

Flix Engine Windows

8.0.17.2

Generated by Doxygen 1.6.1

Tue Nov 2 15:37:47 2010

Contents

1	Flix Engine Documentation	1
1.1	Page Contents	1
1.2	Introduction	1
1.3	Starting Points	2
1.4	New Codecs & Container Formats	2
1.4.1	Flash Player Support and AVC Licensing	2
1.5	Support Options & FAQ	2
1.6	Upgrading to Flix Engine 8.0.6.0	2
2	About Flix Engine	5
2.1	Version Information	6
2.2	Third Parties	6
3	FE2_VP6_CONCURRENCY Performance	9
3.1	Intel Pentium 4 540	10
3.1.1	320x240 (QVGA)	10
3.1.2	640x480 (VGA)	11
3.1.3	1024x464	12
3.1.4	1280x720 (720p)	12
3.2	Intel Xeon X5355	13
3.2.1	320x240 (QVGA)	13
3.2.2	640x480 (VGA)	14
3.2.3	1024x464	15
3.2.4	1280x720 (720p)	15
4	CHANGELOG	17
5	Upgrade Notes	31
5.1	Page Contents	32
5.2	Overview	32

5.3	Important: COM ID	32
5.4	Flix Engine Improvements	33
5.4.1	Exception Handling	33
5.4.2	PNG Thumbnails	33
5.4.3	Concurrent Encodes	33
5.4.4	Sane Naming Conventions	33
5.4.5	New COM Interface is Deprecation-Aware	33
5.4.6	Audio Decoding Doesn't Require Audio Hardware	33
5.5	API Reference Shortcuts	33
5.6	Functions with No Equivalent in the New Flix Engine	34
6	Samples	37
6.1	CGI	39
6.1.1	flx2_sample.asp	39
6.1.2	process_sample.asp	80
6.2	ASP .NET	86
6.3	CGI	86
6.3.1	flx2_sample.aspx	87
6.3.2	process_sample.aspx	128
6.4	Command Line	134
6.5	C#	138
6.6	Command Line	138
6.7	ColdFusion	141
6.8	CGI	141
6.8.1	flx2_sample.cfm	142
6.8.2	process_sample.cfm	182
6.9	Java	186
6.10	Command Line	187
6.11	Perl	190
6.12	Command Line	190
6.13	CGI	192
6.13.1	flx2_sample.cgi	192
6.13.2	process_sample.cgi	233
6.14	Command Line	237
6.15	CGI	240
6.15.1	flx2_sample.php	240
6.15.2	process_sample.php	281

6.15.3	process_sample.php4 - PHP4 processing	284
6.15.4	process_sample.php5 - PHP5 processing	287
6.16	Command Line	290
6.17	Visual Basic .NET	294
6.18	Command Line	294
7	Deprecated List	297
8	Module Index	315
8.1	Modules	315
9	Namespace Index	317
9.1	Namespace List	317
10	Data Structure Index	319
10.1	Data Structures	319
11	File Index	321
11.1	File List	321
12	Module Documentation	323
12.1	Audio Encoding Options	323
12.1.1	Enumeration Type Documentation	324
12.1.1.1	FE2_AudioBitrates	324
12.2	Deprecated	325
12.2.1	Enumeration Type Documentation	326
12.2.1.1	FE2_AudioSamplingrates	326
12.2.1.2	FE2_FlvAudioFormat	326
12.2.2	Function Documentation	326
12.2.2.1	audio_options_GetBitrate	326
12.2.2.2	audio_options_GetFlvAudioFormat	327
12.2.2.3	audio_options_GetSamplingrate	327
12.2.2.4	audio_options_GetStereo	328
12.2.2.5	audio_options_Reset	328
12.2.2.6	audio_options_SetBitrate	329
12.2.2.7	audio_options_SetFlvAudioFormat	329
12.2.2.8	audio_options_SetSamplingrate	330
12.2.2.9	audio_options_SetStereo	330
12.2.2.10	audio_options_Validate	331

12.3	Codecs	332
12.3.1	Detailed Description	332
12.4	Common Codec Parameters	333
12.4.1	Define Documentation	333
12.4.1.1	FE2_CODECPARAM_BITRATE	333
12.4.1.2	FE2_VCODECPARAM_KFFREQ	333
12.4.1.3	FE2_VCODECPARAM_KFINTTYPE	334
12.4.1.4	FE2_VCODECPARAM_RC_MODE	334
12.4.2	Enumeration Type Documentation	334
12.4.2.1	FE2_VideoBitrateControls	334
12.4.2.2	FE2_VideoKeyframeTypes	334
12.5	Video Codecs	335
12.6	Audio Codecs	336
12.6.1	Define Documentation	336
12.6.1.1	FE2_CODEC_PCM	336
12.7	AAC	337
12.7.1	Detailed Description	337
12.7.2	Define Documentation	338
12.7.2.1	FE2_AAC_BITRATE	338
12.7.2.2	FE2_CODEC_AAC	338
12.8	AAC+	339
12.8.1	Detailed Description	339
12.8.2	Define Documentation	340
12.8.2.1	FE2_AACPLUS_BITRATE	340
12.8.2.2	FE2_AACPLUS_PARAMETRIC_STEREO	340
12.8.2.3	FE2_CODEC_AACPLUS	340
12.9	AMR_NB - FFmpeg	341
12.9.1	Detailed Description	341
12.9.2	Define Documentation	342
12.9.2.1	FE2_AMR_BITRATE	342
12.9.2.2	FE2_CODEC_AMR_NB	342
12.10	H263 - FFmpeg	343
12.10.1	Detailed Description	343
12.10.2	Define Documentation	343
12.10.2.1	FE2_CODEC_H263	343
12.10.2.2	FE2_CODEC_H263_BASELINE	344

12.10.2.3 FE2_H263_BITRATE	344
12.10.2.4 FE2_H263_KFFREQ	344
12.10.2.5 FE2_H263_KFINTTYPE	344
12.10.2.6 FE2_H263_MAX_Q	345
12.10.2.7 FE2_H263_MIN_Q	345
12.10.2.8 FE2_H263_RC_MODE	345
12.11H264	346
12.11.1 Detailed Description	347
12.11.2 Apple device support	347
12.11.3 Define Documentation	347
12.11.3.1 FE2_CODEC_H264	347
12.11.3.2 FE2_H264_B_FRAME_RATE	347
12.11.3.3 FE2_H264_BITRATE	348
12.11.3.4 FE2_H264_KFFREQ	348
12.11.3.5 FE2_H264_KFINTTYPE	348
12.11.3.6 FE2_H264_PROFILE	348
12.11.3.7 FE2_H264_RC_MODE	349
12.11.3.8 FE2_H264_SPEED	349
12.11.4 Typedef Documentation	349
12.11.4.1 h264profile_t	349
12.11.5 Enumeration Type Documentation	349
12.11.5.1 h264profile	349
12.12MP3 - LAME	350
12.12.1 Detailed Description	350
12.12.2 Define Documentation	351
12.12.2.1 FE2_CODEC_LAME	351
12.12.2.2 FE2_LAME_BITRATE	351
12.12.2.3 FE2_LAME_CHANNELS	351
12.12.2.4 FE2_LAME_QUALITY	351
12.12.2.5 FE2_LAME_RC_MODE	352
12.12.3 Typedef Documentation	352
12.12.3.1 lame_rcmode_t	352
12.12.4 Enumeration Type Documentation	352
12.12.4.1 lame_rcmode	352
12.13libvorbis - FFmpeg	353
12.13.1 Detailed Description	353

12.13.2 Define Documentation	353
12.13.2.1 FE2_CODEC_VORBIS	353
12.13.2.2 FE2_VORBIS_BITRATE	353
12.14 VP6	354
12.14.1 Detailed Description	356
12.14.2 Defaults dependent on bits per pixel	356
12.14.3 Notes on datarate control	356
12.14.4 Define Documentation	357
12.14.4.1 FE2_CODEC_VP6	357
12.14.4.2 FE2_VP6_2PASS_MAX_SECTION	357
12.14.4.3 FE2_VP6_2PASS_MIN_SECTION	357
12.14.4.4 FE2_VP6_BITRATE	357
12.14.4.5 FE2_VP6_CONCURRENCY	357
12.14.4.6 FE2_VP6_CXMODE	358
12.14.4.7 FE2_VP6_KFFREQ	359
12.14.4.8 FE2_VP6_KFINTTYPE	359
12.14.4.9 FE2_VP6_MAX_Q	359
12.14.4.10 FE2_VP6_MIN_Q	360
12.14.4.11 FE2_VP6_NOISE_REDUCTION	360
12.14.4.12 FE2_VP6_PROFILE	360
12.14.4.13 FE2_VP6_RC_MODE	360
12.14.4.14 FE2_VP6_SHARPNESS	360
12.14.4.15 FE2_VP6_STREAM_MAX_BUFFER	361
12.14.4.16 FE2_VP6_STREAM_OPTIMAL_BUFFER	361
12.14.4.17 FE2_VP6_STREAM_PEAK_BITRATE	361
12.14.4.18 FE2_VP6_STREAM_PREBUFFER	361
12.14.4.19 FE2_VP6_TEMPORAL_DOWN_WATERMARK	362
12.14.4.20 FE2_VP6_TEMPORAL_RESAMPLING	362
12.14.4.21 FE2_VP6_UNDERSHOOT_PCT	362
12.14.5 Typedef Documentation	362
12.14.5.1 vp6profile_t	362
12.14.6 Enumeration Type Documentation	363
12.14.6.1 FE2_CompressMode	363
12.14.6.2 vp6profile	363
12.15 VP6 with Alpha	364
12.15.1 Detailed Description	365

12.15.2 Define Documentation	366
12.15.2.1 FE2_CODEC_VP6ALPHA	366
12.15.2.2 FE2_VP6A_2PASS_MAX_SECTION	366
12.15.2.3 FE2_VP6A_2PASS_MIN_SECTION	366
12.15.2.4 FE2_VP6A_ALPHA_BITRATE	366
12.15.2.5 FE2_VP6A_ALPHA_MAX_Q	366
12.15.2.6 FE2_VP6A_ALPHA_MIN_Q	367
12.15.2.7 FE2_VP6A_ALPHA_NOISE_REDUCTION	367
12.15.2.8 FE2_VP6A_ALPHA_SHARPNESS	367
12.15.2.9 FE2_VP6A_BITRATE	367
12.15.2.10 FE2_VP6A_CXMODE	367
12.15.2.11 FE2_VP6A_KFFREQ	368
12.15.2.12 FE2_VP6A_KFINTTYPE	368
12.15.2.13 FE2_VP6A_MAX_Q	368
12.15.2.14 FE2_VP6A_MIN_Q	368
12.15.2.15 FE2_VP6A_NOISE_REDUCTION	369
12.15.2.16 FE2_VP6A_RC_MODE	369
12.15.2.17 FE2_VP6A_SHARPNESS	369
12.15.2.18 FE2_VP6A_STREAM_MAX_BUFFER	369
12.15.2.19 FE2_VP6A_STREAM_OPTIMAL_BUFFER	370
12.15.2.20 FE2_VP6A_STREAM_PEAK_BITRATE	370
12.15.2.21 FE2_VP6A_STREAM_PREBUFFER	370
12.15.2.22 FE2_VP6A_TEMPORAL_DOWN_WATERMARK	370
12.15.2.23 FE2_VP6A_TEMPORAL_RESAMPLING	371
12.15.2.24 FE2_VP6A_UNDERSHOOT_PCT	371
12.16 VP8	372
12.16.1 Detailed Description	374
12.16.2 Define Documentation	374
12.16.2.1 FE2_CODEC_VP8	374
12.16.2.2 FE2_VP8_2PASS_MAX_SECTION	374
12.16.2.3 FE2_VP8_2PASS_MIN_SECTION	374
12.16.2.4 FE2_VP8_ALTREF	374
12.16.2.5 FE2_VP8_AR_MAX_FRAMES	375
12.16.2.6 FE2_VP8_AR_STRENGTH	375
12.16.2.7 FE2_VP8_AR_TYPE	375
12.16.2.8 FE2_VP8_BITRATE	375

12.16.2.9 FE2_VP8_CXMODE	375
12.16.2.10 FE2_VP8_DROP_THRESH	376
12.16.2.11 FE2_VP8_KFFREQ	376
12.16.2.12 FE2_VP8_KFINTTYPE	376
12.16.2.13 FE2_VP8_LAG	376
12.16.2.14 FE2_VP8_MAX_Q	377
12.16.2.15 FE2_VP8_MB_STATIC_THRESHOLD	377
12.16.2.16 FE2_VP8_MIN_Q	377
12.16.2.17 FE2_VP8_NOISE_REDUCTION	377
12.16.2.18 FE2_VP8_OVERSHOOT_PCT	378
12.16.2.19 FE2_VP8_PROFILE	378
12.16.2.20 FE2_VP8_RC_MODE	378
12.16.2.21 FE2_VP8_SHARPNESS	378
12.16.2.22 FE2_VP8_STREAM_INITIAL_BUFFER	379
12.16.2.23 FE2_VP8_STREAM_MAX_BUFFER	379
12.16.2.24 FE2_VP8_STREAM_OPTIMAL_BUFFER	379
12.16.2.25 FE2_VP8_THREADS	379
12.16.2.26 FE2_VP8_TOKEN_PARTITIONS	379
12.16.2.27 FE2_VP8_UNDERSHOOT_PCT	380
12.17 Encoding Statistics	381
12.17.1 Function Documentation	381
12.17.1.1 encoding_status_GetAverageBitrate	381
12.17.1.2 encoding_status_GetAverageFramesize	382
12.17.1.3 encoding_status_GetElapsedTime	382
12.17.1.4 encoding_status_GetEndTime	382
12.17.1.5 encoding_status_GetMaximumFramesize	383
12.17.1.6 encoding_status_GetMinimumFramesize	383
12.17.1.7 encoding_status_GetStartTime	384
12.17.1.8 encoding_status_GetTotalFrames	384
12.17.1.9 encoding_status_PercentComplete	384
12.18 Filters	386
12.18.1 Detailed Description	386
12.19 Video Filters	387
12.20 Audio Filters	388
12.21 Deinterlace	389
12.21.1 Detailed Description	389

