Application Primers

By Joseph Medley
Technical Writer

Much media work requires changing characteristics of media files, such as bitrate or resolution. Finding a straightforward way to get started can be bewildering and intimidating. In this section, I intend to provide an easy onramp into that world.

You'll find two articles in this section. On this page, I provide some basic instruction in using two common media command-line utilities: Shaka Packager and ffmpeg. Why cover two applications? While both are powerful and useful by themselves, neither does everything needed to prepare media for the web.

The next page is a cheat sheet showing common operations with those applications. Corrections and additions to that part are welcome.

Media file characteristics

One thing you'll do often is look at the characteristics of a video: resolution, bitrate, codecs, and so on. Since this is just about the easiest thing to do in both Shaka Packager and ffmpeg, let's use this to get comfortable with these packages.

When we peek into a media file, we're going to see many file characteristics. For this article, I'm only focusing on characteristics in the cheat sheet.

Let's start with streams. Media files can almost be thought of as multiple files in one. The multiple "files" are called *streams*. A media file can have any number of streams, of more types than I will go into here. My examples will contain at most two, an audio stream and a video stream. (Among the other types you might encounter are captions and data, both of which are beyond the scope of this article). There are many instances where audio and video streams are dealt with separately.

There are several characteristics that apply to each stream.

Bitrate is the maximum number of bits used to encode one second of a stream. The more bits used to encode a second of stream, the higher the potential detail and fidelity.

Resolution is the amount of information in a single frame of video, given as the number of logical pixels in each dimension. For example, a resolution of 1920 by 1080 works out to 1080 stacked horizontal lines, each of which is one logical pixel high and 1920 logical pixels wide. This resolution is frequently abbreviate 1080p because technically the width can vary. The numbers I've given produce an aspect ratio of 16:9, which is the ratio of movie screens and modern television sets. By the way this is the resolution defined as full HD.

When you look at the file characteristics using Shaka Packager and ffmpeg, you'll notice that the word 'resolution' doesn't appear. What the two applications output are just the dimensions, the numbers themselves.

Codec, which is short for *coder-decoder*, is a compression format for video or audio data. This is not the same as a file format. Think of the file format as the container and the codec as a way of arranging what's in the container. Different codecs are used for audio and video streams. Many file formats support multiple codecs for the same stream type. A complete list of available codecs would be a whole website itself. Listed below are the currently preferred codecs for mp4 and webm files.

Video

Extension	Codec
mp4	H264
webm	VP9

Audio

Extension	Codec
mp4	aac
webm	vorbis, opus

Shaka Packager

Shaka Packager is a free media packaging SDK for creating DASH/HLS packager applications with common encryption support, Widevine DRM support, live video, and video-on-demand. Despite what it says on the packaging, this utility is for more than C++

developers. It can be used as both a library for building media software and as a command-line utility for preparing media files for playback. It's the later capacity that interests me here. In fact, for web media creators, Shaka Packager is the only way to do some tasks without spending money on expensive commercial applications.

Here's the basic pattern for a Shaka Packager command line.

```
packager stream_descriptor[ stream_descriptor-2[ stream_descriptor-n]] [fla
```

This isn't quite what you get if you type **`packager -help`**. This is how I think of it, and this reflects the examples in the Shaka Packager README. Note the multiple *stream_descriptors*. This is useful for manipulating the video and audio streams of the same file simultaneously.

Compare this basic pattern with something simple like displaying file characteristics. (I've lined up equivalent parts.)

```
packager stream_descriptor[ stream_descriptor-2[ stream_descriptor-n]] [fla
```

```
packager input=glocken.mp4                                          --dump_st
```

I've shown the output below.

```
[0416/140029:INFO:demuxer.cc(88)] Demuxer::Run() on file 'glocken.mp4'.
[0416/140029:INFO:demuxer.cc(158)] Initialize Demuxer for file 'glocken.mp4'.

File "glocken.mp4":
Found 2 stream(s).
Stream [0] type: Video
 codec_string: avc1.640028
 time_scale: 30000
 duration: 3225222 (107.5 seconds)
 is_encrypted: false
 codec: H264
 width: 1920
 height: 1080
 pixel_aspect_ratio: 1:1
 trick_play_rate: 0
 nalu_length_size: 4

Stream [1] type: Audio
 codec_string: mp4a.40.2
 time_scale: 48000
 duration: 5161379 (107.5 seconds)
 is_encrypted: false
 codec: AAC
 sample_bits: 16
```

```
 num_channels: 2
 sampling_frequency: 48000
 language: eng

Packaging completed successfully.
```

Look for the characteristics discussed in the last section and notice a few things. The height and width are correct for full HD, and the audio and video codecs are the preferred codecs for their container types, AAC for audio and H264 for video. Notice also that streams are identified with numbers. These are useful for operations that manipulate the audio and video separately.

Notice that it doesn't show the bitrate. Despite what's missing, It's easier to read, which is why I use it whenever I can. When I need information that Shaka Packager can't get, such as the bitrate, I use ffmpeg.

ffmpeg

ffmpeg is also a free application for recording, converting, and streaming media files. I won't say its capabilities are better or worse than Shaka Packager's. They're just different.

The basic pattern for an ffmpeg command looks like this:

```
ffmpeg [GeneralOptions] [InputFileOptions] -i input [OutputFileOptions] out
```

Like Shaka Packager this application can handle multiple streams. Also, some of the options can be used in multiple locations and have different meanings depending on where they are in the command.

I'll again compare the basic pattern to the example for displaying file characteristics.

```
ffmpeg [GeneralOptions] [InputFileOptions] -i input        [OutputFileOptio

ffmpeg                                     -i glocken.mp4
```

This is technically an incorrect usage of ffmpeg. In addition to the output I care about, I'll see an error message, as shown in the example below.

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'glocken.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
```

```
    compatible_brands: isomiso2avc1mp41
    encoder         : Lavf57.56.100
  Duration: 00:01:47.53, start: 0.000000, bitrate: 10715 kb/s
    Stream #0:0(eng): Video: h264 (High) (avc1 / 0x31637661), yuvj420p(pc), 1920x
    Metadata:
      handler_name    : VideoHandler
    Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp
    Metadata:
      handler_name    : SoundHandler
At least one output file must be specified
```

So that's a few bits about media manipulation software in a nutshell. Now jump next door to the cheat sheet. Check back every few weeks as we continue to add new content about media for the web.

Last updated July 2, 2018.