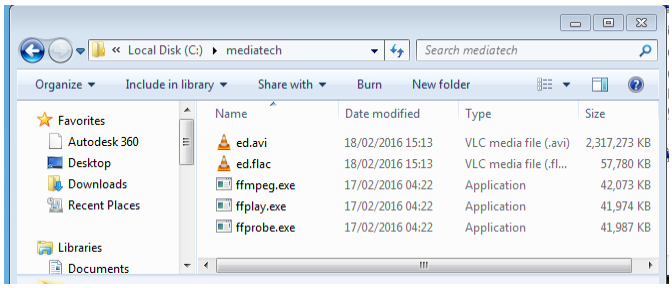
Windows Mac

Download (on windows)

- Extract the downloaded file using 7-zip.

- Copy all files in the bin folder to the mediatech folder.
- Open a command line window, and navigate to C:\mediatech
- Run: ffmpeg





Key parameters

This is the base structure of an FFmpeg invocation:

ffmpeg -i input_file [...parameter list...] output_file

- *Input_file* and *output_file* can be defined not only as file system objects but a number of protocols are supported: **file, http, pipe, rtp/rtsp, raw udp, rtmp**.
- FFmpeg supports hundreds of parameters and options. Very often, FFmpeg infers the parameters from the context
  - for example the input or output format from the file extension and it also applies default values to unspecified parameters.
  - Sometimes it is instead necessary to specify some important parameters to avoid errors or to optimize the encoding.
- Let's start with a selection of generic parameters:

-formats print the list of supported file formats
-codecs print the list of supported codecs (E=encode,D=decode)
-I set the input file. Multiple -i switchs can be used
-f set video format (for the input if before of -i, for output otherwise)
-an ignore audio
-vn ignore video
-ar set audio rate (in Hz)
-ac set the number of channels
-b:a set audio bitrate
-c:a choose audio codec or use "copy" to bypass audio encoding
-c:v choose video codec or use "copy" to bypass video encoding
-r video fps. You can also use fractional values like 30000/1001 instead of 29.97
-s frame size (w x h, ie: 320x240) -aspect set the aspect ratio i.e: 4:3 or 16:9
-sameq ffmpeg tries to keep the visual quality of the input
-t N encode only N seconds of video (you can use also the hh:mm:ss.ddd format)
-croptop, -cropleft, -cropright, -cropbottom crop input video frame on each side
-y automatic overwrite of the output file
-ss select the starting time in the source file
-vol change the volume of the audio
-g Gop size (distance between keyframes)
-b:v Video bitrate
-bt Video bitrate tolerance
-metadata add a key=value metadata

check ffmpeg documentation for any updates of the parameters

Some examples

1. Getting info from a video file

ffmpeg -i ed.avi

(or ffprobe ed.avi)

Useful to retrieve info from a media file like audio/video codecs, fps, frame size and other params.

```
Input #0, avi, from 'ED-1080-png/ed.avi':
  Duration: 00:10:53.79, start: 0.000000, bitrate: 29035 kb/s
    Stream #0:0: Video: mjpeg (MJPG / 0x47504A4D), yuvj420p(pc,
bt470bg/unknown/unknown), 1920x1080 [SAR 1:1 DAR 16:9], 29030 kb/s, 24 fps,
24 tbr, 24 tbn, 24 tbc
```

11

Some examples

2. Encode a video to FLASH video format (ignore audio, reduce video resolution)

ffmpeg -i ed.avi -r 24 -s 320x240 -an video.flv

By default FFmpeg encodes flv files in the old Sorenson's Spark format (H.263). This can be useful today only for compatibility with older systems or if you wanto to encode for a Wii (which supports only Flash Player 7).

```
frame=15691 fps=142 q=2.3 Lsize=   16402kB time=00:10:53.79 bitrate=
205.5kbits/s
video:16156kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB
muxing overhead: 1.518766%
```

then encode audio:

ffmpeg -i ed.flac -c:a aac -ab 196k -ac 2 -ar 48000 audio.mp4

12

Some examples

3. Decode a video into a sequence of frames

mkdir images

ffmpeg -i ed.avi –r 25 images/image%d.jpg

Decodes the video in a sequence of images (25 images per second) with names like image1, image2, image 3, … , imageN.jpg.