12.21.2 Define Documentation	390
12.21.2.1 FE2_ADAPTIVE_DEINTERLACE_MODE	390
12.21.2.2 FE2_FILTER_ADAPTIVE_DEINTERLACE	390
12.21.3 Typedef Documentation	390
12.21.3.1 deintmode_t	390
12.21.4 Enumeration Type Documentation	390
12.21.4.1 deinterlacemode	390
12.21.5 Function Documentation	390
12.21.5.1 video_options_GetDeinterlace	390
12.21.5.2 video_options_SetDeinterlace	391
12.22 Brightness/Contrast/Hue/Saturation	392
12.22.1 Detailed Description	393
12.22.2 Define Documentation	393
12.22.2.1 FE2_BCHS_BRIGHTNESS	393
12.22.2.2 FE2_BCHS_CONTRAST	394
12.22.2.3 FE2_BCHS_HUE	394
12.22.2.4 FE2_BCHS_SATURATION	394
12.22.2.5 FE2_FILTER_BCHS	394
12.22.3 Function Documentation	395
12.22.3.1 editor_options_GetBrightness	395
12.22.3.2 editor_options_GetContrast	395
12.22.3.3 editor_options_GetHue	395
12.22.3.4 editor_options_GetSaturation	396
12.22.3.5 editor_options_GetUseBrightness	396
12.22.3.6 editor_options_GetUseContrast	397
12.22.3.7 editor_options_GetUseHue	397
12.22.3.8 editor_options_GetUseSaturation	397
12.22.3.9 editor_options_SetBrightness	398
12.22.3.10 editor_options_SetContrast	398
12.22.3.11 editor_options_SetHue	399
12.22.3.12 editor_options_SetSaturation	399
12.22.3.13 editor_options_SetUseBrightness	400
12.22.3.14 editor_options_SetUseContrast	400
12.22.3.15 editor_options_SetUseHue	401
12.22.3.16 editor_options_SetUseSaturation	401
12.23 Blur	402

12.23.1 Detailed Description	402
12.23.2 Define Documentation	403
12.23.2.1 FE2_BLUR_FILTER	403
12.23.2.2 FE2_BLUR_MASKSIZE	403
12.23.2.3 FE2_FILTER_BLUR	403
12.23.3 Typedef Documentation	403
12.23.3.1 blurfilter_t	403
12.23.3.2 masksiz_t	403
12.23.4 Enumeration Type Documentation	403
12.23.4.1 blurfilter	403
12.23.4.2 masksiz	404
12.24Crop	405
12.24.1 Detailed Description	407
12.24.2 Define Documentation	407
12.24.2.1 FE2_CROP_BOTTOM	407
12.24.2.2 FE2_CROP_LEFT	407
12.24.2.3 FE2_CROP_RIGHT	407
12.24.2.4 FE2_CROP_TOP	408
12.24.2.5 FE2_FILTER_CROP	408
12.24.3 Function Documentation	408
12.24.3.1 editor_options_GetBrightness	408
12.24.3.2 editor_options_GetContrast	408
12.24.3.3 editor_options_GetCrop	409
12.24.3.4 editor_options_GetCropBounds	409
12.24.3.5 editor_options_GetCutStartTime	410
12.24.3.6 editor_options_GetCutStopTime	410
12.24.3.7 editor_options_GetHue	411
12.24.3.8 editor_options_GetSaturation	411
12.24.3.9 editor_options_GetUseBrightness	412
12.24.3.10 editor_options_GetUseContrast	412
12.24.3.11 editor_options_GetUseCut	413
12.24.3.12 editor_options_GetUseHue	413
12.24.3.13 editor_options_GetUseSaturation	414
12.24.3.14 editor_options_SetBrightness	414
12.24.3.15 editor_options_SetContrast	414
12.24.3.16 editor_options_SetCrop	415

12.24.3.17	editor_options_SetCropBounds	415
12.24.3.18	editor_options_SetCutStartTime	416
12.24.3.19	editor_options_SetCutStopTime	417
12.24.3.20	editor_options_SetHue	417
12.24.3.21	editor_options_SetSaturation	418
12.24.3.22	editor_options_SetUseBrightness	418
12.24.3.23	editor_options_SetUseContrast	419
12.24.3.24	editor_options_SetUseCut	419
12.24.3.25	editor_options_SetUseHue	420
12.24.3.26	editor_options_SetUseSaturation	420
12.25	Cut	421
12.25.1	Detailed Description	421
12.25.2	Define Documentation	422
12.25.2.1	FE2_CUT_START_SEC	422
12.25.2.2	FE2_CUT_STOP_SEC	422
12.25.2.3	FE2_CUT_USE_SEEK	422
12.25.2.4	FE2_FILTER_CUT	422
12.25.3	Function Documentation	423
12.25.3.1	editor_options_GetCutStartTime	423
12.25.3.2	editor_options_GetCutStopTime	423
12.25.3.3	editor_options_GetUseCut	424
12.25.3.4	editor_options_SetCutStartTime	424
12.25.3.5	editor_options_SetCutStopTime	425
12.25.3.6	editor_options_SetUseCut	426
12.26	Denoise	427
12.26.1	Detailed Description	427
12.26.2	Define Documentation	427
12.26.2.1	FE2_DENOISE_NOISE_LEVEL	427
12.26.2.2	FE2_FILTER_DENOISE	427
12.27	Frame Rate	428
12.27.1	Detailed Description	428
12.27.2	Define Documentation	429
12.27.2.1	FE2_FILTER_FRAMERATE	429
12.27.2.2	FE2_FRAMERATE_DECIMATE	429
12.27.2.3	FE2_FRAMERATE_FPS	429
12.27.3	Function Documentation	429

12.27.3.1 video_options_GetDecimateValue	429
12.27.3.2 video_options_GetUseSourceFramerate	430
12.27.3.3 video_options_GetVideoFramerate	430
12.27.3.4 video_options_GetVideoFramerateAsDouble	431
12.27.3.5 video_options_SetDecimateValue	431
12.27.3.6 video_options_SetUseSourceFramerate	432
12.27.3.7 video_options_SetVideoFramerate	432
12.27.3.8 video_options_SetVideoFramerateAsDouble	433
12.28Highpass	434
12.28.1 Detailed Description	434
12.28.2 Define Documentation	434
12.28.2.1 FE2_FILTER_HIGHPASS	434
12.28.2.2 FE2_HIGHPASS_CUTOFF	434
12.28.2.3 FE2_HIGHPASS_Q	434
12.29Lowpass	436
12.29.1 Detailed Description	436
12.29.2 Define Documentation	436
12.29.2.1 FE2_FILTER_LOWPASS	436
12.29.2.2 FE2_LOWPASS_CUTOFF	436
12.29.2.3 FE2_LOWPASS_Q	436
12.30Mirror	438
12.30.1 Detailed Description	438
12.30.2 Define Documentation	438
12.30.2.1 FE2_FILTER_MIRROR	438
12.30.2.2 FE2_MIRROR_HORIZONTAL	438
12.30.2.3 FE2_MIRROR_VERTICAL	439
12.31Overlay (Watermark)	440
12.31.1 Detailed Description	442
12.31.2 Define Documentation	442
12.31.2.1 FE2_FILTER_OVERLAY	442
12.31.2.2 FE2_OVERLAY_FILE	442
12.31.2.3 FE2_OVERLAY_MASK_B	443
12.31.2.4 FE2_OVERLAY_MASK_G	443
12.31.2.5 FE2_OVERLAY_MASK_R	443
12.31.2.6 FE2_OVERLAY_MASK_RGB	443
12.31.2.7 FE2_OVERLAY_MASK_X	444

12.31.2.8 FE2_OVERLAY_MASK_XY	444
12.31.2.9 FE2_OVERLAY_MASK_Y	444
12.31.2.10 FE2_OVERLAY_POS	444
12.31.2.11 FE2_OVERLAY_POS_X	445
12.31.2.12 FE2_OVERLAY_POS_Y	445
12.31.3 Enumeration Type Documentation	445
12.31.3.1 FE2_OverlayPositionMode	445
12.31.4 Function Documentation	445
12.31.4.1 overlay_options_GetMaskPixelRGB	445
12.31.4.2 overlay_options_GetMaskPixelXY	446
12.31.4.3 overlay_options_GetOverlayPath	447
12.31.4.4 overlay_options_GetOverlayPosition	447
12.31.4.5 overlay_options_GetUseOverlay	448
12.31.4.6 overlay_options_Reset	448
12.31.4.7 overlay_options_SetMaskPixelRGB	449
12.31.4.8 overlay_options_SetMaskPixelXY	450
12.31.4.9 overlay_options_SetOverlayPath	450
12.31.4.10 overlay_options_SetOverlayPosition	451
12.31.4.11 overlay_options_SetUseOverlay	452
12.32 PNG Image Export (Thumbnail)	453
12.32.1 Detailed Description	454
12.32.2 Define Documentation	455
12.32.2.1 FE2_FILTER_PNGEX	455
12.32.2.2 FE2_PNGEX_AUTO_EXPORT_COUNT	455
12.32.2.3 FE2_PNGEX_AUTO_EXPORT_END_TIME	456
12.32.2.4 FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD	456
12.32.2.5 FE2_PNGEX_AUTO_EXPORT_START_TIME	456
12.32.2.6 FE2_PNGEX_COMPRESSION_LEVEL	457
12.32.2.7 FE2_PNGEX_ENABLE_ALPHA	457
12.32.2.8 FE2_PNGEX_EXPORT_CUE_POINTS	457
12.32.2.9 FE2_PNGEX_EXPORT_FIRST_FRAME_PNG	458
12.32.2.10 FE2_PNGEX_EXPORT_INTERVAL	458
12.32.2.11 FE2_PNGEX_EXPORT_TIME_STRING	458
12.32.2.12 FE2_PNGEX_FILENAME_PREFIX	458
12.32.2.13 FE2_PNGEX_FILENAME_SUFFIX	459
12.32.2.14 FE2_PNGEX_HEIGHT	459

12.32.2.15	FE2_PNGEX_OUTPUT_DIRECTORY	459
12.32.2.16	FE2_PNGEX_WIDTH	460
12.32.3	Enumeration Type Documentation	460
12.32.3.1	FE2_PNGEXCuePtMode	460
12.33	Resample	461
12.33.1	Detailed Description	461
12.33.2	Define Documentation	461
12.33.2.1	FE2_FILTER_RESAMPLE	461
12.33.2.2	FE2_RESAMPLE_CHANNELS	461
12.33.2.3	FE2_RESAMPLE_RATE	461
12.34	Rotate	462
12.34.1	Detailed Description	462
12.34.2	Define Documentation	462
12.34.2.1	FE2_FILTER_ROTATE	462
12.34.2.2	FE2_ROTATE_ANGLE	462
12.35	Scale	463
12.35.1	Detailed Description	464
12.35.2	Reserved values	464
12.35.3	Define Documentation	465
12.35.3.1	FE2_FILTER_SCALE	465
12.35.3.2	FE2_SCALE_HEIGHT	465
12.35.3.3	FE2_SCALE_WIDTH	465
12.35.4	Function Documentation	466
12.35.4.1	video_options_GetDecimateValue	466
12.35.4.2	video_options_GetDeinterlace	466
12.35.4.3	video_options_GetImageHeight	467
12.35.4.4	video_options_GetImageWidth	467
12.35.4.5	video_options_GetUseSourceDimensions	468
12.35.4.6	video_options_GetUseSourceFramerate	468
12.35.4.7	video_options_GetVideoFramerate	469
12.35.4.8	video_options_GetVideoFramerateAsDouble	469
12.35.4.9	video_options_SetDecimateValue	470
12.35.4.10	video_options_SetDeinterlace	470
12.35.4.11	video_options_SetImageHeight	471
12.35.4.12	video_options_SetImageWidth	471
12.35.4.13	video_options_SetUseSourceDimensions	472

12.35.4.14	video_options_SetUseSourceFramerate	472
12.35.4.15	video_options_SetVideoFramerate	473
12.35.4.16	video_options_SetVideoFramerateAsDouble	473
12.36	Sharpen	475
12.36.1	Detailed Description	475
12.36.2	Define Documentation	475
12.36.2.1	FE2_FILTER_SHARPEN	475
12.37	Flix Engine API	476
12.37.1	Detailed Description	476
12.38	Main Engine Interface	477
12.38.1	Detailed Description	479
12.38.2	Enumeration Type Documentation	479
12.38.2.1	FE2_EncState	479
12.38.2.2	FE2_errno	479
12.38.2.3	FE2_ExportedVideoType	480
12.38.3	Function Documentation	480
12.38.3.1	Flix2_Copyright	480
12.38.3.2	Flix2_Create	480
12.38.3.3	Flix2_Destroy	480
12.38.3.4	Flix2_Encode	481
12.38.3.5	Flix2_Errno	481
12.38.3.6	Flix2_GetEncoderState	482
12.38.3.7	Flix2_GetExportAudio	482
12.38.3.8	Flix2_GetExportVideo	483
12.38.3.9	Flix2_GetExportVideoType	483
12.38.3.10	Flix2_GetInputFile	484
12.38.3.11	Flix2_GetOutputFile	484
12.38.3.12	Flix2_GetOverwriteExistingFiles	485
12.38.3.13	Flix2_GetSourceDuration	485
12.38.3.14	Flix2_IsEncoderRunning	485
12.38.3.15	Flix2_Reset	486
12.38.3.16	Flix2_SetExportAudio	486
12.38.3.17	Flix2_SetExportVideo	487
12.38.3.18	Flix2_SetExportVideoType	487
12.38.3.19	Flix2_SetInputFile	488
12.38.3.20	Flix2_SetOutputFile	488

