

Flash Video Encoding Basics

Well here we are once again. Oh Flash video how long will you **rule the virtual video universe**? Will **HTML 5 kill you** dead Flash? Is the new **HTML 5 Video Player** signaling the end of days? Has Adobe laid-off off too many talented people to still be **innovate**?

No matter, Flash video is here now, works now and is still the top of the heap. In this tutorial we'll be running through the basics of video encoding, bringing in that video into flash via a URL connection (externally called video) and controlling that video.

Recording video and audio to a digital format involves balancing quality with file size and bitrate. Most formats use compression to reduce file size and bitrate by selectively reducing quality. Compression is essential for reducing the size of movies so that they can be stored, transmitted, and played back effectively.

When exporting a movie file for playback on a specific type of device at a certain bandwidth, you choose an encoder (**codec**). Various encoders use various compression schemes to compress the information. Each encoder has a corresponding decoder that decompresses and interprets the data for playback.

A wide range of codecs is available; no single codec is best for all situations. For example, the best codec for compressing cartoon animation is generally not efficient for compressing live-action video.

Compression can be *lossless* (in which no data is discarded from the image) or *lossy* (in which data is selectively discarded).

You can control many of the factors that influence compression and other aspects of encoding in the Export Settings dialog box

- Flash is no longer the only way to show video online, but it is the most reliable
- You can use “cue points” to start other actions inside flash
- You can apply Pixel Bender filters to your videos inside of flash (fun)

RESOURCES

- [Flash Video Basics Form Adobe | Link](#)
- [Flash video \(FLV\) bitrate calculator, Adobe | Link](#)
- [Optimal frame dimensions for Flash video, Adobe | Link](#)

- [Video Learning Guide for Flash: Capturing and encoding, Adobe | Link](#)
- [Playing Flash Video Using The AS3 FLVPlayback Component, Republic of Code | Link](#)
- [Create An XML Video Player Using ActionScript 3.0, Republic of Code | Link](#)
- [Introduction to Pixel Bender: Part 1, gotoAndLearn, | Link](#)
- [Pixel Bender basics for Flash, Adobe | Link](#)
- [ActionScript 3 Sound Basics, gotoAndLearn\(\) | Link](#)
- [Tutorial – How to Add Video in Flash, School of FLash | Link](#)
- [Full Screen Flv Video and SWFs: Flash CS4 Tutorial, YouTube | Link](#)

REQUIRED READING

- Flash CS 5.5, The missing Manual; Pg 411-425 - All video related Chapters
- Flash Professional CS5 Essential Training; Ch 18 & 19 – Working With Video

Vocabulary

Cue points cause the video playback to start other actions within the presentation. For example, you can create a Flash presentation that has video playing in one area of the screen while text and graphics appear in another area. A cue point placed in the video starts an update to the text and graphic, while they remain relevant to the content of the video.

Bitrate specifies the number of megabits per second of playback for the encoded file. (This setting is available only if you select CBR as the Bitrate Encoding option.)

The following options appear only if you select VBR as the Bitrate Encoding option:

Encoding Passes specifies the number of times the encoder will analyze the clip before encoding. Multiple passes increase the time it takes to encode the file, but generally result in more efficient compression and higher image quality. (Adobe After Effects doesn't support multiple encoding passes.)

Set Bitrate is available only for the QuickTime format. Select to keep the bitrate of the output file constant.

Bitrate [kbps] is available only for the QuickTime format. Select if you want to determine the bitrate. Then, drag the slider until the hot text displays the desired value.

Maximum Bitrate [Kbps] specifies the maximum bitrate you want the encoder to allow.

Average Video Bitrate [Kbps] specifies the average video bitrate you want the encoder to allow.

Target Bitrate [Mbps] specifies the average video bitrate you want the encoder to allow when encoding video using the H.264 video codec.

Peak Video Bitrate [Kbps] specifies the top bitrate you want the encoder to allow.

Video Encode Basics (Step One)

Flash is a bit picky. You need to have your video in a specific few video **Codec** (compression) for the Flash Player to work. So our first task is to get our large HD video encoded properly. There are many

tools that can be used for that but we'll be using Adobe Media Encoder that ships with Flash for this exercise.

