

Encoding H.264 Video for Streaming and Progressive Download

Jan Ozer
Doceo Publishing
[jozer@mindspring.com/](mailto:jozer@mindspring.com)
[276-238-9135](tel:276-238-9135)



Agenda

- Technology Overview
- Understanding H.264 encoding parameters
- Producing H.264 video for computers and devices
- Comparing the H.264 codecs
- Settings for common encoders (Adobe Media Encoder, Apple Compressor, Sorenson Squeeze, Telestream Episode Pro)
- Bonus topics: Accessing YouTube's and Facebook's 720p delivery modes



Techie Overview



- Generic encoding parameters
 - Delivery options
 - Constant and variable bitrate encoding
 - I, B and P frames
- H.264 overview
 - What is H.264?
 - What does H.264 cost?

Generic Streaming Concepts



- Delivery options
 - Progressive download
 - Streaming
 - Multiple file delivery
- Constant and variable bit rate encoding
- Working with I, B and P frames

What are My Delivery Options?



- Progressive download
 - What people think it is
 - trailers that take hours to download but then play smoothly because they're cached on your hard drive.
 - What it really is
 - Video distributed from a web server rather than a streaming server
 - Video is typically stored locally
 - The vast majority of video is distributed via progressive download (YouTube, ESPN, CNN, etc)

What are My Delivery Options?



- Streaming
 - What people think it is
 - Video that plays immediately when you click and smoothly to the end
 - What it really is
 - Video distributed via a streaming server
 - May or may not be cached locally

Delivery Mode – Key Points



- If streaming
 - Ask if there any server specific requirements
 - Ask if cached on viewer's computer
 - If not, use data rate that ensures uninterrupted delivery
 - usually produced with constant bit rate encoding to produce a smooth stream

Delivery Mode – Key Points



- If progressive
 - May be some specific encoding requirements (Fast Start for QuickTime, MOOV atom for Flash)
 - Produce at streaming rates to ensure responsive user experience (most web sites)
 - Or, produce at higher bit rates to ensure high quality video
 - Movie trailer scenario
 - Usually encode with variable bit rate encoding (for best quality)

Considering a Streaming Server?



Pros

- More secure stream (not cached)
- Bandwidth negotiation (dynamic streaming)
- Enables DRM
- More efficient streaming (video doled out rather than burst)
- Better statistics

Cons

- Cost
- Complexity

Multiple File H.264 Streaming



- Concept
 - Customize file for viewer device and bandwidth
 - Adapt to changing conditions
 - Requires some intelligent component (Flash Media Server/IIS Extension)
- Multiple options
 - Adobe's Dynamic Streaming
 - Move Networks Adaptive Streaming
 - Microsoft's Smooth Streaming
 - H.264 Scalable Video Coding (not here yet)

Availability



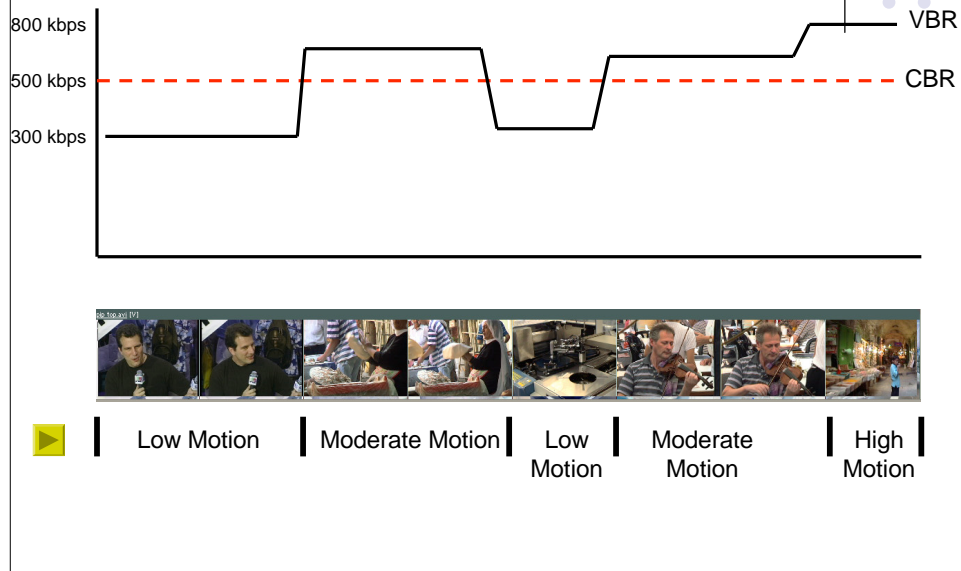
- Dynamic Streaming
 - Now (VP6 and H.264)
- Smooth Streaming
 - Silverlight 3 player (for H.264 compatibility)
- Adaptive Streaming
 - VP6/7 today, H.264 coming
- Scalable Video Coding
 - Down the road (1-2 years out)

CBR and VBR



- What's the difference?
- When should I use them?
- How do I produce the optimal CBR file?
- How do I produce the optimal VBR file?

Constant vs Variable Bit Rate



Constant vs Variable Bit Rate

- Constant Bit Rate (CBR)
 - One bit rate applied to entire video, irrespective of content
 - Pros: Easy and fast
 - Cons: Doesn't optimize quality

Constant vs Variable Bit Rate



- Variable Bit Rate (VBR)
 - Dynamic bit rate matches motion in video
 - Pros: Best quality
 - Cons: Slow, can produce erratic stream

When Should I Use VBR/CBR?



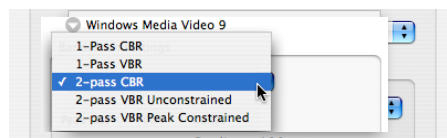
- Consider VBR when:
 - Clips are longer than 60 seconds – too short, no difference
 - Varied motion in clip (some action, some talking head) – all similar – no difference
 - Producing for ***progressive download***

When Should I Use VBR/CBR?



- Consider CBR when:
 - In a hurry (or live encoding)
 - Producing for **streaming** (as opposed to progressive download – to produce more consistent stream)
 - Consistent motion (especially talking head – if no varied motion, no reason to use VBR)

How do I Produce the Best Quality CBR?

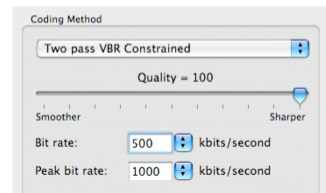
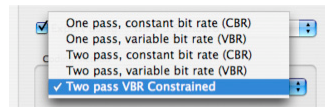


- Use 2-pass CBR when available
 - Scans file (like VBR), but packs data into a consistent stream
 - Best of both worlds when available
- 1-pass of live or draft work

How Do I Produce the Optimal VBR File?



- 2 passes or more
- Use “Constrained”
 - Constrains to data rate
- Set Target and Max/Min
 - Overall target – 500 kbps
 - Max/Peak bit rate – how high rate can go when varying
 - Rule of thumb is 1.5 - 2X of target
 - If minimum setting, use .5x



Working with I, B and P Frames



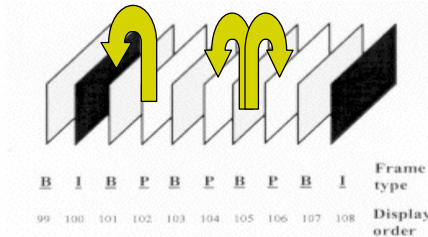
- Defined
- Relevant controls



What are I, B and P Frames?



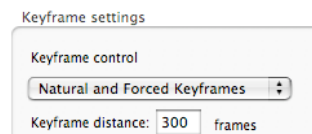
- I-Frame - encoded without reference to other frames (also called Key Frames)
- P - looks backward to I and P frames (predicted)
- B - looks forward and backward to previous I and P frames (Bi-directional interpolated)



What do I Need to Know About Key Frames?