12.38.3.2	Flix2_SetOverwriteExistingFiles	489
12.38.3.22	Flix2_StopEncoding	489
12.38.3.23	Flix2_Validate	489
12.38.3.24	Flix2_Version	490
12.39	Filter Manipulation	491
12.39.1	Function Documentation	491
12.39.1.1	Flix2_AddFilter	491
12.39.1.2	Flix2_FilterGetParam	492
12.39.1.3	Flix2_FilterGetParamAsStr	492
12.39.1.4	Flix2_FilterSetParam	493
12.39.1.5	Flix2_FilterSetParamAsStr	493
12.39.1.6	Flix2_RemoveFilter	494
12.40	Codec Manipulation	495
12.40.1	Function Documentation	495
12.40.1.1	Flix2_AddCodec	495
12.40.1.2	Flix2_CodecGetParam	496
12.40.1.3	Flix2_CodecGetParamAsStr	496
12.40.1.4	Flix2_CodecSetParam	497
12.40.1.5	Flix2_CodecSetParamAsStr	497
12.40.1.6	Flix2_RemoveCodec	498
12.41	Muxer Manipulation	499
12.41.1	Function Documentation	499
12.41.1.1	Flix2_AddMuxer	499
12.41.1.2	Flix2_MuxerGetParam	500
12.41.1.3	Flix2_MuxerGetParamAsStr	500
12.41.1.4	Flix2_MuxerSetParam	501
12.41.1.5	Flix2_MuxerSetParamAsStr	501
12.41.1.6	Flix2_RemoveMuxer	502
12.42	Deprecated	503
12.42.1	Function Documentation	503
12.42.1.1	editor_options_Reset	503
12.42.1.2	editor_options_Validate	503
12.43	Muxers	504
12.43.1	Detailed Description	504
12.44	FLV	505
12.44.1	Detailed Description	506

12.44.2 Supported Codecs	506
12.44.3 Format Restrictions	506
12.44.4 Define Documentation	507
12.44.4.1 FE2_FLV_CUEPT_EVENT	507
12.44.4.2 FE2_FLV_CUEPT_NAV	507
12.44.4.3 FE2_FLV_CUEPT_PARAM	507
12.44.4.4 FE2_FLV_METADATA_DISABLE	508
12.44.4.5 FE2_FLV_METADATA_ENABLE	508
12.44.4.6 FE2_MUXER_FLV	508
12.44.5 Typedef Documentation	508
12.44.5.1 flvmetadata_t	508
12.44.6 Enumeration Type Documentation	508
12.44.6.1 flv_metadata	508
12.45FXM	510
12.45.1 Detailed Description	510
12.45.2 Supported Codecs	510
12.45.3 Format Restrictions	511
12.45.4 Define Documentation	511
12.45.4.1 FE2_FXM_CUEPT_EVENT	511
12.45.4.2 FE2_FXM_CUEPT_NAV	511
12.45.4.3 FE2_FXM_CUEPT_PARAM	512
12.45.4.4 FE2_FXM_METADATA_DISABLE	512
12.45.4.5 FE2_FXM_METADATA_ENABLE	512
12.45.4.6 FE2_MUXER_FXM	512
12.45.5 Typedef Documentation	512
12.45.5.1 fxmmetadata_t	512
12.463GPP - FFmpeg	513
12.46.1 Detailed Description	513
12.46.2 Supported Codecs	513
12.46.3 Define Documentation	513
12.46.3.1 FE2_3GP_FASTSTART	513
12.46.3.2 FE2_MUXER_3GP	514
12.473GPP2 - FFmpeg	515
12.47.1 Detailed Description	515
12.47.2 Supported Codecs	515
12.47.3 Define Documentation	515

12.47.3.1 FE2_3G2_FASTSTART	515
12.47.3.2 FE2_MUXER_3G2	516
12.48MOV - FFmpeg	517
12.48.1 Detailed Description	517
12.48.2 Supported Codecs	517
12.48.3 Define Documentation	517
12.48.3.1 FE2_MOV_FASTSTART	517
12.48.3.2 FE2_MUXER_MOV	518
12.49MP4 - FFmpeg	519
12.49.1 Detailed Description	519
12.49.2 Supported Codecs	519
12.49.3 Define Documentation	519
12.49.3.1 FE2_MP4_FASTSTART	519
12.49.3.2 FE2_MUXER_MP4	520
12.50SWF	521
12.50.1 Detailed Description	525
12.50.2 Supported Codecs	526
12.50.3 Define Documentation	526
12.50.3.1 FE2_MUXER_SWF	526
12.50.3.2 FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR	526
12.50.3.3 FE2_SWF_ADD_VARIABLE	527
12.50.3.4 FE2_SWF_DELETE_VARIABLE	527
12.50.3.5 FE2_SWF_EMBEDDED_URL	527
12.50.3.6 FE2_SWF_EMBEDDED_URL_TARGET	527
12.50.3.7 FE2_SWF_EMBEDDED_URL_TYPE	528
12.50.3.8 FE2_SWF_FIXED_PRELOAD_PCT	528
12.50.3.9 FE2_SWF_FRAMERATE	528
12.50.3.10FE2_SWF_HEIGHT	528
12.50.3.11FE2_SWF_LOOP_COUNT	529
12.50.3.12FE2_SWF_ON_END_OPTION	529
12.50.3.13FE2_SWF_ON_END_URL	529
12.50.3.14FE2_SWF_ON_START_OPTION	529
12.50.3.15FE2_SWF_PRELOAD_TYPE	529
12.50.3.16FE2_SWF_START_BLANK_FRAME	530
12.50.3.17FE2_SWF_START_WAIT_SEC	530
12.50.3.18FE2_SWF_WIDTH	530

12.50.4 Enumeration Type Documentation	530
12.50.4.1 FE2_EmbeddedUrlType	530
12.50.4.2 FE2_SwfOnEndOptions	530
12.50.4.3 FE2_SwfOnStartOptions	531
12.50.4.4 FE2_SwfPreloaderOptions	531
12.50.5 Function Documentation	531
12.50.5.1 swf_options_AddVariable	531
12.50.5.2 swf_options_DeleteVariable	532
12.50.5.3 swf_options_GetAdaptivePreloaderBufferFactor	532
12.50.5.4 swf_options_GetEmbeddedUrl	533
12.50.5.5 swf_options_GetEmbeddedUrlTarget	534
12.50.5.6 swf_options_GetEmbeddedUrlType	534
12.50.5.7 swf_options_GetEnablePreloader	535
12.50.5.8 swf_options_GetInsertBlankFrameOnStart	535
12.50.5.9 swf_options_GetLoadMovieOnEndUrl	536
12.50.5.10 swf_options_GetLoopCount	537
12.50.5.11 swf_options_GetMovieOnEndOptions	537
12.50.5.12 swf_options_GetMovieOnStartOptions	538
12.50.5.13 swf_options_GetPercentToPreload	538
12.50.5.14 swf_options_GetPreloaderType	539
12.50.5.15 swf_options_GetSwfFramerate	539
12.50.5.16 swf_options_GetSwfFramerateAsDouble	540
12.50.5.17 swf_options_GetVariableCount	540
12.50.5.18 swf_options_GetWaitTimeToStart	541
12.50.5.19 swf_options_Reset	541
12.50.5.20 swf_options_SetAdaptivePreloaderBufferFactor	542
12.50.5.21 swf_options_SetEmbeddedUrl	543
12.50.5.22 swf_options_SetEmbeddedUrlTarget	543
12.50.5.23 swf_options_SetEmbeddedUrlType	544
12.50.5.24 swf_options_SetEnablePreloader	545
12.50.5.25 swf_options_SetInsertBlankFrameOnStart	545
12.50.5.26 swf_options_SetLoadMovieOnEndUrl	546
12.50.5.27 swf_options_SetLoopCount	546
12.50.5.28 swf_options_SetMovieOnEndOptions	547
12.50.5.29 swf_options_SetMovieOnStartOptions	548
12.50.5.30 swf_options_SetPercentToPreload	548

12.50.5.3	lswf_options_SetPreloaderType	549
12.50.5.32	swf_options_SetSwfFramerate	549
12.50.5.33	swf_options_SetSwfFramerateAsDouble	550
12.50.5.34	swf_options_SetWaitTimeToStart	551
12.50.5.35	swf_options_UpdateVariable	551
12.51	WebM - FFmpeg	553
12.51.1	Detailed Description	553
12.51.2	Supported Codecs	553
12.51.3	Define Documentation	553
12.51.3.1	FE2_MUXER_WEBM	553
12.52	Base Types	554
12.52.1	Define Documentation	555
12.52.1.1	INT64_MAX	555
12.52.1.2	INT64_MIN	555
12.52.1.3	ON264	555
12.52.1.4	ON2TC	555
12.52.1.5	OTC	555
12.52.1.6	PRId64	556
12.52.1.7	PRId64	556
12.52.1.8	UINT64_MAX	556
12.52.2	Typedef Documentation	556
12.52.2.1	int16_t	556
12.52.2.2	int32_t	556
12.52.2.3	int64_t	556
12.52.2.4	int8_t	556
12.52.2.5	on2bool	556
12.52.2.6	on2s16	556
12.52.2.7	on2s32	556
12.52.2.8	on2s64	556
12.52.2.9	on2s8	557
12.52.2.10	on2tc	557
12.52.2.11	on2u16	557
12.52.2.12	on2u32	557
12.52.2.13	on2u64	557
12.52.2.14	on2u8	557
12.52.2.15	uint16_t	557

12.52.2.16	uint32_t	557
12.52.2.17	uint64_t	557
12.52.2.18	uint8_t	557
12.52.3	Enumeration Type Documentation	557
12.52.3.1	_on2bool	557
12.52.3.2	on2sc	558
12.53	Video Encoding Options	559
12.53.1	Function Documentation	559
12.53.1.1	video_options_GetSourceHeight	559
12.53.1.2	video_options_GetSourceWidth	560
12.53.1.3	video_options_GetSwfHeight	560
12.53.1.4	video_options_GetSwfWidth	560
12.53.1.5	video_options_GetUseCustomSwfDimensions	561
12.53.1.6	video_options_Reset	561
12.53.1.7	video_options_SetSwfHeight	561
12.53.1.8	video_options_SetSwfWidth	562
12.53.1.9	video_options_SetUseCustomSwfDimensions	562
12.53.1.10	video_options_Validate	563
12.54	Deprecated	564
12.54.1	Enumeration Type Documentation	566
12.54.1.1	FE2_CuePointType	566
12.54.1.2	FE2_VideoCodec	566
12.54.2	Function Documentation	566
12.54.2.1	video_options_AddFLVCuePoint	566
12.54.2.2	video_options_AddFLVCuePointParameter	567
12.54.2.3	video_options_GetAlphaPercentage	568
12.54.2.4	video_options_GetCompressMode	568
12.54.2.5	video_options_GetImageQuality	569
12.54.2.6	video_options_GetKeyframeInterval	569
12.54.2.7	video_options_GetKeyframeIntervalType	569
12.54.2.8	video_options_GetMaximumBitrate	570
12.54.2.9	video_options_GetRateControlType	570
12.54.2.10	video_options_GetSwfFramerate	571
12.54.2.11	video_options_GetUseMaximumBitrate	571
12.54.2.12	video_options_GetVideoCodec	572
12.54.2.13	video_options_SetAlphaPercentage	572

12.54.2.14	video_options_SetCompressMode	573
12.54.2.15	video_options_SetImageQuality	573
12.54.2.16	video_options_SetKeyframeInterval	574
12.54.2.17	video_options_SetKeyframeIntervalType	574
12.54.2.18	video_options_SetMaximumBitrate	575
12.54.2.19	video_options_SetRateControlType	575
12.54.2.20	video_options_SetSwfFramerate	576
12.54.2.21	video_options_SetUseMaximumBitrate	576
12.54.2.22	video_options_SetVideoCodec	576
12.55	Flix Engine COM	578
12.55.1	Detailed Description	578
13	Namespace Documentation	581
13.1	flixengine_com Namespace Reference	581
14	Data Structure Documentation	583
14.1	flixengine_com::IEncodingStatus Interface Reference	583
14.1.1	Detailed Description	584
14.1.2	Member Function Documentation	584
14.1.2.1	averageBitrate	584
14.1.2.2	averageFramesize	584
14.1.2.3	elapsedTime	584
14.1.2.4	endTime	584
14.1.2.5	maximumFramesize	584
14.1.2.6	minimumFramesize	584
14.1.2.7	percentComplete	584
14.1.2.8	startTime	585
14.1.2.9	totalFrames	585
14.1.3	Property Documentation	585
14.1.3.1	sc	585
14.2	flixengine_com::IFlix Interface Reference	586
14.2.1	Detailed Description	600
14.2.2	Member Function Documentation	600
14.2.2.1	addCodec	600
14.2.2.2	addFilter	600
14.2.2.3	addMuxer	600
14.2.2.4	com_version	600