Overview

- Choose your Codec wisely; one encode to rule them all (cough, h.264)
- Every video is unique so there is no one size fits all for encoding
- Always start with the highest quality source
- Avoid fast pans and zooms
- Avoid fades and dissolves
- Steady footage is best

ENCODING RESOURCES

- [Flash video \(FLV\) bitrate calculator, Adobe | Link](#)
- [Optimal frame dimensions for Flash video, Adobe | Link](#)
- [Video Learning Guide for Flash: Capturing and encoding, Adobe | Link](#)
- [Audio Sample Rate, Wiki | Link](#)

ACCEPTABLE FLASH CODECS

Video

- Sorenson Spark
- h.264
- On2 VP6

Audio

- AAC
- ADPCM
- MP3
- HE-AAC

ENCODING TOOL OPTIONS

- [Adobe Media Encoder, CommunityMX | Link](#)
- [Using Compressor to Encode HD Content for Vimeo, Ken Stone | Link](#)

Supported video formats

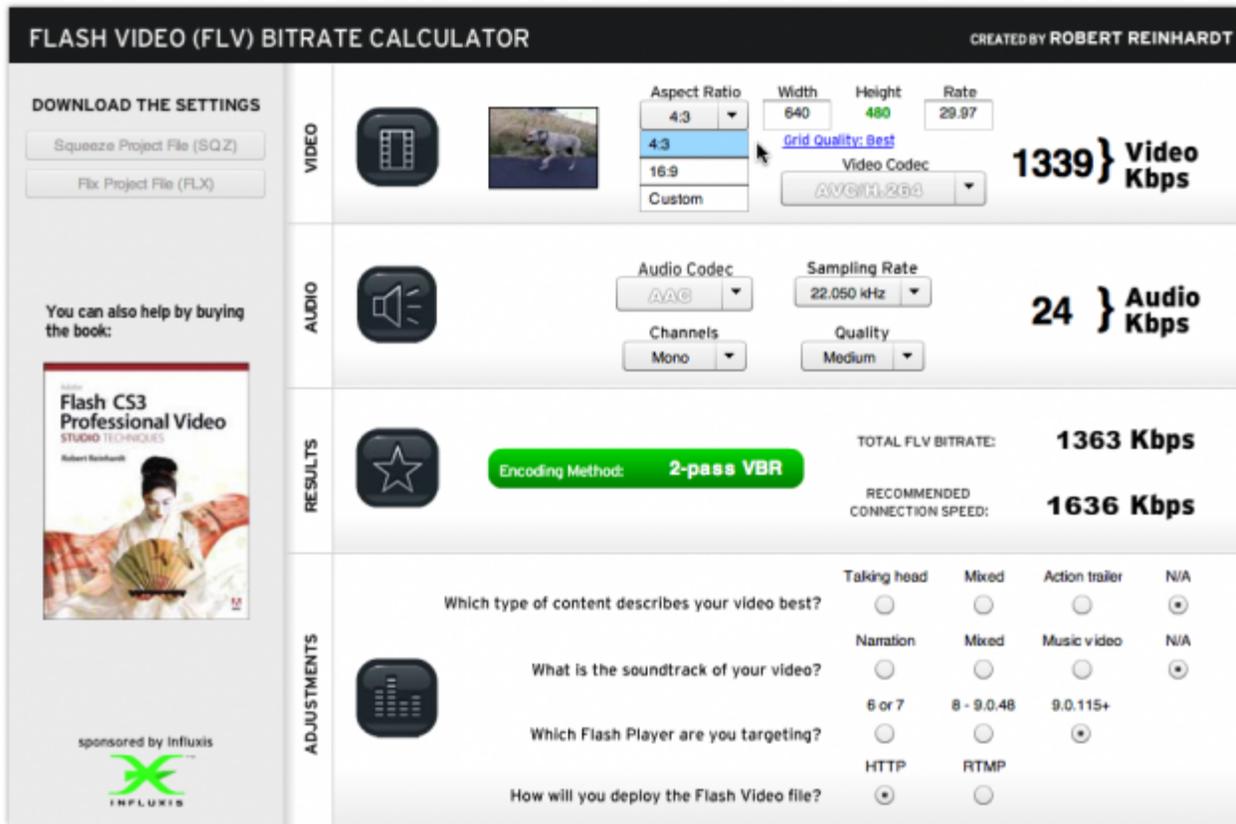
FLV	Most played video format on the web. YouTube and Google Videos are using it .	Basically all clients with Flash capability can see these videos. Has been there since version 6.	Referred to a proprietary file format and does not offer as good video and audio quality as H.264 does.
-----	---	---	---