ffmpeg –i ed.avi –r 0.1 images/image%3d.jpg

Decodes a picture every 10 seconds (1/0.1=10). Useful to create a thumbnail gallery for your video. In this case the output files have names like image001, image002, etc.

ffmpeg –i ed.avi –r 0.1 –t 20 images/image%3d.jpg

Extracts 2-3 images from the first 20 seconds of the source.

13

Some examples

4. Encode from a sequence of pictures

ffmpeg -f image2 -i images/image%d.jpg –r 25 video.flv

Build a video from a sequence of frame with a name like image1.jpg,image2.jpg,..,imageN.jpg
Is it possible to use different naming conventions like image%3d.jpeg where FFmpeg search for file with names like image 001.jpeg, image002.jpg, etc…The output is an FLV file at 25Fps.

14

Some examples

5. Extract an image from a video

ffmpeg -i ed.avi -frames:v 1 -ss 00:01:00 -f image2 images/image-%3d.jpg

This is a more accurate command for image extraction. It extracts only 1 single frame starting 1min from the start of the video. The thumbnail will have the name image-001.jpg.

ffmpeg -i ed.avi -r 0.5 -frames :v 3 -ss 00:00:05 -f image2 images/image-%3d.jpg

In this case FFmpeg will extract 3 frames, each every 1/0.5=2seconds, starting from time 5s. Useful for video CMS where you want to offer a selection of thumbnails and a backend user choose the best one.

15

Some examples

6. Combine audio + video

ffmpeg -i audio.mp4 -i video.flv -c:v copy -c:a copy output.flv

Depending by the content of the input file you may need to use the map setting to choose and combine the audio video tracks correctly. Here I suppose to have an audio only and video only input files.

```
Duration: 00:10:58.39, start: 0.000000, bitrate: 406 kb/s
   Stream #0:0: Video: flv1, yuv420p, 320x240, 200 kb/s, 24 fps, 24 tbr…
   Stream #0:1: Audio: aac (LC), 48000 Hz, stereo, fltp, 196 kb/s
```

16

Some examples

7. Extract only audio track without re-encoding

ffmpeg -i output.flv -vn -c:a copy audio2.mp4

Similarly you can extract the video track without re-encoding.

8. Extract audio track with re-encoding

ffmpeg -i output.flv -vn -ar 44100 -ac 2 -ab 128k -f mp3 audio3.mp3

This command extract audio and transcode to MP3.

17

Some examples

9. Transcode (using default AV codec configuration)

ffmpeg –i output.flv av.mp4

10. Extract a time slice without re-encoding

ffmpeg -i av.mp4 -ss 00:01:00 -t 00:01:00 -c:a copy -c:v copy slice.mp4

Extract 1min of video starting from time 00:01:00. Be aware that putting the -ss and -t parameters before or after -i may have different effects.

18

H.264 encoding

FFmpeg uses x264 library to encode to H.264. x264 offers a very wide set of parameters and therefore an accurate control over compression.

x264 is a free software library and application for encoding video streams into the H.264/MPEG-4 AVC compression format, and is released under the terms of the GNU GPL.