14.2.2.5	copyright	600
14.2.2.6	encode	600
14.2.2.7	encodingStatus	601
14.2.2.8	errno_	601
14.2.2.9	flxerrno	601
14.2.2.10	getEncoderState	601
14.2.2.11	getExportAudio	601
14.2.2.12	getExportVideo	601
14.2.2.13	getExportVideoType	601
14.2.2.14	getLogLevel	601
14.2.2.15	getOverwriteExistingFiles	601
14.2.2.16	getSourceDuration	601
14.2.2.17	isEncoderRunning	602
14.2.2.18	reset	602
14.2.2.19	setExportAudio	602
14.2.2.20	setExportVideo	602
14.2.2.21	setExportVideoType	602
14.2.2.22	setInputFile	602
14.2.2.23	setLogLevel	602
14.2.2.24	setLogPath	602
14.2.2.25	setOutputFile	602
14.2.2.26	setOverwriteExistingFiles	602
14.2.2.27	stopEncoding	602
14.2.2.28	swfOptions	603
14.2.2.29	syserrno	603
14.2.2.30	validate	603
14.2.2.31	version	603
14.2.2.32	videoOptions	603
14.2.3	Property Documentation	603
14.2.3.1	FE2_3G2_FASTSTART	603
14.2.3.2	FE2_3GP_FASTSTART	603
14.2.3.3	FE2_AAC_BITRATE	603
14.2.3.4	FE2_AACPLUS_BITRATE	603
14.2.3.5	FE2_AACPLUS_PARAMETRIC_STEREO	603
14.2.3.6	FE2_ADAPTIVE_DEINTERLACE_MODE	603
14.2.3.7	FE2_AMR_BITRATE	604

14.2.3.8 FE2_BCHS_BRIGHTNESS	604
14.2.3.9 FE2_BCHS_CONTRAST	604
14.2.3.10 FE2_BCHS_HUE	604
14.2.3.11 FE2_BCHS_SATURATION	604
14.2.3.12 FE2_BLUR_FILTER	604
14.2.3.13 FE2_BLUR_MASKSIZE	604
14.2.3.14 FE2_CODEC_AAC	604
14.2.3.15 FE2_CODEC_AACPLUS	604
14.2.3.16 FE2_CODEC_AMR_NB	604
14.2.3.17 FE2_CODEC_H263	604
14.2.3.18 FE2_CODEC_H263_BASELINE	605
14.2.3.19 FE2_CODEC_H264	605
14.2.3.20 FE2_CODEC_LAME	605
14.2.3.21 FE2_CODEC_PCM	605
14.2.3.22 FE2_CODEC_VORBIS	605
14.2.3.23 FE2_CODEC_VP6	605
14.2.3.24 FE2_CODEC_VP6ALPHA	605
14.2.3.25 FE2_CODEC_VP8	605
14.2.3.26 FE2_CODECPARAM_BITRATE	605
14.2.3.27 FE2_CROP_BOTTOM	605
14.2.3.28 FE2_CROP_LEFT	605
14.2.3.29 FE2_CROP_RIGHT	606
14.2.3.30 FE2_CROP_TOP	606
14.2.3.31 FE2_CUT_START_SEC	606
14.2.3.32 FE2_CUT_STOP_SEC	606
14.2.3.33 FE2_CUT_USE_SEEK	606
14.2.3.34 FE2_DENOISE_NOISE_LEVEL	606
14.2.3.35 FE2_FILTER_ADAPTIVE_DEINTERLACE	606
14.2.3.36 FE2_FILTER_BCHS	606
14.2.3.37 FE2_FILTER_BLUR	606
14.2.3.38 FE2_FILTER_CROP	606
14.2.3.39 FE2_FILTER_CUT	606
14.2.3.40 FE2_FILTER_DENOISE	607
14.2.3.41 FE2_FILTER_FRAMERATE	607
14.2.3.42 FE2_FILTER_HIGHPASS	607
14.2.3.43 FE2_FILTER_LOWPASS	607

14.2.3.44 FE2_FILTER_MIRROR	607
14.2.3.45 FE2_FILTER_OVERLAY	607
14.2.3.46 FE2_FILTER_PNGEX	607
14.2.3.47 FE2_FILTER_RESAMPLE	607
14.2.3.48 FE2_FILTER_ROTATE	607
14.2.3.49 FE2_FILTER_SCALE	607
14.2.3.50 FE2_FILTER_SHARPEN	607
14.2.3.51 FE2_FLV_CUEPT_EVENT	608
14.2.3.52 FE2_FLV_CUEPT_NAV	608
14.2.3.53 FE2_FLV_CUEPT_PARAM	608
14.2.3.54 FE2_FLV_METADATA_DISABLE	608
14.2.3.55 FE2_FLV_METADATA_ENABLE	608
14.2.3.56 FE2_FRAMERATE_DECIMATE	608
14.2.3.57 FE2_FRAMERATE_FPS	608
14.2.3.58 FE2_FXM_CUEPT_EVENT	608
14.2.3.59 FE2_FXM_CUEPT_NAV	608
14.2.3.60 FE2_FXM_CUEPT_PARAM	608
14.2.3.61 FE2_FXM_METADATA_DISABLE	608
14.2.3.62 FE2_FXM_METADATA_ENABLE	609
14.2.3.63 FE2_H263_BITRATE	609
14.2.3.64 FE2_H263_KFFREQ	609
14.2.3.65 FE2_H263_KFINTTYPE	609
14.2.3.66 FE2_H263_MAX_Q	609
14.2.3.67 FE2_H263_MIN_Q	609
14.2.3.68 FE2_H263_RC_MODE	609
14.2.3.69 FE2_H264_B_FRAME_RATE	609
14.2.3.70 FE2_H264_BITRATE	609
14.2.3.71 FE2_H264_KFFREQ	609
14.2.3.72 FE2_H264_KFINTTYPE	609
14.2.3.73 FE2_H264_PROFILE	610
14.2.3.74 FE2_H264_RC_MODE	610
14.2.3.75 FE2_H264_SPEED	610
14.2.3.76 FE2_HIGHPASS_CUTOFF	610
14.2.3.77 FE2_HIGHPASS_Q	610
14.2.3.78 FE2_ISOMEDIA_FASTSTART	610
14.2.3.79 FE2_LAME_BITRATE	610

14.2.3.80 FE2_LAME_CHANNELS	610
14.2.3.81 FE2_LAME_QUALITY	610
14.2.3.82 FE2_LAME_RC_MODE	610
14.2.3.83 FE2_LOWPASS_CUTOFF	610
14.2.3.84 FE2_LOWPASS_Q	611
14.2.3.85 FE2_MIRROR_HORIZONTAL	611
14.2.3.86 FE2_MIRROR_VERTICAL	611
14.2.3.87 FE2_MOV_FASTSTART	611
14.2.3.88 FE2_MP4_FASTSTART	611
14.2.3.89 FE2_MUXER_3G2	611
14.2.3.90 FE2_MUXER_3GP	611
14.2.3.91 FE2_MUXER_FLV	611
14.2.3.92 FE2_MUXER_FXM	611
14.2.3.93 FE2_MUXER_MOV	611
14.2.3.94 FE2_MUXER_MP4	611
14.2.3.95 FE2_MUXER_SWF	612
14.2.3.96 FE2_MUXER_WEBM	612
14.2.3.97 FE2_OVERLAY_FILE	612
14.2.3.98 FE2_OVERLAY_MASK_B	612
14.2.3.99 FE2_OVERLAY_MASK_G	612
14.2.3.100 FE2_OVERLAY_MASK_R	612
14.2.3.101 FE2_OVERLAY_MASK_RGB	612
14.2.3.102 FE2_OVERLAY_MASK_X	612
14.2.3.103 FE2_OVERLAY_MASK_XY	612
14.2.3.104 FE2_OVERLAY_MASK_Y	612
14.2.3.105 FE2_OVERLAY_POS	612
14.2.3.106 FE2_OVERLAY_POS_X	613
14.2.3.107 FE2_OVERLAY_POS_Y	613
14.2.3.108 FE2_PNGEX_AUTO_EXPORT_COUNT	613
14.2.3.109 FE2_PNGEX_AUTO_EXPORT_END_TIME	613
14.2.3.110 FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD	613
14.2.3.111 FE2_PNGEX_AUTO_EXPORT_START_TIME	613
14.2.3.112 FE2_PNGEX_COMPRESSION_LEVEL	613
14.2.3.113 FE2_PNGEX_ENABLE_ALPHA	613
14.2.3.114 FE2_PNGEX_EXPORT_CUE_POINTS	613
14.2.3.115 FE2_PNGEX_EXPORT_FIRST_FRAME_PNG	613

14.2.3.116FE2_PNGEX_EXPORT_INTERVAL	613
14.2.3.117FE2_PNGEX_EXPORT_TIME_STRING	614
14.2.3.118FE2_PNGEX_FILENAME_PREFIX	614
14.2.3.119FE2_PNGEX_FILENAME_SUFFIX	614
14.2.3.120FE2_PNGEX_HEIGHT	614
14.2.3.121FE2_PNGEX_OUTPUT_DIRECTORY	614
14.2.3.122FE2_PNGEX_WIDTH	614
14.2.3.123FE2_RESAMPLE_CHANNELS	614
14.2.3.124FE2_RESAMPLE_RATE	614
14.2.3.125FE2_ROTATE_ANGLE	614
14.2.3.126FE2_SCALE_HEIGHT	614
14.2.3.127FE2_SCALE_WIDTH	614
14.2.3.128FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR	615
14.2.3.129FE2_SWF_ADD_VARIABLE	615
14.2.3.130FE2_SWF_DELETE_VARIABLE	615
14.2.3.131FE2_SWF_EMBEDDED_URL	615
14.2.3.132FE2_SWF_EMBEDDED_URL_TARGET	615
14.2.3.133FE2_SWF_EMBEDDED_URL_TYPE	615
14.2.3.134FE2_SWF_FIXED_PRELOAD_PCT	615
14.2.3.135FE2_SWF_FRAMERATE	615
14.2.3.136FE2_SWF_HEIGHT	615
14.2.3.137FE2_SWF_LOOP_COUNT	615
14.2.3.138FE2_SWF_ON_END_OPTION	615
14.2.3.139FE2_SWF_ON_END_URL	616
14.2.3.140FE2_SWF_ON_START_OPTION	616
14.2.3.141FE2_SWF_PRELOAD_TYPE	616
14.2.3.142FE2_SWF_START_BLANK_FRAME	616
14.2.3.143FE2_SWF_START_WAIT_SEC	616
14.2.3.144FE2_SWF_WIDTH	616
14.2.3.145FE2_VCODECPARAM_KFFREQ	616
14.2.3.146FE2_VCODECPARAM_KFINTTYPE	616
14.2.3.147FE2_VCODECPARAM_RC_MODE	616
14.2.3.148FE2_VORBIS_BITRATE	616
14.2.3.149FE2_VP6_2PASS_MAX_SECTION	616
14.2.3.150FE2_VP6_2PASS_MIN_SECTION	617
14.2.3.151FE2_VP6_BITRATE	617

14.2.3.152FE2_VP6_CONCURRENCY	617
14.2.3.153FE2_VP6_CXMODE	617
14.2.3.154FE2_VP6_KFFREQ	617
14.2.3.155FE2_VP6_KFINTTYPE	617
14.2.3.156FE2_VP6_MAX_Q	617
14.2.3.157FE2_VP6_MIN_Q	617
14.2.3.158FE2_VP6_NOISE_REDUCTION	617
14.2.3.159FE2_VP6_PROFILE	617
14.2.3.160FE2_VP6_RC_MODE	617
14.2.3.161FE2_VP6_SHARPNESS	618
14.2.3.162FE2_VP6_STREAM_MAX_BUFFER	618
14.2.3.163FE2_VP6_STREAM_OPTIMAL_BUFFER	618
14.2.3.164FE2_VP6_STREAM_PEAK_BITRATE	618
14.2.3.165FE2_VP6_STREAM_PREBUFFER	618
14.2.3.166FE2_VP6_TEMPORAL_DOWN_WATERMARK	618
14.2.3.167FE2_VP6_TEMPORAL_RESAMPLING	618
14.2.3.168FE2_VP6_UNDERSHOOT_PCT	618
14.2.3.169FE2_VP6A_2PASS_MAX_SECTION	618
14.2.3.170FE2_VP6A_2PASS_MIN_SECTION	618
14.2.3.171FE2_VP6A_ALPHA_BITRATE	618
14.2.3.172FE2_VP6A_ALPHA_MAX_Q	619
14.2.3.173FE2_VP6A_ALPHA_MIN_Q	619
14.2.3.174FE2_VP6A_ALPHA_NOISE_REDUCTION	619
14.2.3.175FE2_VP6A_ALPHA_SHARPNESS	619
14.2.3.176FE2_VP6A_BITRATE	619
14.2.3.177FE2_VP6A_CXMODE	619
14.2.3.178FE2_VP6A_KFFREQ	619
14.2.3.179FE2_VP6A_KFINTTYPE	619
14.2.3.180FE2_VP6A_MAX_Q	619
14.2.3.181FE2_VP6A_MIN_Q	619
14.2.3.182FE2_VP6A_NOISE_REDUCTION	619
14.2.3.183FE2_VP6A_RC_MODE	620
14.2.3.184FE2_VP6A_SHARPNESS	620
14.2.3.185FE2_VP6A_STREAM_MAX_BUFFER	620
14.2.3.186FE2_VP6A_STREAM_OPTIMAL_BUFFER	620
14.2.3.187FE2_VP6A_STREAM_PEAK_BITRATE	620