- Least efficient frame
 - from quality perspective, fewer is better
- Key frames enhance interactivity
 - All playback starts on a key frame
 - When seeking to random frame, must start playback at key frame
 - Max interval should be 5-10 sec.
- Key frames "reset" quality:
 - Useful at scene changes
 - Enable natural key frames

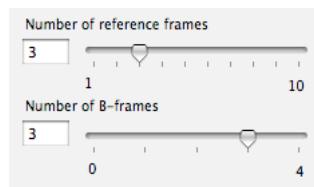


What do I Need to Know About B Frames?



- The most “efficient” frame
 - So improves quality (comparisons to come)
- Hardest to decode
 - Use for computer playback (Main and High)
 - Not for devices (not in Baseline Profile)

Typical B-Frame Encoding Parameters



- Number is number of B frames between I and P-Frames; (IBBBPBBBBPBBBBPBBBP)
 - 2 - 3 is recommended (I use 3)
- Reference frames
 - Number of frames searched for redundancies
 - 3-5 is recommended setting (I use 5)

About H.264



- What is H.264?
- What's H.264 cost?
- What's with all the H.264 extensions?

What is H.264?



ITU – International Telecommunications Union Telephone, Radio, TV		ISO – International Standardization Organization Photography, Computer, Consumer Electronics	
1984	H.120		
1990	H.261 – Video Conferencing		
1993		MPEG-1 – Video CD	
1994	(H.262)	MPEG-2 – Digital Cable and Satellite TV	
1995	H.263 – Improved Video Conferencing		
1997		ATSC – U.S. HDTV	
1999		MPEG-4	
2002	AVC (H.264)	AVC (MPEG-4 Part 10)	

- Adapted by ISO and ITU
 - Telephony/cellular
 - TV - consumer electronics
 - Computer electronics
- Only codec adopted by top three streaming providers (Apple, Adobe, Microsoft)

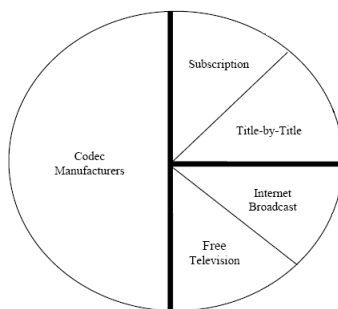
Streamcrest Associates
<http://www.streamcrest.com/SDF%20Final1.pdf>

What's H.264 Cost?



- MPEG-LA -
 - “For AVC video delivered via the Internet to an end user who does not pay for the right to view, i.e., neither title-by-title nor subscription, there will be no royalty through December 31, 2010”
 - **Are all AVC essential patents included?** No assurance is or can be made that the License includes every essential patent.”
 - AT&T has threatened to sue multiple companies, including Apple over MPEG-4 patents

Royalty Classes - Content



- Paying customers
 - Subscriptions
 - Pay per view
- Free broadcast
 - TV
 - Internet broadcast

Royalties – Internet Broadcast



- In the case of Internet broadcast (AVC video that is delivered via the Worldwide Internet to an end user for which the End User does not pay) ... there will be no royalty during the first term of the License (ending December 31, 2010) and after the first term the royalty shall be no more than the economic equivalent of royalties payable during *the same time for free television*.

H.264 Royalties – Free TV



- One of two royalty options:
 - (i) a one-time payment of \$2,500 per AVC transmission encoder (for each AVC encoder used by Licensee in transmitting AVC video or
 - (ii) annual fee per Broadcast Market
 - \$2.5K/year/market - 100K - 500K households,
 - \$5K/year/market - 500K – 1M households
 - \$10K/year/market – 1 M+

Impact



- Internet –unique views of all H.264 video
 - If under 100K, probably no royalty
 - If > 100K, probably royalty
 - Different internet sites might generate royalty obligation
 - Pod casts are definitely covered
- Intranet – same deal, separate market
 - If under 100K, probably no royalty
 - If > 100K, probably royalty
- Either way:
 - VC-1 – same structure (and duplicative)

What is an MP4 file (and what are the variants)?



- .MP4 - official MPEG-4 wrapper
- .M4V - Apple's variant for iTunes and devices
- .MOV - H.264 file for editing or QuickTime delivery
- .F4V - H.264 for Flash
- .3GP - (not shown) - phone
- .MPG - H.264 in MPEG-2 transport stream
- W4M? - will Microsoft create a new extension?

File Format:	MPEG-4	:
Extension:	mp4	<input checked="" type="checkbox"/> Allow Job Segmenting

File Format:	H.264 for Apple Devices	:
Extension:	m4v	<input checked="" type="checkbox"/> Allow Job Segmenting

File Format:	H.264 for DVD Studio Pro	:
Extension:	mov	<input checked="" type="checkbox"/> Allow Job Segmenting

Stream Type	F4V
Video-Basic	MPEG-2 Transport Stream
Width	MPEG-4 System
Height	F4V
	Raw H.264 Stream

H.264 Encoding Parameters



- The basics
 - Profiles
 - Levels
 - Entropy encoding
- Stream related options
- Search related options
- Miscellaneous options

What are H.264 Profiles?



- “Define a set of coding tools or algorithms that can be used in generating a bitstream”

	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive
I and P Slices	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B Slices	No	Yes	Yes	Yes	Yes	Yes	Yes
Multiple Reference Frames	Yes	Yes	Yes	Yes	Yes	Yes	Yes
In-Loop Deblocking Filter	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CAVLC Entropy Coding	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CABAC Entropy Coding	No	No	Yes	Yes	Yes	Yes	Yes
Interlaced Coding (PicAFF, MBAFF)	No	Yes	Yes	Yes	Yes	Yes	Yes
8x8 vs. 4x4 Transform Adaptivity	No	No	No	Yes	Yes	Yes	Yes
Quantization Scaling Matrices	No	No	No	Yes	Yes	Yes	Yes
Separate Cb and Cr OP control	No	No	No	Yes	Yes	Yes	Yes
Separate Color Plane Coding	No	No	No	No	No	No	Yes
Predictive Lossless Coding	No	No	No	No	No	No	Yes
	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive

Main vs. Baseline



Main vs. Baseline



- Critical to know your target profile before encoding
- Device - iPod/iPhone - Always Baseline
- Computer playback - typically Main
 - QuickTime - need latest version of QuickTime Player to play High profile (so consider Main in the short term)
 - Flash - should be compatible with all relevant profiles

What are H.264 Levels?



- “Constrains key parameters in the bitstream”

Level number	Max video bit rate (VCL) for Baseline, Extended and Main Profiles	Max video bit rate (VCL) for High Profile	Examples for high resolution @ frame rate (max stored frames in Level)
1	64 kbit/s	80 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1b	128 kbit/s	160 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1.1	192 kbit/s	240 kbit/s	176x144@30.3 (9) 320x240@10.0 (3) 352x288@7.5 (2)
1.2	384 kbit/s	480 kbit/s	320x240@20.0 (7) 352x288@15.2 (5)

H.264 Levels



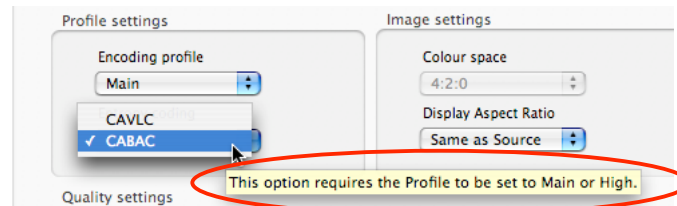
- Primarily an issue when encoding for devices
 - Must ensure that encoding parameters are within target *level* (most templates do this)
- For computer playback,
 - The major concern is the playback performance of viewer's computer, not the level
 - If level too low for selected encoding parameters, just raise the level (Episode does this automatically) to equal or exceed parameters

Level signalling

Video level Auto (Recommended) ▼

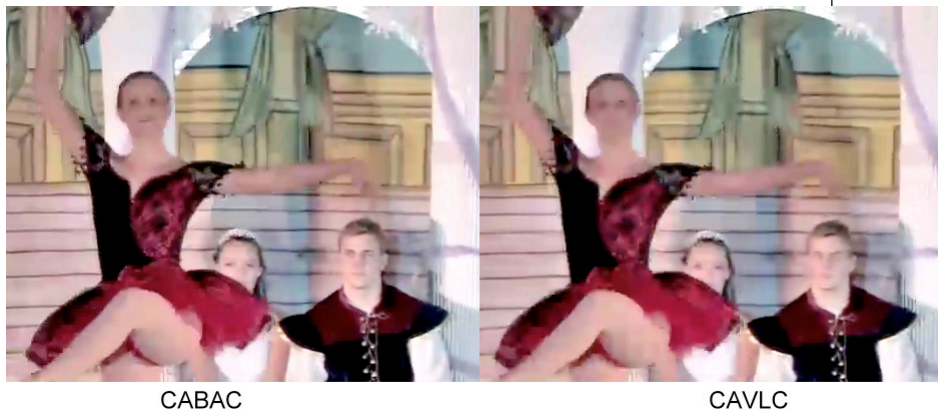
If a level that is too low for current configuration is selected, the lowest valid level will be used.