http://www.videolan.org/developers/x264.html

19

H.264 encoding

ffmpeg -i ed.avi *-r 24 -b:v 5000k -s 1920x1080 -c:v libx264 -flags +loop -me_method hex -g 250 -qcomp 0.6 -qmin 10 -qmax 51 -qdiff 4 -bf 3 -b_strategy 1 -i_qfactor 0.71 -cmp +chroma -subq 8 -me_range 16 -coder 1 -sc_threshold 40 -b-pyramid normal -weightb 1 -mixed-refs 1 -8x8dct 1 -fast-pskip 1 -keyint_min 25 -refs 3 -trellis 1 -level 30 -partitions -parti8x8-parti4x4-partp8x8-partp4x4-partb8x8 -threads 0 -c:a libvo_aacenc -ar 44100 -ab 96k -y* ***h264video1080.mp4***

```
Output #0, mp4, to 'h264video1080.mp4':
  Metadata:
    encoder         : Lavf56.40.101
    Stream #0:0: Video: h264 (libx264) ([33][0][0][0] / 0x0021), yuvj420p(pc),
1920x1080 [SAR 1:1 DAR 16:9], q=10-51, 5000 kb/s, 24 fps, 12288 tbn, 24 tbc
(default)
    Metadata:
      encoder         : Lavc56.60.100 libx264
Stream mapping:
  Stream #0:0 -> #0:0 (mjpeg (native) -> h264 (libx264))
Press [q] to stop, [?] for help
frame=  732 fps= 27 q=26.0 size=   17633kB time=00:00:28.08 bitrate=5143.7kbits/s
```

20

H.264 encoding

ffmpeg -i ed.avi *-r 24 -b:v 1000k -s 640x360 -c:v libx264 -flags +loop -me_method hex -g 250 -qcomp 0.6 -qmin 10 -qmax 51 -qdiff 4 -bf 3 -b_strategy 1 -i_qfactor 0.71 -cmp +chroma -subq 8 -me_range 16 -coder 1 -sc_threshold 40 -b-pyramid normal -weightb 1 -mixed-refs 1 -8x8dct 1 -fast-pskip 1 -keyint_min 25 -refs 3 -trellis 1 -level 30 -partitions -parti8x8-parti4x4-partp8x8-partp4x4-partb8x8 -threads 0 -c:a libvo_aacenc -ar 44100 -ab 96k -y* ***h264video360.mp4***

```
Output #0, mp4, to 'h264video.mp4':
  Metadata:
    encoder         : Lavf56.40.101
    Stream #0:0: Video: h264 (libx264) ([33][0][0][0] / 0x0021), yuvj420p(pc),
640x360 [SAR 1:1 DAR 16:9], q=10-51, 1000 kb/s, 24 fps, 12288 tbn, 24 tbc
(default)
    Metadata:
      encoder         : Lavc56.60.100 libx264
Stream mapping:
  Stream #0:0 -> #0:0 (mjpeg (native) -> h264 (libx264))
Press [q] to stop, [?] for help
frame= 4407 fps= 96 q=27.0 size=   21985kB time=00:03:01.25 bitrate= 993.7kbits/s
```

22

H.264 encoding (how/what to configure?)

ffmpeg -i ed.avi -r 24 -b:v 1000k -s 640x360 -c:v libx264 -flags +loop -me_method hex -g 250 -qcomp 0.6 -qmin 10 -qmax 51 -qdiff 4 -bf 3 -b_strategy 1 -i_qfactor 0.71 -cmp +chroma -subq 8 -me_range 16 -coder 1 -sc_threshold 40 -b-pyramid normal -weightb 1 -mixed-refs 1 -8x8dct 1 -fast-pskip 1 -keyint_min 25 -refs 3 -trellis 1 -level 30 -partitions -parti8x8-parti4x4-partp8x8-partp4x4-partb8x8 -threads 0 -c:a libvo_aacenc -ar 44100 -ab 96k -y ***h264video.mp4***

https://www.ffmpeg.org/ffmpeg-codecs.html#toc-libx264_002c-libx264rgb

a framerate of 24 Fps (**-r**),
a target bitrate of 1000Kbit/s (**-b:v**),
a gop max-size of 250 frames (**-g**),
3 b-frames (**-bf**)
resizing the input to 640×360 (**-s**).
The level is set to 3.0 (**-level**)
the entropy coder to CABAC (**-coder 1**)
the number of reference frames to 3 (**-refs**).
The profile is determined by the presence of b-frames, dct8x8 and Cabac, so it is an *high-profile*.

23

H.264 encoding (how/what to configure?)