14.2.3.188FE2_VP6A_STREAM_PREBUFFER	620
14.2.3.189FE2_VP6A_TEMPORAL_DOWN_WATERMARK	620
14.2.3.190FE2_VP6A_TEMPORAL_RESAMPLING	620
14.2.3.191FE2_VP6A_UNDERSHOOT_PCT	620
14.2.3.192FE2_VP8_2PASS_MAX_SECTION	620
14.2.3.193FE2_VP8_2PASS_MIN_SECTION	620
14.2.3.194FE2_VP8_ALTREF	621
14.2.3.195FE2_VP8_AR_MAX_FRAMES	621
14.2.3.196FE2_VP8_AR_STRENGTH	621
14.2.3.197FE2_VP8_AR_TYPE	621
14.2.3.198FE2_VP8_BITRATE	621
14.2.3.199FE2_VP8_CXMODE	621
14.2.3.200FE2_VP8_DROP_THRESH	621
14.2.3.201FE2_VP8_KFFREQ	621
14.2.3.202FE2_VP8_KFINTTYPE	621
14.2.3.203FE2_VP8_LAG	621
14.2.3.204FE2_VP8_MAX_Q	621
14.2.3.205FE2_VP8_MB_STATIC_THRESHOLD	622
14.2.3.206FE2_VP8_MIN_Q	622
14.2.3.207FE2_VP8_NOISE_REDUCTION	622
14.2.3.208FE2_VP8_OVERSHOOT_PCT	622
14.2.3.209FE2_VP8_PROFILE	622
14.2.3.210FE2_VP8_RC_MODE	622
14.2.3.211FE2_VP8_SHARPNESS	622
14.2.3.212FE2_VP8_STREAM_INITIAL_BUFFER	622
14.2.3.213FE2_VP8_STREAM_MAX_BUFFER	622
14.2.3.214FE2_VP8_STREAM_OPTIMAL_BUFFER	622
14.2.3.215FE2_VP8_THREADS	622
14.2.3.216FE2_VP8_TOKEN_PARTITIONS	623
14.2.3.217FE2_VP8_UNDERSHOOT_PCT	623
14.2.3.218sc	623
14.3 flxengine_com::IFlixPlgn Interface Reference	624
14.3.1 Detailed Description	624
14.3.2 Member Function Documentation	624
14.3.2.1 getParam	624
14.3.2.2 remove	624

14.3.2.3	setParam	625
14.3.2.4	setParamAsStr	625
14.3.3	Property Documentation	625
14.3.3.1	sc	625
14.4	flixengine_com::ISwfOptions Interface Reference	626
14.4.1	Detailed Description	628
14.4.2	Member Function Documentation	628
14.4.2.1	addVariable	628
14.4.2.2	deleteVariable	628
14.4.2.3	getAdaptivePreloaderBufferFactor	628
14.4.2.4	getEmbeddedUrlType	629
14.4.2.5	getEnablePreloader	629
14.4.2.6	getInsertBlankFrameOnStart	629
14.4.2.7	getLoopCount	629
14.4.2.8	getMovieOnEndOptions	629
14.4.2.9	getMovieOnStartOptions	630
14.4.2.10	getPercentToPreload	630
14.4.2.11	getPreloaderType	630
14.4.2.12	getSwfFramerateAsDouble	630
14.4.2.13	getVariableCount	630
14.4.2.14	getWaitTimeToStart	631
14.4.2.15	reset	631
14.4.2.16	setAdaptivePreloaderBufferFactor	631
14.4.2.17	setEmbeddedUrl	631
14.4.2.18	setEmbeddedUrlTarget	631
14.4.2.19	setEmbeddedUrlType	632
14.4.2.20	setEnablePreloader	632
14.4.2.21	setInsertBlankFrameOnStart	632
14.4.2.22	setLoadMovieOnEndUrl	632
14.4.2.23	setLoopCount	633
14.4.2.24	setMovieOnEndOptions	633
14.4.2.25	setMovieOnStartOptions	633
14.4.2.26	setPercentToPreload	633
14.4.2.27	setPreloaderType	633
14.4.2.28	setSwfFramerateAsDouble	634
14.4.2.29	setWaitTimeToStart	634

14.4.2.30	updateVariable	634
14.4.3	Property Documentation	634
14.4.3.1	sc	634
14.5	flixengine_com::IVideoOptions Interface Reference	635
14.5.1	Detailed Description	636
14.5.2	Member Function Documentation	636
14.5.2.1	addFLVCuePoint	636
14.5.2.2	addFLVCuePointParameter	636
14.5.2.3	getDecimateValue	637
14.5.2.4	getSourceHeight	637
14.5.2.5	getSourceWidth	637
14.5.2.6	getSwfHeight	637
14.5.2.7	getSwfWidth	637
14.5.2.8	getUseCustomSwfDimensions	637
14.5.2.9	getUseSourceFramerate	638
14.5.2.10	getVideoFramerateAsDouble	638
14.5.2.11	reset	638
14.5.2.12	setDecimateValue	638
14.5.2.13	setSwfHeight	638
14.5.2.14	setSwfWidth	638
14.5.2.15	setUseCustomSwfDimensions	639
14.5.2.16	setUseSourceFramerate	639
14.5.2.17	setVideoFramerateAsDouble	639
14.5.2.18	validate	639
14.5.3	Property Documentation	639
14.5.3.1	sc	639
15	File Documentation	641
15.1	about.dox File Reference	641
15.2	flixengine2/audio_options.h File Reference	642
15.2.1	Detailed Description	643
15.3	flixengine2/codec_constants.h File Reference	644
15.3.1	Detailed Description	644
15.4	flixengine2/codecs/aac.h File Reference	645
15.4.1	Detailed Description	645
15.5	flixengine2/codecs/amr.h File Reference	646
15.5.1	Detailed Description	646

15.6 flixengine2/codecs/codec_common.h File Reference	647
15.7 flixengine2/codecs/h263.h File Reference	648
15.7.1 Detailed Description	648
15.8 flixengine2/codecs/h264.h File Reference	649
15.8.1 Detailed Description	650
15.9 flixengine2/codecs/lame.h File Reference	651
15.9.1 Detailed Description	651
15.10 flixengine2/codecs/vorbis.h File Reference	652
15.10.1 Detailed Description	652
15.11 flixengine2/codecs/vp6.h File Reference	653
15.11.1 Detailed Description	654
15.12 flixengine2/codecs/vp6_alpha.h File Reference	655
15.12.1 Detailed Description	656
15.13 flixengine2/codecs/vp8.h File Reference	657
15.13.1 Detailed Description	658
15.14 flixengine2/encoding_status.h File Reference	659
15.14.1 Detailed Description	659
15.15 flixengine2/filter_constants.h File Reference	660
15.15.1 Detailed Description	660
15.16 flixengine2/filters/adaptive_deinterlace.h File Reference	661
15.16.1 Detailed Description	661
15.17 flixengine2/filters/bchs.h File Reference	662
15.17.1 Detailed Description	662
15.18 flixengine2/filters/blur.h File Reference	663
15.18.1 Detailed Description	663
15.19 flixengine2/filters/crop.h File Reference	664
15.19.1 Detailed Description	664
15.20 flixengine2/filters/cut.h File Reference	665
15.20.1 Detailed Description	665
15.21 flixengine2/filters/denoise.h File Reference	666
15.21.1 Detailed Description	666
15.22 flixengine2/filters/framerate.h File Reference	667
15.22.1 Detailed Description	667
15.23 flixengine2/filters/highpass.h File Reference	668
15.23.1 Detailed Description	668
15.24 flixengine2/filters/lowpass.h File Reference	669

15.24.1 Detailed Description	669
15.25flixengine2/filters/mirror.h File Reference	670
15.25.1 Detailed Description	670
15.26flixengine2/filters/overlay.h File Reference	671
15.26.1 Detailed Description	672
15.27flixengine2/filters/png_export.h File Reference	673
15.27.1 Detailed Description	674
15.28flixengine2/filters/resample.h File Reference	675
15.28.1 Detailed Description	675
15.29flixengine2/filters/rotate.h File Reference	676
15.29.1 Detailed Description	676
15.30flixengine2/filters/scale.h File Reference	677
15.30.1 Detailed Description	677
15.31flixengine2/filters/sharpen.h File Reference	678
15.31.1 Detailed Description	678
15.32flixengine2/flixengine2.h File Reference	679
15.32.1 Detailed Description	683
15.32.2 Define Documentation	683
15.32.2.1 FLIXENGINE_API	683
15.32.2.2 FLIXENGINE_VERSION_CHIEF	683
15.32.2.3 FLIXENGINE_VERSION_EXTRA	683
15.32.2.4 FLIXENGINE_VERSION_MAJOR	683
15.32.2.5 FLIXENGINE_VERSION_MINOR	683
15.32.2.6 FLIXENGINE_VERSION_PATCH	684
15.32.2.7 FLIXENGINE_VERSION_STR	684
15.32.3 Typedef Documentation	684
15.32.3.1 FLIX2HANDLE	684
15.32.3.2 FLIX2PLGNHANDLE	684
15.32.4 Function Documentation	684
15.32.4.1 Flix2_GetLogLevel	684
15.32.4.2 Flix2_GetLogPath	684
15.32.4.3 Flix2_SetLogLevel	685
15.32.4.4 Flix2_SetLogPath	685
15.33flixengine2/media_editor_options.h File Reference	687
15.33.1 Detailed Description	688
15.34flixengine2/muxer_constants.h File Reference	689

15.34.1 Detailed Description	689
15.35flixengine2/muxers/flv.h File Reference	690
15.35.1 Detailed Description	691
15.36flixengine2/muxers/fxm.h File Reference	692
15.36.1 Detailed Description	692
15.37flixengine2/muxers/isomedia.h File Reference	693
15.37.1 Detailed Description	693
15.37.2 Define Documentation	693
15.37.2.1 FE2_ISOMEDIA_FASTSTART	693
15.38flixengine2/muxers/swf.h File Reference	695
15.38.1 Detailed Description	696
15.39flixengine2/muxers/webm.h File Reference	697
15.39.1 Detailed Description	697
15.40flixengine2/on2types.h File Reference	698
15.40.1 Detailed Description	699
15.40.2 Define Documentation	699
15.40.2.1 DLLEXPORT	699
15.40.2.2 DLLIMPORT	699
15.40.2.3 DLLLOCAL	699
15.40.2.4 ON2API	700
15.41flixengine2/overlay_options.h File Reference	701
15.41.1 Detailed Description	701
15.42flixengine2/swf_options.h File Reference	702
15.42.1 Detailed Description	704
15.43flixengine2/video_options.h File Reference	705
15.43.1 Detailed Description	708
15.44flixengine_com.idl File Reference	709
15.44.1 Detailed Description	709
15.44.2 Function Documentation	709
15.44.2.1 warning	709
15.45mainpage.dox File Reference	710
15.46migration.dox File Reference	711
15.47samples.dox File Reference	712

Chapter 1

Flix Engine Documentation



1.1 Page Contents

- [Introduction](#)
- [Starting Points](#)
- [New Codecs & Container Formats](#)
- [Support Options & FAQ](#)
- [Upgrading to Flix Engine 8.0.6.0](#)

1.2 Introduction

Thank you for licensing the On2 Flix Engine SDK. This SDK provides a framework for embedding the acclaimed On2 Flix multimedia engine in your server-based applications. Flix Engine is able to transcode nearly any type of multimedia input to high-quality Adobe[®] Flash Video (FLV and SWF) and other popular formats, including 3GPP, 3GPP2, MOV, and MP4.

On Windows, the On2 Flix Engine provides full COM automation support for access from ASP/ASP.NET, C/C++/C#, ColdFusion, Delphi, Java (via JNI/COM Bridge), PHP, Python, Visual Basic/VB.NET, VBScript, and any other COM-enabled container application or programming language. On2 Flix Engine for Windows is available as a dynamically linked library (DLL).

Also available for Linux! On Linux, the Flix Engine provides a RPC-based solution that can be run locally on a server or across a network, and supports access from Java, PHP, Python, and Perl.

1.3 Starting Points

- Consult the [change log](#) for a complete list of improvements and fixes in this and prior Flix Engine releases.
- Reading and running the [sample code](#) gives a strong overview of what you can achieve with Flix Engine.
- [Flix Engine COM](#) reference

1.4 New Codecs & Container Formats

Flix Engine 8.0.17.0 supports the html5 ready [WebM](#) container using [VP8](#) and [Vorbis](#).

Flix Engine 8.0.10.0 now supports [MPEG-4 Part 10 standard](#), commonly known as H.264. Specifically, Flix Engine can now encode video using the [H.264](#) codec and audio using [AAC / AAC+](#), in addition to the already available encoders: [On2 VP6](#) and [H.263](#) video, and [MP3](#) and [AMR-NB](#) audio.

Accordingly, Flix Engine is now able to produce the common MPEG-4 container formats, [MP4](#), [MOV](#) and [3GPP2](#).

Another new capability of Flix Engine 8.0.10.0 is the [VP6_S encoding profile](#), which is optimized for encoding higher resolution material destined for playback in low-resource environments, such as older PCs. The standard VP6 encoding profile ([VP6_E](#)) is the default. See [FE2_VP6_PROFILE](#) in the [API](#) reference.

1.4.1 Flash Player Support and AVC Licensing

Please note carefully that, at this writing, no Flash Player release supports MPEG-4 streams. We have added these new codecs and capabilities to Flix Engine in anticipation of our users' future needs, and to expand choice.

Also note that the distribution of AVC/H.264-encoded media is not royalty-free in many cases. Be sure to comply with the H.264 licensing terms available at <http://www.mpegla.com> and <http://www.vialicensing.com>.