H.264	A video compression standard .	Provides good video quality at lower bit rates than FLV. As a standard it is not Apple's proprietary format.	Supported since Flash version 9.115 which is used by 81.7% of all Flash users.
MP4	a video compression standard that is a subset of the whole H.264 standard family	same advantages as in H.264	There seems to be many different flavors of MP4 and only some of them are supported by Flash. You can always try a given file to see if it plays in Flowplayer, if it does not the most likely reason is that it is not compatible with Flash.

Encoding Hints

1. Export out of your video editor > Save for Quicktime > h.264
2. If you intend on using Cue Points in your Flash app you'll need to use the .f4v option
3. Watch video, analyze content
4. Choose bandwidth target (who's your audience?)
5. Choose target bitrate based on target audience.
6. Choose audio bitrate; higher for music, lower for voice only
7. Choose video dimension; keep ratio the same to avoid distortion; *use a multiple of 16*
8. Smaller videos can be higher quality with smoother playback
9. Framerate should match your source
10. Set keyframe interval; can be tricky, more keyframes larger file
 - a. • Choose automatic in your encoder
 - b. • To force keyframe, add cuepoint
 - c. • Fewer keyframes the better
 - d. • More movement in clip requires more keyframes

The Adobe Bitrate Calculator | <http://tinyurl.com/adobe-bitrate-calculator>



Aspect Ratio: As we go back a look at what our QuickTime Movie Inspector told us our Movie is 768×576 or 4:3 in ratio so set it to 4:3

Width: Where's a hint, click on the blue "Grid Quality" link this will give us our best options for sizes as recommended by Adobe. Let's select 512×384 a nice middle sized video (grid suggested) Read up, it's good intel.

Height: Set Automatically because you selected the predefined aspect ratio

Frame Rate: As a rule keep this at 29.97. You can half it, but your audio may drift and end up looking like an old Kung Fu movie.

Video Codec: There's a lot of options (and opinions) but the most flexible is the h.264

codec.there is some question about future liability issues The main reason is users can view this file type with out the flash player, power to the people!

Audio: Our goal is this set-up to to balance the desire for quality and download speed. It's a bit easier to give up a bit of quality in the audio end of things.

- **Audio Codec:** Honestly, AAC works just fine and works with our encoder
- **Sample rate:** Affects the quality of the audio, the lower the number the less the quality. To offer an example CD quality would be 44.1

Encoding Steps

WHAT YOU GOT SON?

First things first, it helps to know where you are to know where you are going.

1. Download the exercise files from the class blog.
2. Open the high-rez video > basic-flash-video-exercise > source-to compress
3. Control-click on the .mov file and Open With > Quicktime Pro
4. When you .mov is open mouse over to Window > Show Movie Inspector
5. The things we're checking for are the format, size and data rate remembering what we just learned about those topics (see Adobe bit-rate calculator)
6. Open the Adobe Media Encoder
7. Place the high-rez .mov file into the Queue window
8. Here's where it gets fun (har har), Adobe has some great pre-tweaked encode options for us and in fact Vimeo SD is the default.
9. Change your preset to the Vimeo HD option
10. That's fine place to start but we need a unique size. Use the default settings a starting point for our next step. Easy, right?
11. Click on the Vimeo preset text or Control Click > Edit.
12. There are many settings you can adjust in the Adobe Media Encoder. To get a detailed overview visit <http://tinyurl.com/adobe-media-encoder-basics>.
13. Click on the video tab
14. Based on your original high-rez .mov file select a proportional file size. In our case we'll use 768 432.
15. Scroll down and adjust the bitrate
16. On the right side of the window select a small section of the video, say 10 seconds. TO do this grab the yellow triangles and make a short portion selected.
17. Close the edit window and click the play button located in the upper right
18. Get a hard drink
19. Play the test clip; inspect the quality and file size. Adjust keyframes/bitrate as needed.
20. Encode full clip