-me_method Sets the accuracy of the search method in motion estimation.
-subq Sets the accuracy of motion vectors. Accepts values in the range 1-10.
-g, -keyint_min, -sc_threshold x264 uses by default a dynamic gop size. -g selects the max gop size, -keyint_min the min size.–sc_threshold is the Scene Change sensitivity (0-100). At every scene change a new i-frame (intra compressed frame) is inserted. Depending by -g and –keyint_min an I-frame (IDR frame alias keyframe) is inserted instead. The gop can be long (i.e. -g 300) for compression efficiency sake, or short (i.e. 25/50) for accessibility sake.
-bf, b-strategy -bf sets the max number of consecutive b-frames (H.264 supports up to 16 b-frames). Remember that b-frames are not allowed in baseline profile. B-strategy defines the technique used for b-frames placement.
-refs sets the number of reference frames (H.264 supports up to 16 reference frames). Influences the encoding time. Using more than 4-5 refs gives commonly very little or null gain.
-partitions H.264 supports several partitions modes for MBs estimation and compensation. P-macroblocks can be subdivided into 16×8, 8×16, 8×8, 4×8, 8×4, and 4×4 partitions. B-macroblocks can be divided into 16×8, 8×16, and 8×8 partitions. I-macroblocks can be divided into 4×4 or 8×8 partitions.
-b, -pass, -crf, -maxrate, -bufsize -b sets the desired bitrate that will be achieved using a single pass or multi-pass process using the -pass parameter. -crf define a desired average quality instead of a target bitrate.

24

H.264 encoding - Rate control

- x264 supports different rate control techniques: Average Bit Rate (**ABR**), Constant Bit Rate (**CBR**), Variable Bit Rate (**VBR** at constant quality or constant quantization).
- FFmpeg supports **multi pass encoding**. The most common is the 2 pass encoding. In the first pass the encoder collects information about the video's complexity and create a stat file. In the 2nd pass the stat file is used for final encoding and better bit allocation. This is the generic syntax:
- **ffmpeg -i input -pass 1 [parameters] output.mp4**
  ffmpeg -i input -pass 2 [parameters] output.mp4

25

H.264 encoding - Rate control

- **ABR -** Average Bitrate is the default rate control mode. Simply set the desired target average bitrate. Remember that the bitrate can fluctuate freely locally and only the average value over the whole video duration is controlled. Use a 2-pass for better data allocation.
- **CBR -** Using the VBV model (Video Bitrate Verifier) it's possible to obtain CBR encoding with custom buffer control. Single pass CBR is sufficiently efficient. For example, to encode in CBR mode use:
  ffmpeg -i input -b:v 1000k -maxrate 1000k -bufsize 1000k [parameters]…
- **VBR** libx264 supports two unconstrained VBR modes (constant quality or quantisation
  -cqp sets a costant quantization for each frame. It is rarely useful.
  -crf (Constant Rate Factor) sets a target quality factor and lets the encoder to change the quantization depending by frame type and sequence complexity. The -crf factor can usually be chosen in the range 18 to 30-35 (low quality). Usually VBR encoding is performed in single pass.