A [longer discussion of technical and licensing issues](#) is available at On2.com.

1.5 Support Options & FAQ

Before contacting support, please consult our [Frequently Asked Questions](#) list at On2.com, which provides fast answers to the questions that Flix Engine users ask most.

Any further questions can be directed to flixsupport@on2.com. Make sure you add *flix engine windows* to your *Subject* line. Also include the version of Flix Engine you are using, as well as a detailed description of the problem and any pertinent system information, in the message body.

1.6 Upgrading to Flix Engine 8.0.6.0

The Flix Engine API was completely re-engineered at version 8.0.6.0, and differs substantially from earlier versions. If you are upgrading from a version earlier than 8.0.6.0, please see the [Upgrade Notes](#) for more information.

The use or implementation of Flix Engine may require copyright and/or patent licenses from parties other than On2 Technologies. The user of this software is solely responsible for obtaining any and all of such licenses. In addition, supply of this Flix product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial, satellite, cable and/or other distribution channels), streaming applications (via Internet, intranets and/or other networks), other content distribution systems (pay-audio or audio-on-demand applications and the like) or on physical media (compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details regarding some of the foregoing, please visit <http://mp3licensing.com>, <http://www.mpegla.com>, <http://www.vialicensing.com> and <http://www.codingtechnologies.com>.

THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL AND NONCOMMERCIAL USE OF A CONSUMER TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL AND NONCOMMERCIAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com).

Chapter 2

About Flix Engine

2.1 Version Information

Version: 8.0.17.2

2.2 Third Parties

- This Flix product includes H.264 files licensed by Intel Corporation.
Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

- This Flix product includes aacPlus developed by Coding Technologies.

<http://www.codingtechnologies.com>

- This Flix product uses libavcodec, libavformat and libavutil which are part of FFmpeg (ffmpeg.mplayerhq.hu)

Copyright (c) 2001 Fabrice Bellard

- This Flix product uses LAME (www.mp3dev.org)

- This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)

Copyright (c) 1998-2007 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written
by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed.
If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used.
This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
"This product includes cryptographic software written by
Eric Young (eay@cryptsoft.com)"
The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
"This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or

derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

- libpng (<http://www.libpng.org/>)

libpng version 1.2.37 - June 4, 2009
Copyright (c) 1998-2009 Glenn Randers-Pehrson
(Version 0.96 Copyright (c) 1996, 1997 Andreas Dilger)
(Version 0.88 Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.)

- The PCRE library (<http://www.pcre.org/>)

Copyright (c) 1997-2006 University of Cambridge
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- zlib general compression library (<http://zlib.net/>)

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

Chapter 3

FE2_VP6_CONCURRENCY Performance

What follows are graphs of the mean encode time over 5 successive encodes for 4 relatively short videos using v8.0.9.0. In addition the same encodes were run against v8.0.8.2, less the [FE2_VP6_CONCURRENCY](#) parameter, as some modest gains were made outside the scope of this parameter. Note that exceeding the per-processor input buffer maximum (60) will reduce the overall gain as the encode process will become serialized until all frames are buffered.

The graphs are split into 2 main sections grouped by the processor on the host machine:

- [Intel Pentium 4 540](#)

1. [320x240 \(QVGA\)](#)
2. [640x480 \(VGA\)](#)
3. [1024x464](#)
4. [1280x720 \(720p\)](#)

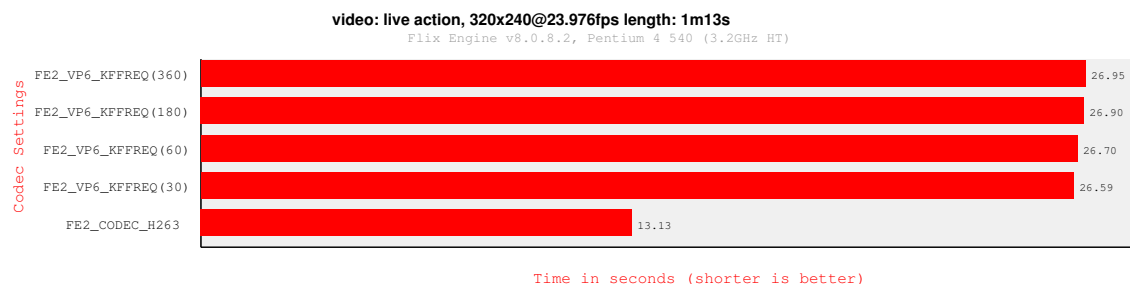
- [Intel Xeon X5355](#)

1. [320x240 \(QVGA\)](#)
2. [640x480 \(VGA\)](#)
3. [1024x464](#)
4. [1280x720 \(720p\)](#)

3.1 Intel Pentium 4 540

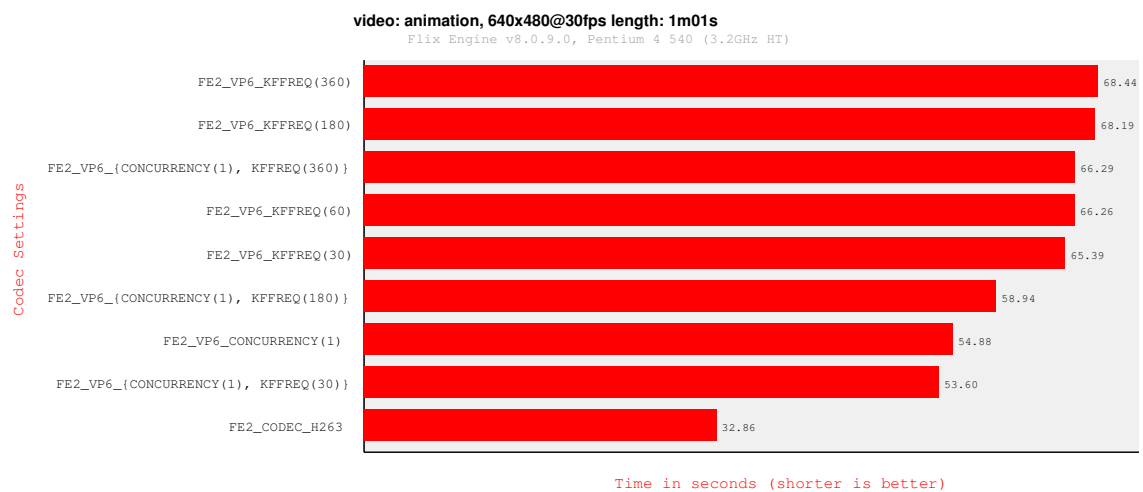
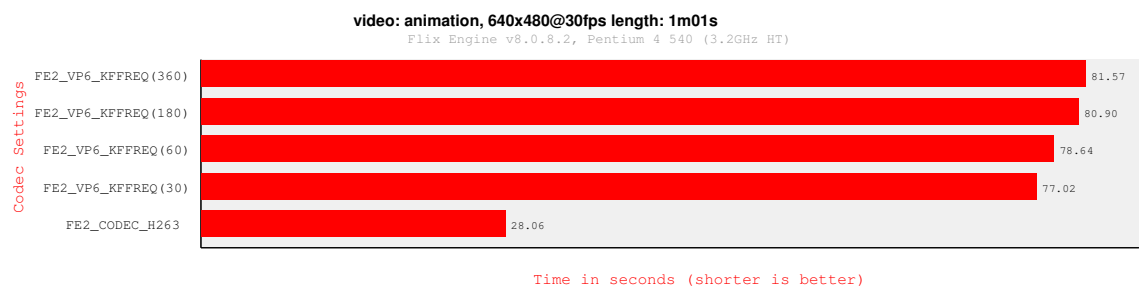
System Specifications: Slackware 10.2 (i686), Pentium 4 540 (3.2GHz HT)

3.1.1 320x240 (QVGA)

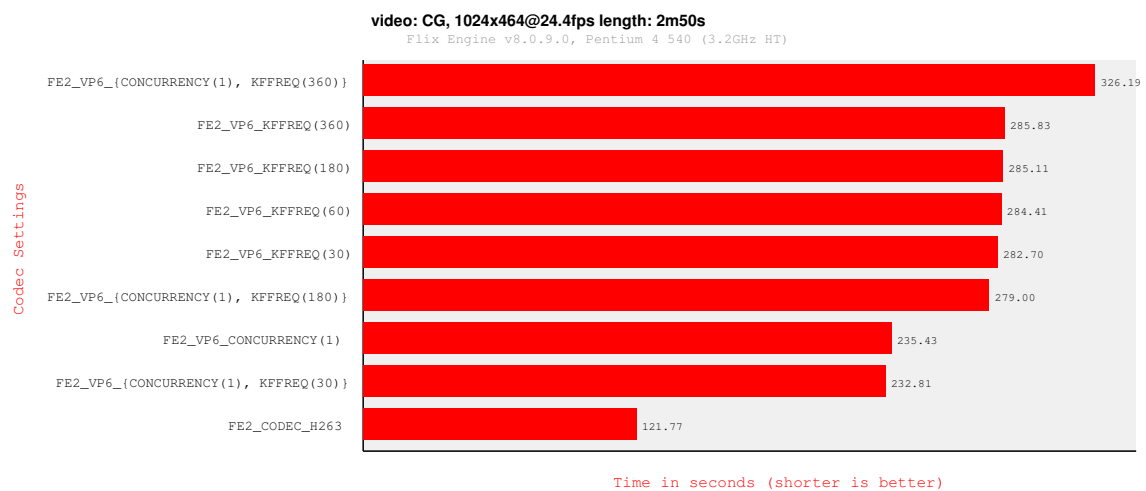
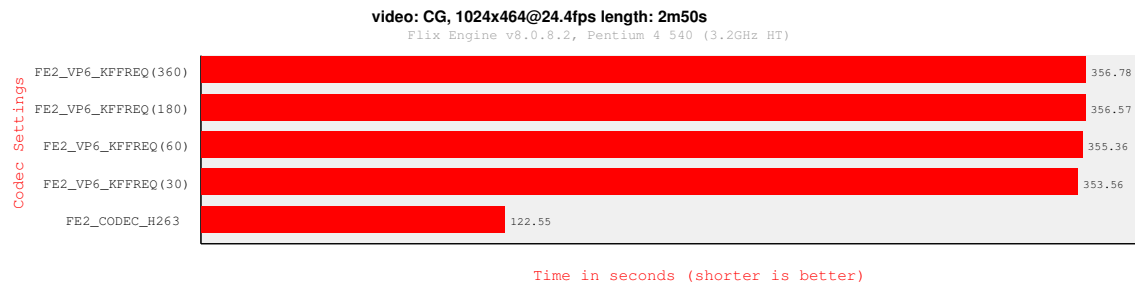




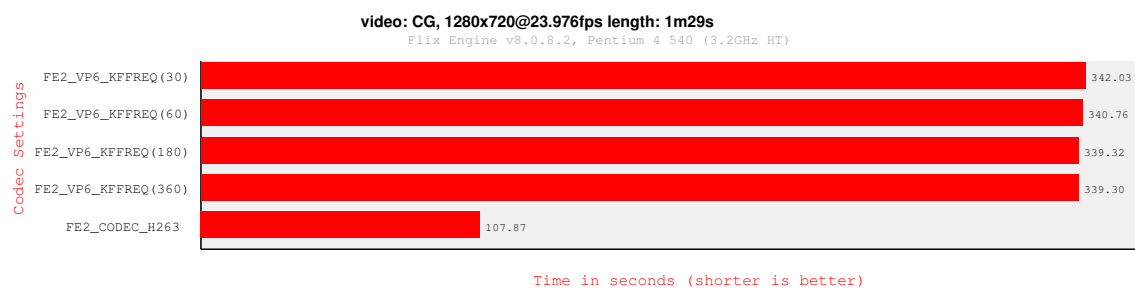
3.1.2 640x480 (VGA)

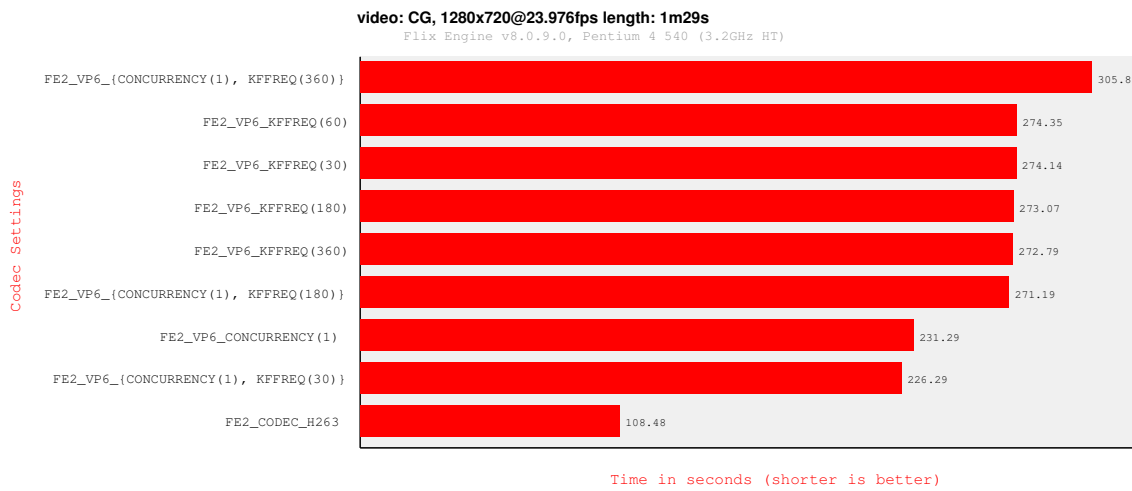


3.1.3 1024x464



3.1.4 1280x720 (720p)