Entropy Encoding



- CABAC (Context-adaptive binary arithmetic coding)
 - More efficient (e.g. better quality), but harder to decode
- CAVLC (Context-adaptive variable-length coding)
 - Less efficient, easier to decode
- Big question - does quality improvement outweigh increase in required CPU horsepower

CABAC vs CAVLC Quality



- In challenging scenes, CABAC was noticeably better
- Most authorities place quality advantage at 10-15%

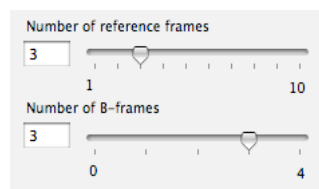
CABAC vs CAVLC Performance



	CABAC	CAVLC
HP 8710w - Core 2 Duo (% of both CPUs)	31.1%	30.5%
PowerMac - Dual 2.7 GHz PPC G5 (% of 1 processor)	71.17	67.34

- Does increase playback requirements slightly on lower power computers
- My recommendation:
 - CABAC - unless really concerned about low power computers (schools, etc)

Reference/B-frames



- Reference frames
 - Frames searched by P frames
 - beyond 3-5 is diminishing returns (most redundancies are close)
- B-frame - most efficient
 - setting is for number in sequence
 - 2-3 is optimum

B-frames - Yes/No



B-frames - Yes/No



B-frames - Yes/No

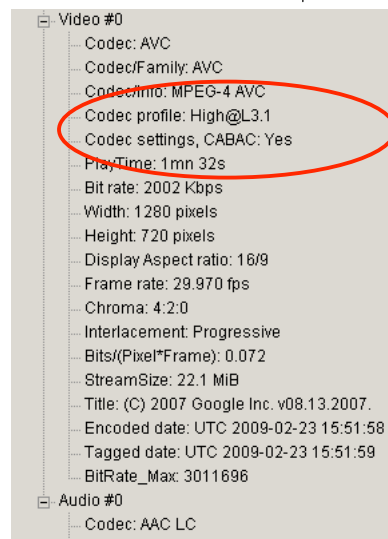


- Noticeable quality improvement
- 5-10% increase in required decode CPU horsepower
- Recommend
 - Say “YES” to B-frames
 - 2-3 is a good number for real world video

Some Perspective - YouTube 720P



- High Profile
- CABAC
- 2 B-frames (not shown in chart)
- 3 Reference frames (ditto)

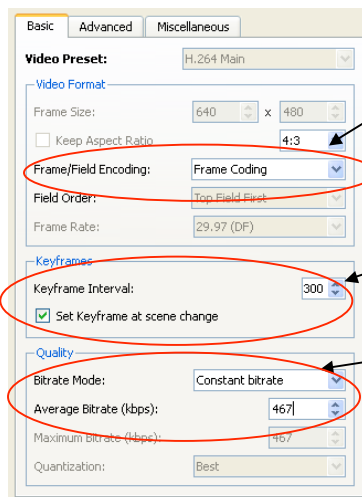


Deep Dive into H.264 Parameters



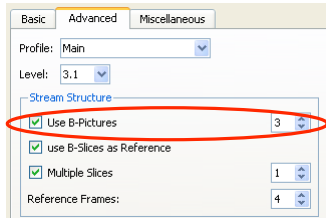
- Caveats:
 - Presented differently by each encoding tool
 - Only have time to cover most critical parameters
- To understand *your* encoder
 - Read manual/help file to understand parameters and their trade-offs; generally involve
 - Encoding time vs. quality
 - Complexity (and maybe encoding time) vs. quality
- Use MainConcept's reference encoder to illustrate

General Options



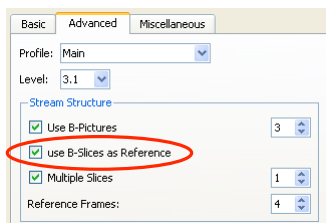
- Frame coding (progressive)
- Key frame interval
 - 300
 - Scene change detection
- Bitrate
 - CBR/VBR
 - Insert data rate

Stream Structure



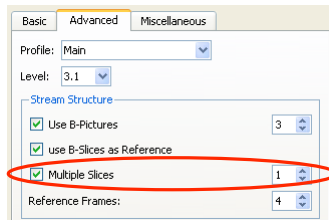
- Stream Structure
 - Should you use B-frames
 - Yes
 - Number of sequential B frames used in the file
 - 2-3

Stream Structure



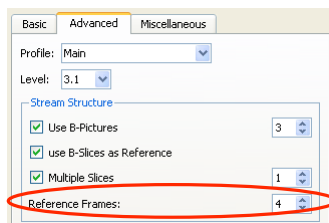
- B-Slices as references?
 - Use B-frames as references for other B frames
 - Also called B-frame pyramid when enabled as reference for B-frames
 - Impact
 - Potential quality improvement (more redundancies)
 - Potential increase in encoding time

Stream Structure



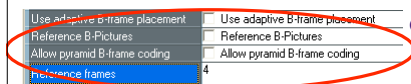
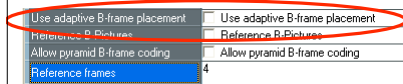
- Multiple Slices
 - Divides image up into “slices”
 - May speed encode on MP systems
 - May speed decode on MP systems
 - May limit quality
 - Searching only occurs within respective slice (Episode help file)
 - Recommend - 1, unless in a hurry

Stream Structure



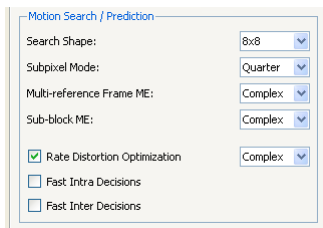
- Reference Frames
 - Number of frames searched for redundancies
 - Higher numbers may improve quality but lengthen encode time
 - Increases decode requirements
 - All referenced frames must be open to decode frame
 - Recommend
 - 4 - 5 for real world video
 - potentially more for animation

More B-frame Options



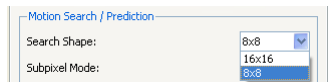
- Adaptive B-frame placement
 - Overrides B-frame quantity when necessary to improve quality
 - Like scene change
 - Always enable
- Reference B-Pictures/Pyramid B-frame coding
 - Use when B-frames greater than 1
 - Adds decode complexity

Search/Prediction Related



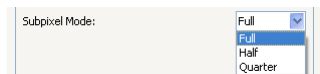
- In general, these manage the trade-off between search accuracy (and quality) and search time
 - Can improve quality
 - Minimal (if any) impact on required decompression

Search Shape



- Search shape is the size of the macro block used for searching redundancies
 - 8x8 faster, lower quality
 - 16x16 slower, better quality
- Use 16x16 unless in a huge hurry