26

List of libx264 options

ffmpeg -h full > options.txt

scroll down to
"libx264 AVOptions"

```
libx264 AVOptions:
  -preset           <string>     E..V.... Set the encoding preset (cf. x264 --fullhelp) (default "medium")
  -tune             <string>     E..V.... Tune the encoding params (cf. x264 --fullhelp)
  -profile          <string>     E..V.... Set profile restrictions (cf. x264 --fullhelp)
  -fastfirstpass    <int>        E..V.... Use fast settings when encoding first pass (from 0 to 1) (default 1)
  -level            <string>     E..V.... Specify level (as defined by Annex A)
  -passlogfile      <string>     E..V.... Filename for 2 pass stats
  -wpredp           <string>     E..V.... Weighted prediction for P-frames
  -x264opts         <string>     E..V.... x264 options
  -crf              <float>      E..V.... Select the quality for constant quality mode (from -1 to FLT_MAX) (default -1)
  -crf_max          <float>      E..V.... In CRF mode, prevents VBV from lowering quality beyond this point. (from -1 to FLT_MAX) (default -1)
  -qp               <int>        E..V.... Constant quantization parameter rate control method (from -1 to INT_MAX) (default -1)
  -aq-mode          <int>        E..V.... AQ method (from -1 to INT_MAX) (default -1)
     none                        E..V....
     variance                    E..V.... Variance AQ (complexity mask)
     autovariance                E..V.... Auto-variance AQ
     autovariance-biased           E..V.... Auto-variance AQ with bias to dark scenes
  -aq-strength      <float>      E..V.... AQ strength. Reduces blocking and blurring in flat and textured areas. (from -1 to FLT_MAX) (default -1)
  -psy              <int>        E..V.... Use psychovisual optimizations. (from -1 to 1) (default -1)
  -psy-rd           <string>     E..V.... Strength of psychovisual optimization, in <psy-rd>:<psy-trellis> format.
  -rc-lookahead     <int>        E..V.... Number of frames to look ahead for frametype and ratecontrol (from -1 to INT_MAX) (default -1)
  -weightb          <int>        E..V.... Weighted prediction for B-frames. (from -1 to 1) (default -1)
  -weightp          <int>        E..V.... Weighted prediction analysis method. (from -1 to INT_MAX) (default -1)
     none                        E..V....
     simple                      E..V....
     smart                       E..V....
  -ssim             <int>        E..V.... Calculate and print SSIM stats. (from -1 to 1) (default -1)
  -intra-refresh    <int>        E..V.... Use Periodic Intra Refresh instead of IDR frames. (from -1 to 1) (default -1)
  -bluray-compat    <int>        E..V.... Bluray compatibility workarounds. (from -1 to 1) (default -1)
  -b-bias           <int>        E..V.... Influences how often B-frames are used (from INT_MIN to INT_MAX) (default INT_MIN)
  -b-pyramid        <int>        E..V.... Keep some B-frames as references. (from -1 to INT_MAX) (default -1)
     none                        E..V....
     strict                      E..V.... Strictly hierarchical pyramid
     normal                      E..V.... Non-strict (not Blu-ray compatible)
  -mixed-refs       <int>        E..V.... One reference per partition, as opposed to one reference per macroblock (from -1 to 1) (default -1)
  -8x8dct           <int>        E..V.... High profile 8x8 transform. (from -1 to 1) (default -1)
  -fast-pskip       <int>        E..V.... (from -1 to 1) (default -1)
  -aud              <int>        E..V.... Use access unit delimiters. (from -1 to 1) (default -1)
  -mbtree           <int>        E..V.... Use macroblock tree ratecontrol. (from -1 to 1) (default -1)
  -deblock          <string>     E..V.... Loop filter parameters, in <alpha:beta> form.
  -cplxblur         <float>      E..V.... Reduce fluctuations in QP (before curve compression) (from -1 to FLT_MAX) (default -1)
  -partitions       <string>     E..V.... A comma-separated list of partitions to consider. Possible values: p8x8, p4x4, b8x8, i8x8, i4x4, none, all
  -direct-pred      <int>        E..V.... Direct MV prediction mode (from -1 to INT_MAX) (default -1)
     none                        E..V....
     spatial                     E..V....
     temporal                    E..V....
     auto                        E..V....
  -slice-max-size   <int>        E..V.... Limit the size of each slice in bytes (from -1 to INT_MAX) (default -1)
  -stats            <string>     E..V.... Filename for 2 pass stats
  -nal-hrd          <int>        E..V.... Signal HRD information (requires vbv-bufsize; cbr not allowed in .mp4) (from -1 to INT_MAX) (default -1)
     none                        E..V....
     vbr                         E..V....
     cbr                         E..V....
  -avcintra-class   <int>        E..V.... AVC-Intra class 50/100/200 (from -1 to 200) (default -1)
  -motion-est       <int>        E..V.... Set motion estimation method (from -1 to 4) (default -1)
     dia                         E..V....
     hex                         E..V....
     umh                         E..V....
     esa                         E..V....
     tesa                        E..V....
  -forced-idr       <int>        E..V.... If forcing keyframes, force them as IDR frames. (from -1 to 1) (default -1)
  -x264-params      <string>     E..V.... Override the x264 configuration using a :-separated list of key=value parameters
```