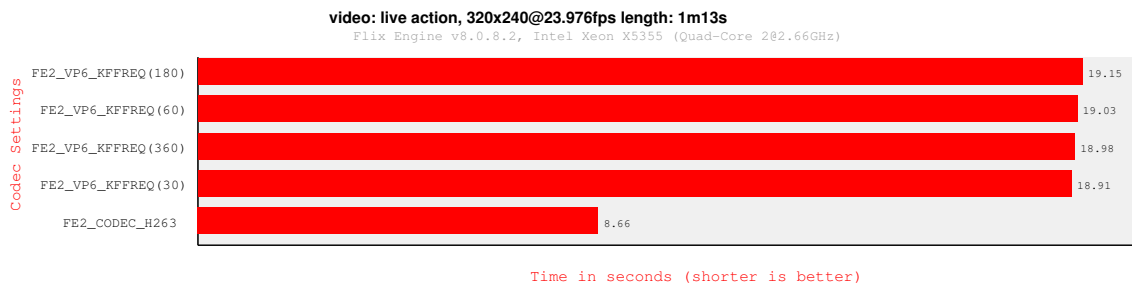


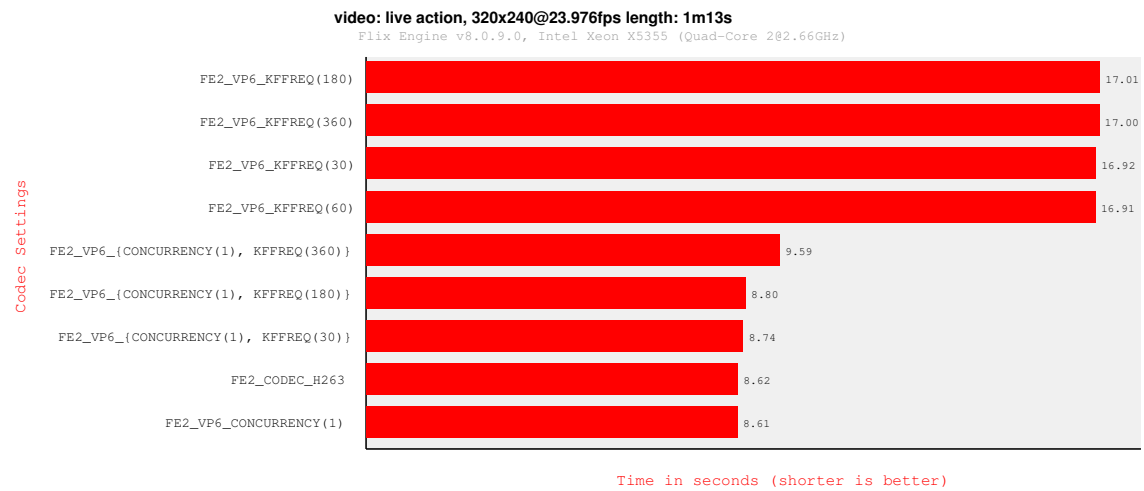


3.2 Intel Xeon X5355

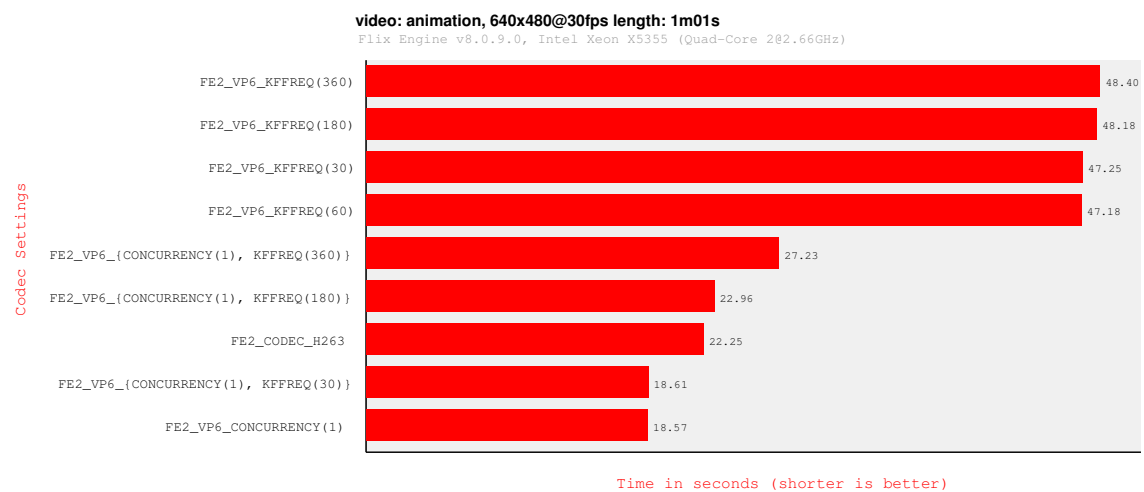
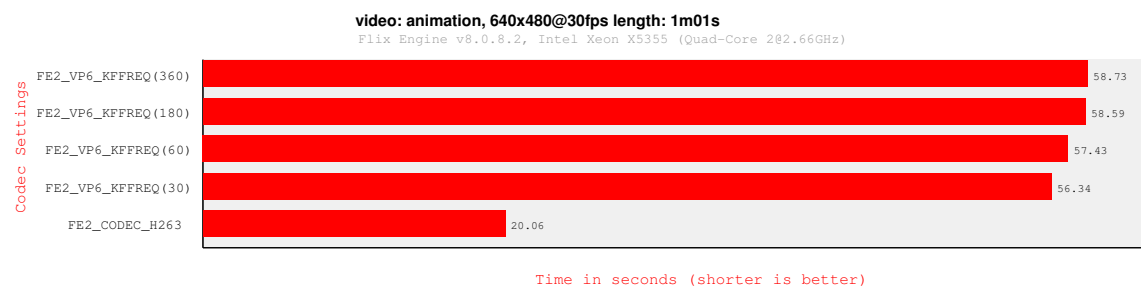
System Specifications: openSUSE 10.2 (x86_64), Intel Xeon X5355 (Quad-Core 2@2.66GHz)

3.2.1 320x240 (QVGA)

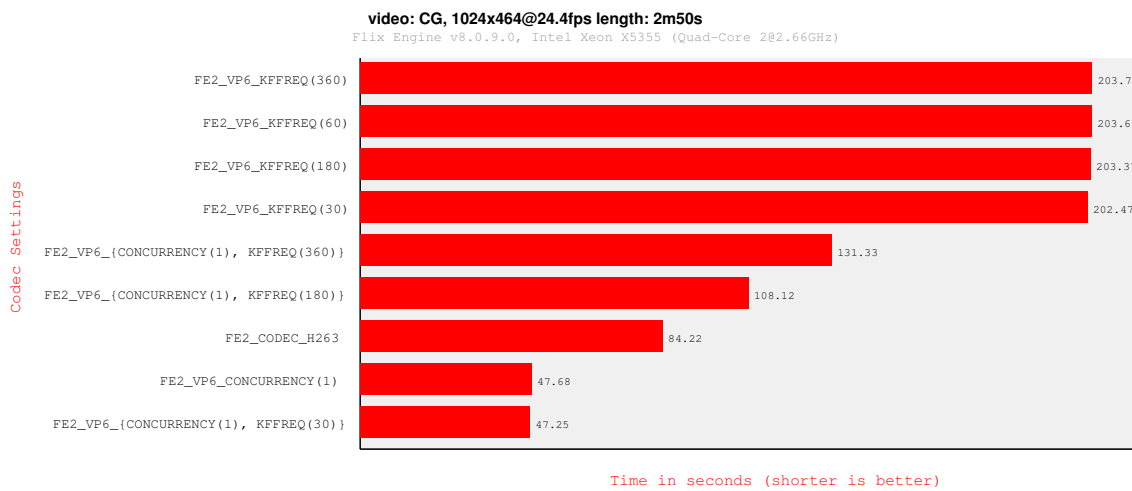
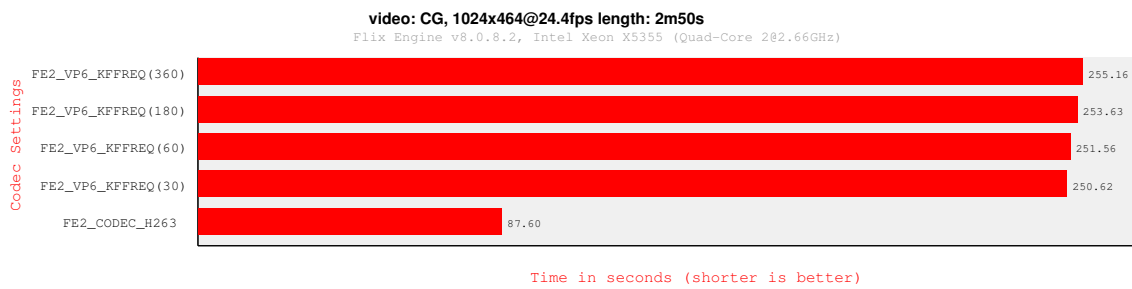




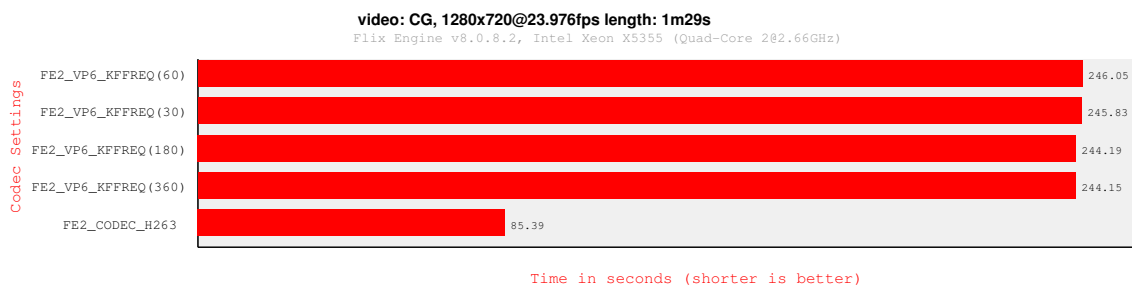
3.2.2 640x480 (VGA)

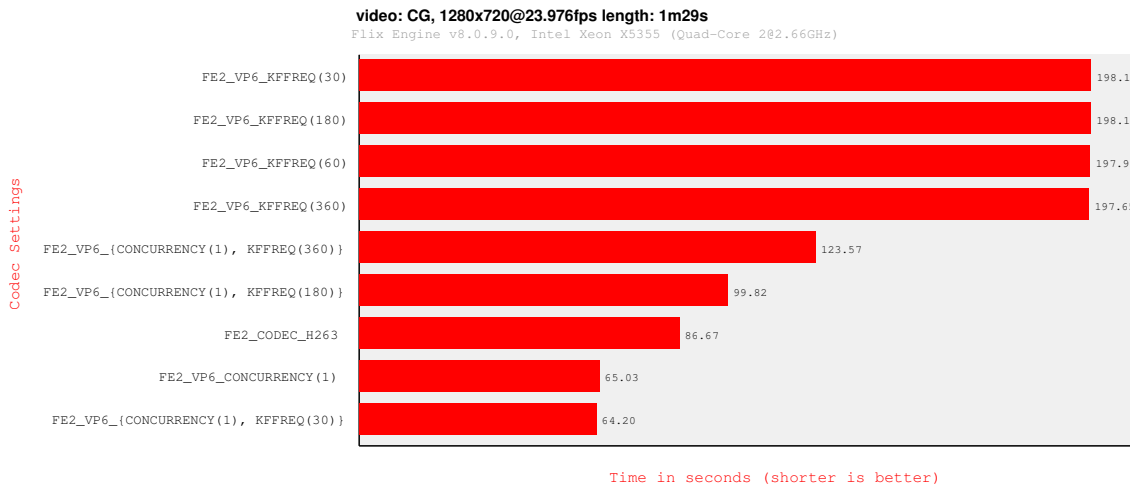


3.2.3 1024x464



3.2.4 1280x720 (720p)