Sub-pixel Mode



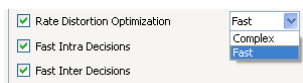
- Sub-pixel mode
 - Dictates depth of search shape
 - Full is faster but less accurate
 - Quarter is slower but may improve quality
 - Use Quarter unless in a real hurry

Motion Estimation (ME) Algorithms



- Multi-reference Frame ME /Sub-block ME
 - Complex for quality (and longer encoding time)
 - Fast for encoding speed and less quality

Other Search-Related Options



- Rate distortion manages quality/data rate trade-off
 - Fast - heuristic optimization that's faster, but can cause quality loss
 - Complex is slower, with better quality
- Fast Intra/Inter Decisions
 - Speed/quality trade-offs in decision metrics

Search Related Quality



- Observations:
 - Time differential can be +/- 50% on tested encoders
 - Quality difference was minimal on 2 of 3 tested encoders. I recommend:
 - Check the help file
 - Episode - help file says that settings beyond 50 typically won't increase quality
 - Test at both extremes to ID time/quality difference
 - Apply to your unique encoding situation

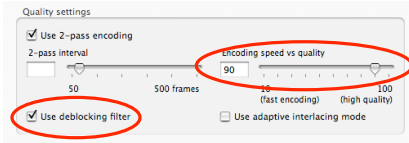
Hadamard Transformation



<input type="checkbox"/> Use fast inter and intra decisions	<input type="checkbox"/> Use fast inter and intra decisions
<input type="checkbox"/> Use fast multi-reference frame mo...	<input type="checkbox"/> Use fast multi-reference frame motion estimation
<input type="checkbox"/> Use fast sub-block motion estimat...	<input type="checkbox"/> Use fast sub-block motion estimation
<input type="checkbox"/> Use rate distortion optimization	<input checked="" type="checkbox"/> Use rate distortion optimization
<input checked="" type="checkbox"/> Use Hadamard transformation	<input checked="" type="checkbox"/> Use Hadamard transformation

- Enable/Disable Hadamard Transformation
 - Encoding technique to improve compressed quality
 - Yes - Better quality, longer encoding time
 - HP xw8400 DP QC, one minute SD file
 - 62 seconds with Hadamard enabled
 - 60 seconds without
 - No noticeable qualitative difference, but minimal affect on encoding time
 - Unless in a real hurry, opt yes

Deblocking Filter/Other



- Deblocking filter
 - Minimizes blockiness, but extends encoding time and can slow decompression
 - Recommend - always enable
- Encoding Speed vs. quality
 - How encoder vendors combine multiple options into one slider
 - Varies by encoder/codec

H.264 Encoding Summary

- Profiles/Levels - dictated by target
- Entropy Encoding - typically CABAC
 - Not available for Baseline
- B-frames - use when available
 - Not available for Baseline

H.264 Encoding Summary



- Divide other parameters into:
 - Boost quality/increase encoding time
 - Boost quality/increase decoding complexity (and perhaps encoding time)
 - As we'll see, max delta is around 10%
 - Make decision based upon your target viewers and encoding workload
 - More later

Producing H.264 Video for Computers and Devices

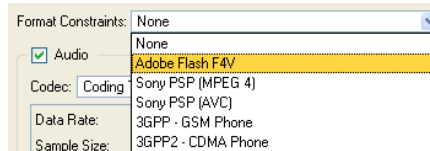


- Format specific considerations
 - Flash
 - QuickTime
 - Silverlight
- Optimizing for computer playback
- Optimizing for devices

Flash and H.264



- Flash Player 9 Update 3 contained:
 - Software H.264 decoder (Baseline/Main/High profiles)
 - AAC decoder
- Hardware scaling to full-screen mode, but NO acceleration at normal resolution
- No special encoding requirements for the Flash Media Server
- Flash player can play mp4, m4v, mov and .3gp files
 - Evolving best practice - FLV for VP6 and F4V for H.264



Flash and the MOOV Atom



- MOOV atom contains file header info
 - If not located at the start of the file, video delivered via progressive download won't start until fully downloaded
 - Video streamed via FMS is OK
- Most pre-CS4 Adobe encoders put moov atom at the end
- Good description of problem here:
<http://www.scottgmorgan.com/blog/index.php/tag/video/>

Flash and the MOOV Atom

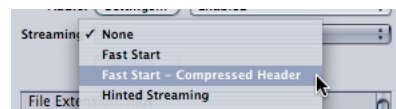


- How to get moov atom to the beginning of the file? Several utilities
 - <http://svn.mplayerhq.hu/ffmpeg/trunk/tools/qt-faststart.c?view=markup>
 - <http://renaun.com/blog/2007/08/22/234/>
- Should be fixed with CS4 encoders, but not yet verified

Producing for QuickTime



- Customizing for QuickTime Streaming
 - When distributing via a streaming server, opt of Hinted Streaming
- For progressive download, use Fast Start - Compressed Header
 - Otherwise file may completely download before playing



Producing for Silverlight 3



- File requirements
 - Self-contained .mp4 (including .f4v and .m4a) and .mov file formats
 - Simple, Main, and High 4:2:0 profiles (progressive only)
 - AAC-LC audio mono or stereo (HE AAC will play back with lower fidelity, as in QuickTime)
 - Local files or http progressive download.
- Or, sliced another way, Silverlight 3 will play pretty much all MPEG-4 files that would play back well in both QuickTime and Flash.

<http://blogs.iis.net/benwagg/archive/2009/03/18/silverlight-3-beta-what-s-new-for-media.aspx>

Optimizing H.264 for Computer Playback



- Perspective
 - We understand H.264 encoding params
 - We understand QuickTime/Flash specifics
- Now we learn how to configure a stream that will smoothly play on the lowest platform you care about

H.264 Playback - SD File



	Dell Latitude	HP xw4100	MacBook Pro	Dell Precision 390
	1600 MHz Pentium M	3.0 GHz P4 with HTT	2.4 GHz Core 2 Duo	3.0 GHz Core 2 Duo
SD Tests				
H.264 - Main - tricked out (QT Player)	88 %	25 %	29 %	12 %
H.264 - Baseline (QT Player)	80 %	30 %	19 %	8 %

H.264 Playback - 720p File



	Dell Latitude	HP xw4100	MacBook Pro	Dell Precision 390
	1600 MHz Pentium M	3.0 GHz 4 with HTT	2.4 GHz Core 2 Duo	3.0 GHz Core 2 Duo
HD Tests				
H.264 - High	99 %	78 %	50 %	28 %
H.264 - Baseline	100 %	68 %	58 %	21 %

H.264 Playback - 1080i File



	Dell Latitude	HP xw4100	MacBook Pro	Dell Precision 390
	1600 MHz Pentium M	3.0 GHz P 4 with HTT	2.4 GHz Core 2 Duo	3.0 Ghz Core 2 Duo
HD Tests				
H.264 - Main - tricked out (QT Player)	100 %	69 %	48 %	40 %
H.264 – Baseline (QT Player)	100 %	79 %	42 %	26 %

H.264 Compared to Other Codecs



- 720p playback tests
- H.264 requires less CPU to playback than VP6 or Silverlight

	Flash VP6E	Flash H.264 - High	Silverlight
HP xw4100, 3.0 GHz P4 with HTT Processor CPU during playback Drop frames	54.6% Yes	45.1% No	52.5% No
HP 8710P, 2.2 GHz Core 2 Duo Processor CPU during playback Drop frames	51.9% No	34.8% No	47.3% No
Precision 390, 2.9 GHz Core 2 Duo Processor CPU during playback Drop frames	22.7% No	7.7% No	26.0% No

My Take on H.264 Quality



- H.264 produces better quality than VP6 or VC-1
- If converting from either codec, assume that you can use the same encoding parameters with no loss in quality
- If a new implementation, start with closest template
 - Set key frame interval to 150 - 300, with scene detection
 - Use High profile
 - Enable B-frames and set to 2, reference to 3
 - Enable CABAC
 - Use stats on following page as a guide for data rate