Now you can run online streaming service

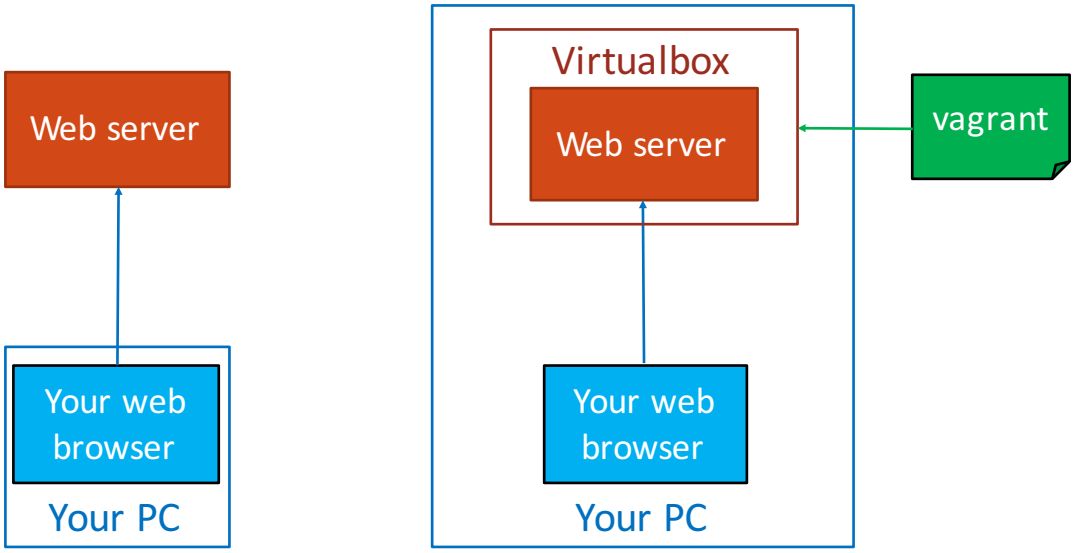
- Combine your H.264 encoded video and AAC encoded audio:

ffmpeg -i h264video1080.mp4 -i audio.mp4 -c:v copy -c:a copy myfilm.mp4

- H.264/AAC in MP4 is one of the standard format for HTML5 streaming. This means that you can put your MP4 on (nearly) any web server and watch them from (nearly) any web browser.
- To test it, we need a server first.

28

Test environment



29

Preparing your WWW streaming server

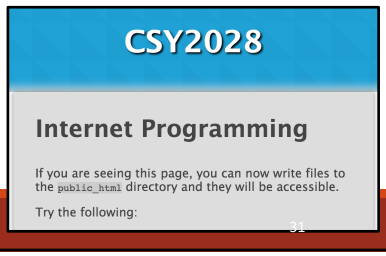
Special thanks to CSY2028!

1. If you are on your laptop or home PC install Virtualbox and Vagrant, if you're using a University Computer you can skip this step.
   https://www.virtualbox.org/
   https://www.vagrantup.com/
2. Download the CSY2028 Vagrantfile:
   http://www.eng.nene.ac.uk/~tom/csy2028/Vagrantfile
3. Copy the Vagrantfile to the directory where you want to store your code. On the university network this will probably be X:\CSY3010.
   The Vagrantfile must be stored in the directory where you want to store your code for this module.