Lessons from the Field



The Good	Width	Height	Total Pixels	Frame Rate	Video Data Rate	Bits per pixel
Cranky Geeks	640	360	230,400	29.97	500	0.072
YouTube - H.264	1280	720	921,600	29.97	2002	0.072
Apple battery video	848	480	407,040	29.97	1089	0.089
Facebook - high rez	1280	720	921,600	29.97	2463	0.089
Facebook - low rez	576	324	186,624	24	428	0.096
The Bad						
McKenzie	774	440	340,560	15	585	0.115
TED Talks (BMW)	432	240	103,680	24	351	0.141
Wal-Mart	640	426	272,640	29.97	1,538	0.188
MacBreak	640	360	230,400	24	1078	0.195
The Ugly						
CNET	320	180	57,600	29.97	492	0.285
Government Tech News	320	240	76,800	29.97	739	0.321
Wine Library	640	480	307,200	29.97	4096	0.445

Bits Per Pixel

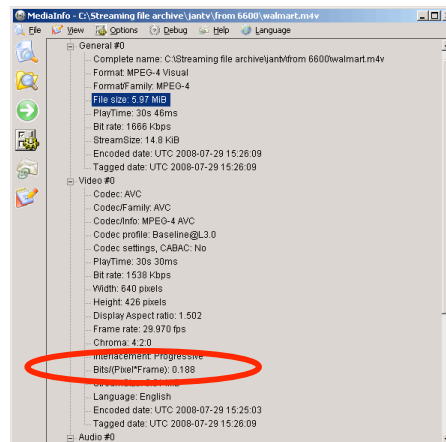


- In general
 - .1 or lower should be fine
 - .15 is very conservative; beyond that is a waste
 - The larger the resolution, the lower you can go
 - .1 @ 320x240 could be dicey
 - .1 @ 720p should be fine
 - Formula
 - $\text{Data rate} / (\text{pixels} \times \text{frame rate})$

Bits Per Pixel



- Or, get MediaInfo



mediainfo.sourceforge.net/en

Lessons from the Field



	Profile	CABAC
Cranky Geeks	Main	No
YouTube - H.264	High	Yes
Apple battery video	Main	No
Facebook - high rez	High	Yes
Facebook - low rez	High	Yes
TED Talks (BMW)	Baseline	NA
MacBreak	Baseline	NA
CNET	Baseline	NA
Government Tech News	Baseline	NA
Wine Library	MPEG-4	NA

Producing for Computers



- Mind your format specific parameters
- Choose profile, resolution and other parameters that ensure smooth playback on your target
 - Or, offer multiple files and let viewer decide which to download
 - In general, if you're converting over from another codec, H.264 will be similar in quality and required playback horsepower to other codecs

Optimizing for Devices



- Digesting iPod/iPhone Specs
- Lessons from iTunes
- Recommendations

iPod Specs



	Original iPod (pre-5g)	iPod Nano	iPod Classic	iPod Touch/ iPhone
Device resolution	320x240	320x240	320x240	480x320
Aspect Ratio	4:3	4:3	4:3	16:9-ish
Video codec	H.264	H.264	H.264	H.264
Data rate	768 kbps	2.5 Mbps	2.5 Mbps	2.5 Mbps
Resolution	320x240	640x480	640x480	640x480
Frame rate	30 fps	30 fps	30 fps	30 fps
Profile	Baseline Profile to Level 1.3	Baseline Profile up to Level 3.0	Baseline Profile up to Level 3.0	Baseline Profile up to Level 3.0
Audio codec	AAC-LC	AAC-LC	AAC-LC	AAC-LC
Data rate	160 kbps	160 kbps	160 kbps	160 kbps
Audio parameters	48 kHz, stereo	48 kHz, stereo	48 kHz, stereo	48 kHz, stereo
Formats	m4v/mp4/mov	m4v/mp4/mov	m4v/mp4/mov	m4v/mp4/mov



Lessons from iTunes

- Best practices of current producers
 - Downloaded and analyzed 50 podcasts from iTunes
 - Review standard encoding parameters
 - Key mistakes that prevented podcasts from playing on iPod
 - Optimizing 16:9 video



Encoding Parameters - Video

	Size	Frame Rate	Codec H.264/ MPEG-4	Aspect 4:3/ 16:9	Data Rate	Extension mov/m4v/mp4	Key Frame
Small	320x240 (25/44)	4 - 15f 21 - 30f	22/2	20/5 (2 letterbox)	605K average	2/13/10	94 average
					low - 83K hi - 1.4 mbps		low - 24 hi - 300
Large	640x360+ (19/44)	1-15f 4-24f 14-30f	18/1	13/6	1.281 mbps average	2/11/5	121 average
					low - 813K hi - 2 mbps		low - 32 hi - 300

Encoding Parameters - Audio



- All used Low Complexity AAC audio
- Average data rate - 116,000+
- Stereo/Mono - 42/2
- Low data rate - 32kbps/High - 160 kbps

Errors that Prevented Playback



- Main or High Profile - 5
- Exceed data rate - 4 (high of 6.5 mbps)
- Exceed resolution - 2
- Wrong codec - 1 (Sorenson Video 3)

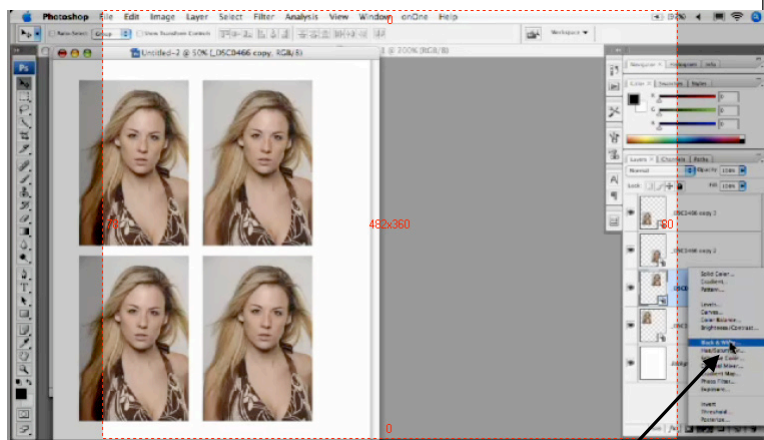
Note that there were some duplication of errors. A total of six videos wouldn't load, including videos produced by HBO, E-Online and Discovery Channel

Optimizing 16:9 Video



- If producing 16:9 video, note that:
 - iPod Touch/iPhone is 16:9, all others 4:3
 - At default configuration, 4:3 players will display “center cut”, cutting off right and left edges
 - OK for many videos, but a potential problem when critical content is on the edge

Center-Cut - Screenshot



Screen action not visible in 4:3 display

Center-Cut - Real World Video



Oprah - logo cut off, but shot with 4:3 safe zones

When Producing 16:9 Video

- Shoot for center cut display (like Oprah)
- Or, instruct viewers to change default playback parameters from “center cut” to letterbox
 - Videos > Settings > Fit to Screen > Off

Sub-Optimal Encoding Decisions



- Data rates above ~ 1.3 for 640x480
 - Typically minimal quality improvement; just longer downloads and increased space consumption
- Key frame intervals below 100
 - Can cause pulsing

Recommended Encoding Parameters



	320x240	640x480 ¹
Video codec	H.264 codec, Baseline profile	H.264 codec, Baseline profile
Data rate	768,000/CBR	1,120,000/CBR
Key frames	150 - 300	150 - 300
Frame rate	match source	match source
Audio	AAC Low	AAC Low
Data rate	128 kbps/stereo	128 kbps/stereo
Extension	.mv4	.mv4

¹ From Compressor

Comparing the H.264 Codecs



- Test description
 - Apple, Dicas (Episode Pro), Main Concept (Carbon Coder/Squeeze)
 - Three files
 - SD - 640x480@30 fps, 468/32, 2-pass VBR, highest supported profile/quality options
 - HD - 1280x720@30 fps, 800/128, 2-pass VBR, highest supported profile/quality options

Test Description



- Still image quality
 - Grab frames and compare
- Motion quality
 - Compare quality during real time playback
 - Look for motion artifacts like banding and mosquitoes
- Smoothness
 - Whether codec/encoder dropped frames at selected parameters

HD Samples



- All codecs perform well with low motion footage

HD Samples



- Walking around (and panning) tends to separate the contenders (note detail in curtain)

HD Samples



- Jumping further separates the contenders

HD Test Results



	Apple	Episode Dicas	Carbon Coder Main Concept
Still Quality	3	2	1
Motion Quality	2	1	1
Smoothness	1	1	1
Total	6	4	3



HD Test Analysis

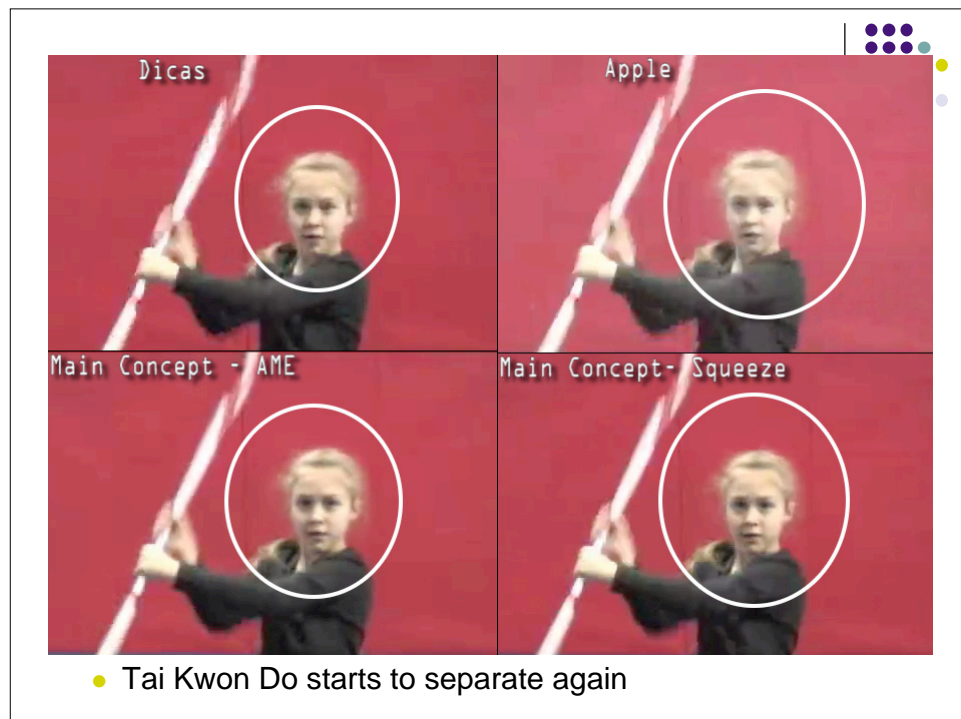


- Difference between Carbon Coder and Episode is commercially irrelevant
 - Carbon costs \$4,000 more
 - Viewers wouldn't notice absent side-by-side comps (which, of course, they never have)
- Apple's is much more significant
 - Avoid if seeking highest quality at lowest bitrate
 - At about 2.5 mbps, the quality is nearly the equivalent of the others

SD Comparisons






- Still
- Motion
- Results
- Summary





SD Test Results



	Apple	Episode Dicas	AME - Main Concept	Squeeze Main Concept
Still Quality	2	1	1	1
Motion Quality	2	1	1	1
Smoothness	1	1	1	1
Total	6	3 	3 	3 

SD Test Analysis



- No meaningful difference between Dicas and Main Concept in SD clips
- Apple again lagging
 - Though fine for low volume SD production where data rate isn't critical
- Squeeze has resolved key frame related issues that plagued initial 5.0 release.

What this means for encoding tools



- Encoding tools follow codec
- Dicas/Main Concept
 - MC slightly better in HD, but not commercially significant
 - Nearly identical in SD
- Apple
 - OK choice for SD
 - Avoid for HD when trying to achieve optimal data rate

Settings for Common Encoders



- Apple Compressor
- Adobe Media Encoder CS4
- Sorenson Squeeze
- Telestream Episode Pro

Apple Compressor - Compression Settings



Compression Type:
Choose codec here

Key Frames:
- Choose Automatic to select scene change detection

Frame Reordering:
- Uncheck for Baseline Profile
- Check for Main with 1 B frame

Encoding:
- Best - multi-pass
- Faster - single-pass

Optimized for:
- Streaming: CBR
- Download: VBR

Standard Video Compression Settings

Compression Type: H.264

Motion

Frame Rate: Current fps

Key Frames: ☒ Automatic ☐ Every ☐ All

☒ Frame Reordering

Compressor Quality

Least Low Medium High Best

Encoding: ☒ Best quality (Multi-pass) ☐ Faster encode (Single-pass)

Data Rate

Data Rate: ☐ Automatic ☒ Restrict to 672 kbits/sec

Optimized for: Streaming

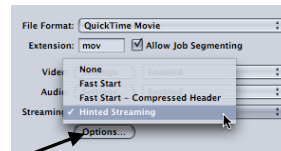
Preview

Apple Compressor - Streaming Related Settings



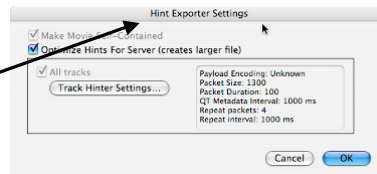
Streaming:

- None - disc based playback
- Fast start - Compressed Header - web, no streaming server
- Hinted Streaming: QuickTime Streaming Server



Click here:

To open this:



Adobe Media Encoder - Format



Format:

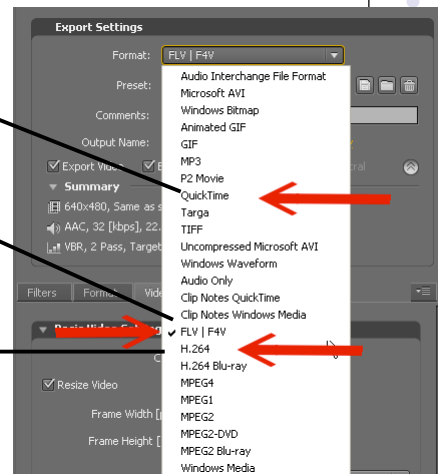
- QuickTime for MOV extension

Format:

- FLV/F4V for Flash

Format:

- H.264 for devices



AME - QuickTime Video

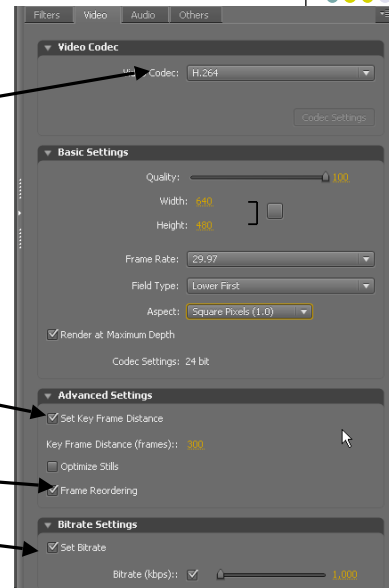


Codec:
- H.264

Set key Frame Distance:
Change to 300 (otherwise 30):