30

Preparing your WWW streaming server

4. Open command prompt and navigate to that directory e.g type:
   > X:
   > cd CSY3010
   to navigate into X:\CSY3010 Now type
   > vagrant up
   if you have followed the previous steps correctly this will launch the server and take a few minutes.
5. Once the server is booted open your browser and navigate to http://192.168.56.1:8080/ or http://192.168.56.2/ and you should see the demo page.



31

Adding content and web pages

6. Navigate to "public_html" folder
   > cd public_html
7. Create a file and name it "myvideo.html"
8. Using a text editor to open "myvideo.html" and paste the following HTML code in. Save the file.

```
<!DOCTYPE html>
<html><body>
<h1>Enjoy! </h1>
<video width="1280" height="720" controls autoplay>
<source src="myfilm.mp4" type="video/mp4">
Your browser does not support the video tag.
</video>
</body></html>
```

9. Copy your myfilm.mp4 file in the same folder
10. On your browser, go to http://192.168.56.1:8080/myvideo.html

32

Adding content and web pages

Adding content and web pages

MPEG-4/H.264 video format - OTHER

Global 80.25% + 10.49% = 90.75%

Commonly used video compression format.

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			45					4.3	
9			46					4.4	
10	12	42	47			8.4		4.4.4	
11	13	43	48	9	34	9.2	8	46	47
	14	44	49	9.1	35	9.3			
		45	50		36				
		46	51						

What else can I do with HTML5 video (like a pro)?