Click Frame Reordering to
enable B-frames

Click Set Bitrate to set bitrate

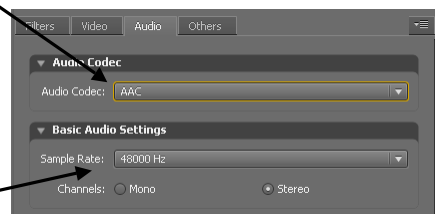


AME - QuickTime - Audio



Codec:
- H.264

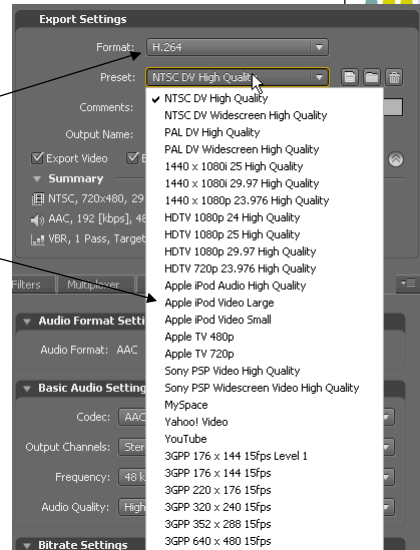
**Choose Sample Rate and
channels:**
AME chooses the data rate



AME - Device

Format:
- Choose H.264

Then, choose device:

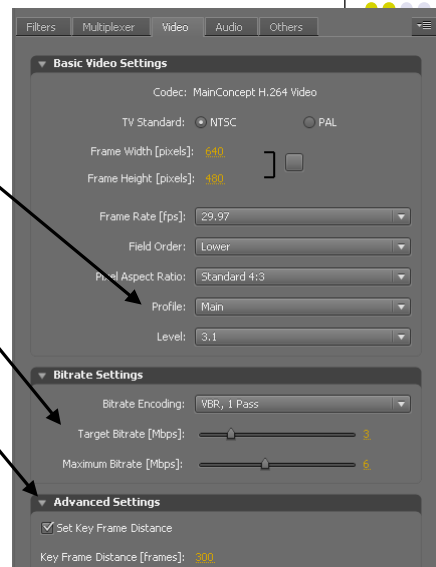


AME - Device - Video

Profile/Level:

Bitrate settings

Click Set key Frame Distance:
Otherwise 30



AME - Device - Audio

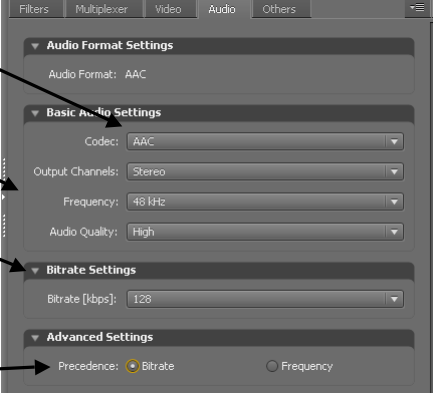


Choose AAC:

Basic audio parameters

Choose bitrate

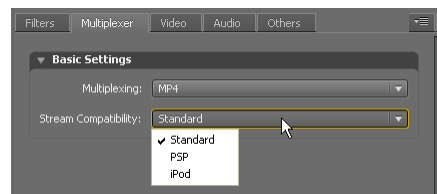
Prioritize bitrate over frequency



AME - Device Multiplexing



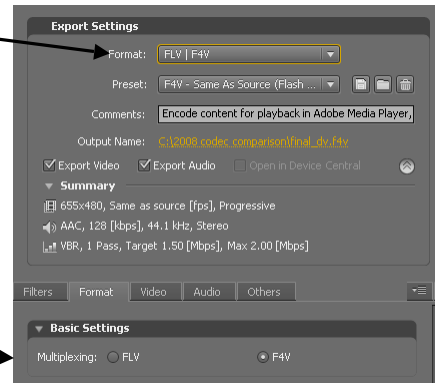
Click Multiplexer tab:
To access 3GPP or MP4 settings. MP4 has settings for Standard, PSP and iPod



AME - Flash - Format



Format:
-Choose FLV|FV4

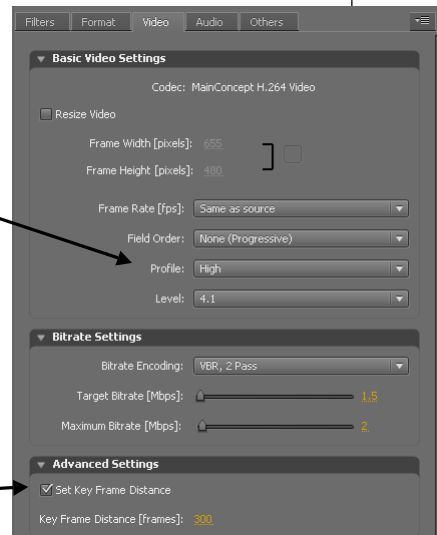


Click Format tab:
FLV for VP6, FV4 for H.264

AME - Flash - Video



Profile/Level:
Set to desired parameters



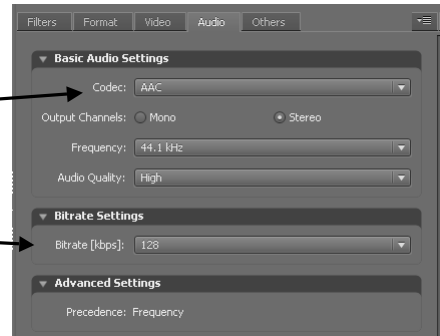
Set key Frame Distance
(otherwise 30):

AME - Flash - Audio



Codec:
Choose AAC

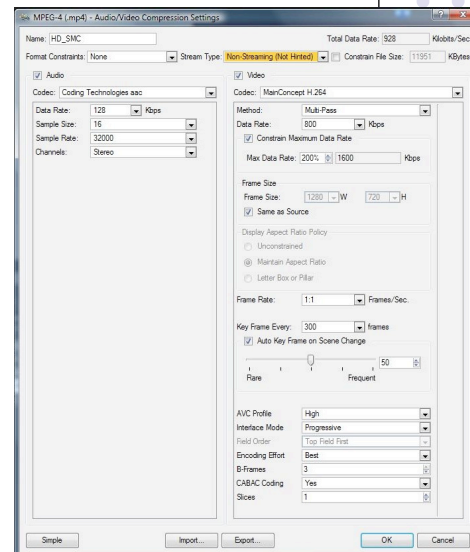
**Normal parameters and
bitrate control**



Sorenson Squeeze - Interface



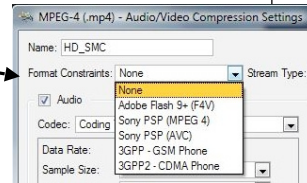
- Template screen
- Format choices on top
- Audio on the left
- Video on the right



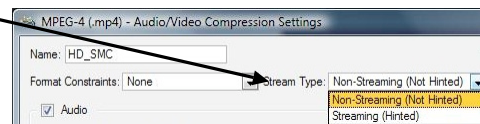
Sorenson Squeeze - Basics



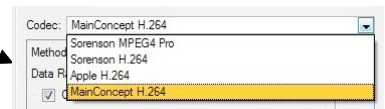
Format Constraints:



Stream Type:
- Hint or not hint



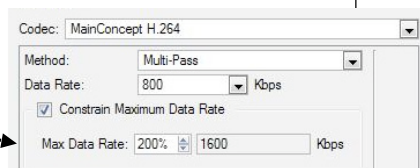
Codec:
Multiple H.264/MPEG-4
codecs; MainConcept is the
best (and the default in 5.0)



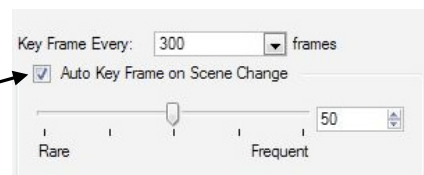
Sorenson Squeeze - Data Rate and Key Frames



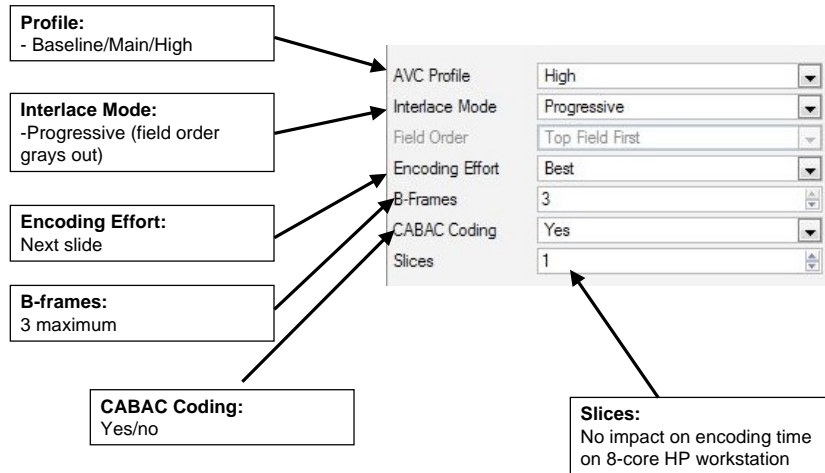
Max Data Rate:
Resolved key frame issue -
set to 200%



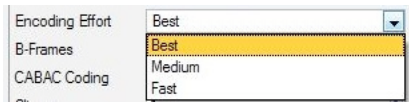
Set key Frame Distance
Enable Auto Key



Sorenson Squeeze - H.264 controls



Sorenson Squeeze - Effort Matters



- Assume amalgam of search functions
- Best produces noticeably better quality in challenging scenes



Best: 63:23 to encode 1 minute file

Fast: 48:34 to encode 1 minute file

Sorenson Squeeze



Audio Codec:

Use Coding Technologies AAC for compatibility

Audio

Codec: Coding Technologies aac

Data Rate: Coding Technologies aacPlus v2

Sample: AMR-NB

Sample: ATSC A/52

Other parameters:

As normal

Data Rate: 128 Kbps

Sample Size: 16

Sample Rate: 32000

Channels: Stereo

Episode Pro



- Output format
- Video settings
- Audio settings
- Switch using tabs atop the Job Batch Window

Output Format: MP4 File Extension: f4v

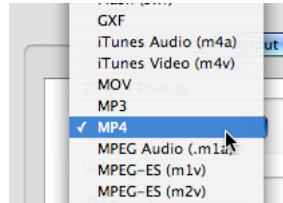
Video: Encode H264

Audio: Encode AAC

Hint: Prepare For Streaming

Hint for streaming

Episode - Output Format



- MP4 or MOV?
 - Both play in QuickTime Player
 - Both play in Flash Player
 - MP4 plays in more devices (e.g. Zune)
- Recommend MP4

Episode - General



Recommended technique unless streaming (CBR)

Set average (target and Peak (2X))

Natural and forced ten seconds recommended

Frames searched when creating P/B frame (5 max recommended)

B-frames 2-3 recommended

Episode - Profile and Quality



Profile (Main or High unless device)

Entropy Encoding (CABAC unless device)

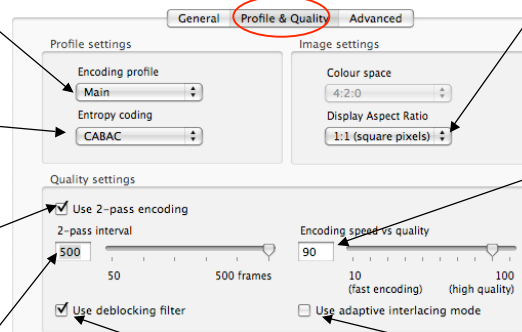
Use 2-pass unless draft work

2-pass interval - set to max (number of frames searched before encoding pass).

Always use deblocking filter

Display aspect ratio use 1:1

Encoding speed - settings over 50 generally show little improvement - I use 90



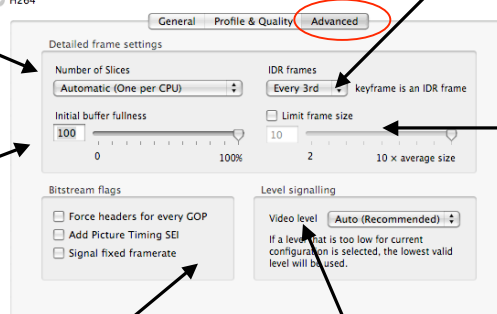
Episode - Advanced



1 slice (for max quality)/may slow encoding

Initial buffer fullness (use default)

H264



Blu-ray or set top options

Auto level

IDR frames (use default)

Limit frame size (don't enable)

Episode - Audio



Audio: Set Bit rate (use Low Complexity)

Press AAC Setting Guidelines to open chart

Channels: Choose

Sample Rate : Make sure within recommended sample rate for bit rate setting

The screenshot shows the AAC audio settings window. The 'Bit Rate' is set to 128 Kbit/s and 'AAC Mode' is set to Low Complexity. The 'Keep Codec Delay' checkbox is checked, and 'Parametric Stereo' is unchecked. The 'Channels' are set to Stereo and the 'Sample Rate' is 44100 Hz. The 'AAC Setting Guidelines' button is highlighted with an arrow pointing to a 'Recommended Sample Rates' chart. The chart shows the following data:

Bitrate [kbit/s]	Mono samplerate [kHz]	Stereo samplerate [kHz]
8	8-12	8-12
16	8-24	8-12
20	11-24	8-12
24	11-32	11-24
28	11-32	11-24
32	11-48	11-24
40	16-48	16-32
48	22-48	22-32
56	22-48	22-48
64	32-48	32-48

Below the table, it states: 'All higher bitrates match 32-48 kHz. Mono sound is only supported up to 160 kbit/s, stereo sound only up to 320 kbit/s. Surround sound requires bitrates above 160 kbit/s, 7.1 channels at least 224 kbit/s.'

Encoding Tool Summary



- All present different parameters in different ways
- Generally two trade-offs
 - Trade encoding time for video quality
 - Trade quality (and perhaps encoding time) for decompression complexity
- Understand and categorize parameters, and apply them to your unique situation

For More Training?



- 2 one-week summer courses
 - Stanford University in Palo Alto, CA (July 20-24)
 - Harvard in Boston, MA (July 27-31),
- The course is offered in two sections
 - 3 days – Producing Video for the Web – focuses on pre-production
 - 2 days – Web Video Compression & Delivery– concentrates on streaming encoding and delivery
- Very hands on
 - all students get their own computers, class size limited

For more information, go to <http://www.digitalmediaacademy.org> or call 866-656-3342.

YouTube's Delivery Options



- The problem - no docs
- What they are
- Upload requirements

New HQ Mode - Compared



	Low Quality	High Quality	HD
Codec	Spark (H.263)	Spark (H.263)	H.264 (high)
Encoded Resolution	320x180 (16:9) 320x240 (4:3)	640x360 (16:9) 480x360 (4:3)	1280x720
Displayed resolution	480x360	640x360	853x480
Data rate	286 (16:9) kbps	628 kbps	1.995 mbps
Frame rate	24 fps	24 fps	24 fps
Audio	64 kbps, stereo	96 kbps, mono	125 kbps stereo

New HQ Mode



New HQ Mode



Accessing HQ Mode



- Upload file with resolution of 480x360 or higher
 - Still 10 minute limit, which means max per second data rate of 13 mbps
- New 1 GB limit for uploaded file

<http://www.pcmag.com/article2/0,2817,2330990,00.asp>

Accessing HD Mode



- Upload file with resolution of 720p
 - Higher resolutions didn't convert to 720p
 - Still 10 minute limit, which means max per second data rate of 13 mbps
- New 1 GB limit for uploaded file

<http://www.streaminglearningcenter.com/articles/18/1/YouTube-does-720P-HD-using-H264/Page1.html>

Facebook Video



- The problem - no docs
- What it is
- How to get it

New HQ Mode - Compared



	Regular Quality	HD
Codec	H.264 (high)	H.264 (high)
Encoded Resolution	576x324	1280x720
Displayed resolution	760x430	760x430
Data rate	428 kbps	2267 kbps
Frame rate	24	24
Audio	stereo/94 kbps	stereo/149 kbps

Accessing HD Mode



- Upload file with resolution of 720p
 - 20 minute limit, which means max per second data rate of 6.5 mbps
- 1 GB limit for uploaded file