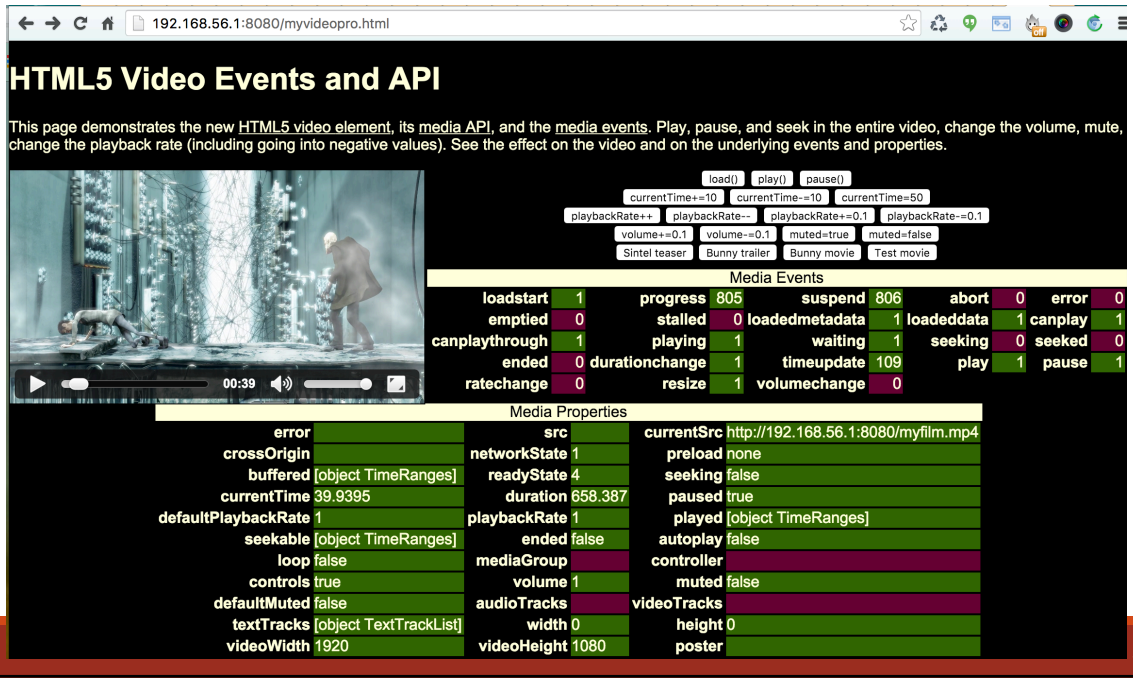
Create a new file: myvideopro.html

Open it with a texteditor and copy the following code in (change the file name if its different from yours).

```
<!DOCTYPE html><html lang="en"><head> <meta charset='utf-8'> <title>HTML5 Video</title> <script src="https://www.w3.org/2010/05/video/script.js"> </script> <link href="https://www.w3.org/2010/05/video/style.css" rel="stylesheet" type="text/css"></head><body> <h1>HTML5 Video Events and API</h1> <p>This page demonstrates the new <a href='http://www.w3.org/TR/html5/embedded-content-0.html#the-video-element'>HTML5 video element</a>, its <a href='http://www.w3.org/TR/html5/embedded-content-0.html#htmlmediaelement'>media API</a>, and the <a href='http://www.w3.org/TR/html5/embedded-content-0.html#mediaevents'>media events</a>. Play, pause, and seek in the entire video, change the volume, mute, change the playback rate (including going into negative values). See the effect on the video and on the underlying events and properties.</p>

<div> <video id='video' controls preload='none' mediagroup='myVideoGroup'> <source id='mp4' src="myfilm.mp4" type='video/mp4'> <p>Your user agent does not support the HTML5 Video element.</p> </video> <div id='buttons'> <button onclick="getController().load()">load()</button> <button onclick="getController().play()">play()</button> <button onclick="getController().pause()">pause()</button><br> <button onclick="getController().currentTime+=10">currentTime+=10</button> <button onclick="getController().currentTime-=10">currentTime-=10</button> <button onclick="getController().currentTime=50">currentTime=50</button><br> <button onclick="getController().playbackRate++">playbackRate++</button> <button onclick="getController().playbackRate--">playbackRate--</button> <button onclick="getController().playbackRate+=0.1">playbackRate+=0.1</button> <button onclick="getController().playbackRate-=0.1">playbackRate-=0.1</button><br> <button onclick="getController().volume+=0.1">volume+=0.1</button> <button onclick="getController().volume-=0.1">volume-=0.1</button> <button onclick="getController().muted=true">muted=true</button> <button onclick="getController().muted=false">muted=false</button><br> <button onclick="switchVideo(0);">Sintel teaser</button> <button onclick="switchVideo(1);">Bunny trailer</button> <button onclick="switchVideo(2);">Bunny movie</button> <button onclick="switchVideo(3);">Test movie</button> </div> <table> <caption>Media Events</caption> <tbody id='events'> </tbody> </table> <table> <caption>Media Properties</caption> <tbody id='properties'> </tbody> </table> <table> <caption>MediaController Events</caption> <tbody id='mc_events'> </tbody> </table> <table> <caption>MediaController Properties</caption> <tbody id='mc_properties'> </tbody> </table> <table id='canPlayType'> <caption>canPlayType</caption> <tbody id='m_video'> </tbody> </table> <table id='tracks'> <caption>Tracks</caption> <tbody> <tr> <th>Audio</th> <th>Video</th> <th>Text</th> </tr> <tr> <td id='m_audiotracks' class='false'>?</td> <td id='m_videotracks' class='false'>?</td> <td id='m_texttracks' class='false'>?</td> </tr> </tbody> </table><!-- <div id='canPlayType'> // @@TODO canPlayType </div> --> </div><h2>Limitations</h2> <ul> <li>Some events may be fired despite the lack of effect on the video itself, in particular while trying to change the playback rate;</li> <li>Not all conditions are necessarily there to generate every single possible event, such as <code>stalled</code> or <code>emptied</code>;</li> <li>The rendered page may not be updated while the video is playing. Thus you may see plenty of <code>timeupdate</code> events counted after the video is paused/stopped;</li> </ul><hr><address>plh@w3.org - November 2014</address><p>Many, many thanks to the <a href='http://www.blender.org/blenderorg/blender-foundation/'>BlenderFoundation</a> and the community around the <a href='http://www.blender.org/'>blender project</a>.</p> </body></html>
```


What else can I do with HTML5 video (like a pro)?



Back up your stuff
Shutdown the VM gracefully: *vagrant halt*