Chapter 4

CHANGELOG

```

2010-11-02 v8.0.17.2 Linux
2010-11-02 v8.0.17.2_DEMO Linux
2010-11-02 v8.0.17.2 Windows COM v0.0.9.0
2010-11-02 v8.0.17.2_DEMO Windows COM v0.0.9.0
Bug Fixes:
- FE2_CODEC_VP8, update libvpx to Aylesbury
- FE2_CODEC_H264, fix a potential crash due to a bad buffer reallocation
- FE2_FILTER_FRAMERATE, fix bug causing unnecessary frame drops when using the
_FPS parameter. This would result in jitter in material even when using a
value matching a fixed-framerate input.
+-----+
2010-07-20 v8.0.17.1 Linux
2010-07-20 v8.0.17.1_DEMO Linux
2010-07-20 v8.0.17.1 Windows COM v0.0.9.0
2010-07-20 v8.0.17.1_DEMO Windows COM v0.0.9.0
Bug Fixes:
- FE2_CODEC_H264, fix bug causing grey dots/grid appearing on scene change
  to a mostly white frame when using BASE_H264PROFILE
+-----+
2010-05-17 v8.0.17.0 Linux
2010-05-17 v8.0.17.0_DEMO Linux
2010-05-17 v8.0.17.0 Windows COM v0.0.9.0
2010-05-17 v8.0.17.0_DEMO Windows COM v0.0.9.0
Features:
- (Linux) NOTE:
http://support.on2.com/flixengine/linux/api/lb\_deviation.html#update
applies to this release
- (Windows) NOTE:
http://support.on2.com/flixengine/windows/api/group\_\_flixengine\_\_com.html#warn\_du
alinterface
applies to this release
- add FE2_MUXER_WEBM
  New header file: muxers/webm.h
- add FE2_CODEC_VP8/FE2_CODEC_VORBIS
  New header files: codecs/{vp8,vorbis}.h
- (Linux) mencoder, add WebM support
  - new patches:
    o libvpxdec.diff, vp8_webm-decode_plumbing.diff
+-----+
2010-01-26 v8.0.16.1 Linux
2010-01-26 v8.0.16.1_DEMO Linux
2010-01-26 v8.0.16.1 Windows COM v0.0.8.0
2010-01-26 v8.0.16.1_DEMO Windows COM v0.0.8.0
Bug Fixes:
- FE2_FILTER_PNGEX, after v8.0.16.0, allow FE2_PNGEX_WIDTH/FE2_PNGEX_HEIGHT to
be queried
- (Linux) mencoder, update snapshot to r30380
Adds further QCELP support and the ability to transcode FLV5 (VP6 + alpha)
source.
- new patches:
  o demux_asf-flush_buffered_asf_packet.diff - flush any pending packets at
  eof, avoids audio/video truncation in certain cases
- updated patches:
  o mencoder-proto_ext_file_conf__UPSTREAM.diff - relocated common components
  to mpcommon.c and marked patch for its viability upstream based on
  comments from mplayer-dev-eng
  o mencoder_02_audio_only_hack.diff - exit immediately when no audio or
  video stream is found
  o codecs_conf-workarounds.diff - for compatibility w/prior releases revert
  r30265 and avoid untested ffwmapro
- removed patches:
  o asf-correct_movielenhth.diff, avi_check_idxflags.diff,
  demux_lavf-probe_small_files.diff, mpegvideo-revert_r18381.diff: rendered
  unnecessary
+-----+
2009-10-08 v8.0.16.0 Linux
2009-10-08 v8.0.16.0_DEMO Linux

```

2009-10-08 v8.0.16.0 Windows COM v0.0.8.0
2009-10-08 v8.0.16.0_DEMO Windows COM v0.0.8.0

Features:

- FE2_FILTER_PNGEX, FE2_PNGEX_WIDTH/FE2_PNGEX_HEIGHT allow use of reserved values in the same manner as FE2_FILTER_SCALE allowing for simpler aspect ratio preservation for thumbnails

Bug Fixes:

- (Linux) mencoder,
 - new patches:
 - o demux_lavf-probe_small_files.diff - allow files smaller than 32kB to be inspected, providing behavior similar to r27474 snapshot.
 - updated patches:
 - o libavformat_mov-avoid_reporting_empty_av_streams.diff, mov_06_tkhd_matrix_scale.diff - fix bug causing rotation of 180 to go unreported
- (Linux) flxhd, return glibc requirement to 2.3.2. glibc-2.3.4 dependencies were unintentionally added in v8.0.15.3.
- (Windows) FE2_FILTER_RESAMPLE, after v8.0.15.3, fix over reading of 8 bits/sample input creating potential for a crash
- (Windows) flvsplit.dll v2.7.0.4, fix bug causing crash in duplicate timestamp handling
- (Windows) On2QTSrc.dll v2.7.0.4, fix bug causing encode timeout/potential crash w/Apple Pro Res files

+-----+

2009-08-07 v8.0.15.3 Linux
2009-08-07 v8.0.15.3_DEMO Linux
2009-08-07 v8.0.15.3 Windows COM v0.0.8.0
2009-08-07 v8.0.15.3_DEMO Windows COM v0.0.8.0

Bug Fixes:

- Upgraded to libpng-1.2.37. Addresses an uninitialized-memory-read bug that may have security implications.

For more information, see:
<http://www.libpng.org/pub/png/libpng.html>

- Automatically add FE2_FILTER_RESAMPLE when input channels exceed 2
- (Linux) Automatically add FE2_FILTER_ROTATE for clips that specify a presentation rotation angle. Most visibly improves support for iPhone 3GS 'tallscreen' video. Original behavior may be restored by adding FE2_FILTER_ROTATE w/FE2_ROTATE_ANGLE=0.
- (Linux) mencoder, update snapshot to r29460
- oggpcm.c, leave audio frame segmentation to libogg allowing non-zero start times to be properly transferred to the output
- new patches:
 - o libavformat_mov-track_rotation_metadata.diff, demux_lavf-ID_VIDEO_ROTATION.diff - allows reporting of presentation rotation angle present in e.g., iPhone 3GS 'tallscreen' videos
 - o libavformat_mov-avoid_reporting_empty_av_streams.diff - more restricted version of removed demux_lavf-skip_empty_audio_streams.diff
 - o mencoder_l0_correct_pts.diff - mplayer derived correct-pts option used in conjunction w/muxer_ogg.
- updated patches:
 - o mov_06_tkhd_matrix_scale.diff - demux_mov analog to demux_lavf-ID_VIDEO_ROTATION.diff
- removed patches:
 - o demux_lavf-skip_empty_audio_streams.diff - Had the potential to miss detect empty audio streams w/FLV.
 - o extension-revert_r29181.diff, demux_lavf-remove_mov_preferred.diff - Favor is now given to LAVF ISO demuxer over demux_mov as it addresses many issues including problems w/various AAC/QCELP containing clips along w/some sync issues.
- (Windows) fix bug causing 1 frame video input & input for which duration could not be retrieved to fail

+-----+

2009-05-14 v8.0.15.2 Linux
2009-05-14 v8.0.15.2_DEMO Linux
2009-05-14 v8.0.15.2 Windows COM v0.0.8.0
2009-05-14 v8.0.15.2_DEMO Windows COM v0.0.8.0

Bug Fixes:

- (Linux) mencoder, update snapshot to r29308
- new patches:
 - o demux_lavf-skip_empty_audio_streams.diff - do not report empty audio streams as this affects default output stream selection
 - o extension-revert_r29181.diff - temporary, currently lavf/mov demuxer does not successfully demux all internal test clips
 - o mencoder-ext_file_conf.diff - from mplayer.c, add support for extension profiles & file specific conf files
 - o mp3lib-validate_winarray_index.diff - avoid potential crash when accessing fixed size array w/out of bounds index
 - o mpegvideo-revert_r18381.diff - temporary, changes at this revision crash select internal test clips
 - o stream_file-clear_eof_on_seek.diff - addresses a regression in the removal of mov_06_invalid_audio_size.diff. Files w/truncated audio data, but complete indexes, would result in video truncation.
 - o vd_qtvideo-validate_ImageDesc_size.diff - avoid crash should this decoder be paired w/the lavf demuxer
- updated patches:
 - o demux_lavf-add_dv_mts_preferred.diff - in addition to DV files, allow libavformat to demux MTS/M2TS files, often the container for AVCHD
 - o reduce_spurious_logging.diff - compact duplicate error messages
- removed patches:
 - o demux_increase_buffer.diff, flv-r16254_backport.diff, mplayer_demux_real.patch, workaround_libswscale_bgr15_confusion.diff: rendered unnecessary
- (Linux) Similar to v8.0.15.1 another case was identified on open that could cause a hang. This has been addressed and a timeout added to avoid further issues related to the change.
- (Windows) If the input file's aspect ratio is available on open it will now be made available for use in FE2_FILTER_SCALE. This will not address those that require a frame decode to provide this information (e.g., some WMVs).

+-----+

2009-04-16 v8.0.15.1 Linux
 2009-04-16 v8.0.15.1_DEMO Linux

Bug Fixes:

- (Linux) Fix hang introduced in change made for encode completion stability for v8.0.15.0. This would occur if mencoder exited prior to producing any data.

+-----+

2009-04-02 v8.0.15.0 Linux
 2009-04-02 v8.0.15.0_DEMO Linux
 2009-04-02 v8.0.15.0 Windows COM v0.0.8.0
 2009-04-02 v8.0.15.0_DEMO Windows COM v0.0.8.0

Features:

- (Linux) NOTE:
http://support.on2.com/flixengine/linux/api/lb_deviation.html#update
 applies to this release
- (Windows) NOTE:
http://support.on2.com/flixengine/windows/api/group__flixengine__com.html#warn_du_alinterface
 applies to this release
- FE2_MUXER_FLV/FE2_MUXER_FXM, add FE2_FLV_METADATA_ENABLE/FE2_FLV_METADATA_DISABLE and FE2_FXM_METADATA_ENABLE/FE2_FXM_METADATA_DISABLE parameters for use in controlling metadata output. Also add output support for lastkeyframetimestamp, lastkeyframelocation and the keyframes object, w/defaults being compatible w/prior releases. See the API documentation for available entries.
- FE2_CODEC_H264, add FE2_H264_SPEED parameter. Default changed from 0 to 1 improving quality on some clips which exhibited extreme degradation.
- FE2_FILTER_SCALE, add further reserved values allowing resolution to be constrained to multiples of 2,4,8,etc. See the API documentation for further details regarding their use.

Bug Fixes:

- FE2_CODEC_AMR_NB, correct output timestamp calculation avoiding potential encode failure due to memory constraints
- Upgraded to libpng-1.2.35. Addresses CVE-2009-0040.

For more information, see:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0040>

<http://www.libpng.org/pub/png/libpng.html>

- (Linux) Modify encode completion procedure improving stability for those seeing crashes at that point. Originally reported under RHEL 4.

+-----+

2009-02-04 v8.0.14.1 Linux

2009-02-04 v8.0.14.1_DEMO Linux

2009-02-04 v8.0.14.1 Windows COM v0.0.7.0

2009-02-04 v8.0.14.1_DEMO Windows COM v0.0.7.0

Bug Fixes:

- FE2_MUXER_3GP/FE2_MUXER_3G2/FE2_MUXER_MOV/FE2_MUXER_MP4, write edit list containing empty edit for streams w/non-zero start time correcting sync
- FE2_CODEC_AAC/FE2_CODEC_AACPLUS/FE2_CODEC_LAME, correct timestamp calculation. Fixes drift on long form content, esp. visible w/FE2_CODEC_AAC.
- FE2_MUXER_FLV/FE2_MUXER_MP4 and their relatives, adjust non-monotone timestamps avoiding flvcheck/potential encode failure
- (Linux) fix potential hang should mencoder exit unexpectedly during open
- (Linux) mencoder,

- new patches:

- o flv-r16254_backport.diff - bug fixes/feature additions from upstream, namely: speex and >2GB file support and correction of timestamp reporting
- o workaround_libswscale_bgr15_confusion.diff - temporary workaround, as libswscale has undergone extensive changes upstream, for issues related to codecs producing BGR15 output. These would have red and blue swapped in prior releases.

- removed patches:

- o mov_06_invalid_audio_size.diff - patch was rendered unnecessary after v8.0.13.0 and in certain cases would result in disabling otherwise valid audio tracks. Remaining mov patches renamed to reflect reduction.

+-----+

2008-12-02 v8.0.14.0 Linux

2008-12-02 v8.0.14.0_DEMO Linux

2008-12-02 v8.0.14.0 Windows COM v0.0.7.0

2008-12-02 v8.0.14.0_DEMO Windows COM v0.0.7.0

Features:

- (Linux) NOTE:

http://support.on2.com/flixengine/linux/api/lb_deviation.html#update

applies to this release

- (Windows) NOTE:

http://support.on2.com/flixengine/windows/api/group__flixengine__com.html#warn_du

applies to this release

- add FE2_MUXER_FXM

New header file: muxers/fxm.h

Bug Fixes:

- silently add FE2_FILTER_RESAMPLE if necessary to ensure FE2_CODEC_AAC/FE2_CODEC_AACPLUS configuration success, as is done w/FE2_MUXER_FLV

- (Linux) mencoder,

- updated patches

- o mov_03_moof_fragments.diff: Do not rebuild sample index when no fragments are present. This avoids large memory consumption and potential failure w/long constant sample size clips.

- (Linux) FE2_FILTER_CUT, ensure proper stream interleaving when discarding frames due to FE2_CUT_START_SEC and FE2_CUT_USE_SEEK=0. Prevents encode failure w/certain clips due to memory constraints (flixerrno=-2).

- (Windows) flvsplit.dll v1.6.6.0, fix bug causing FLV files w/PCM to fail

- (Windows) update image eos check added in v8.0.10.2 to avoid failure on 1 frame files and the addition of 1 frame w/animated GIFs

+-----+

2008-10-06 v8.0.13.0 Linux

2008-10-06 v8.0.13.0_DEMO Linux

2008-10-06 v8.0.13.0 Windows COM v0.0.6.0

2008-10-06 v8.0.13.0_DEMO Windows COM v0.0.6.0

Features:

- (Linux) NOTE:

http://support.on2.com/flixengine/linux/api/lb_deviation.html#update
applies to this release
- (Windows) NOTE:
http://support.on2.com/flixengine/windows/api/group__flixengine__com.html#warn_du
alinterface
applies to this release
- FE2_FILTER_ADAPTIVE_DEINTERLACE, add deintmode_t enum for consistency
w/other filters & codecs
- add FE2_FILTER_DENOISE
New header file: filters/denoise.h
- add FE2_FILTER_BLUR, FE2_FILTER_SHARPEN
New header file: filters/{blur,sharpen}.h
- add FE2_FILTER_MIRROR, FE2_FILTER_ROTATE
New header file: filters/{mirror,rotate}.h
- (Linux) mencoder, update snapshot to r27474
Additional differences may be obtained from the MPlayer
changelog/subversion logs ([svn://svn.mplayerhq.hu/mplayer/trunk](http://svn.mplayerhq.hu/mplayer/trunk)), aside
from overall stability and decoder/demuxer updates, notably addresses/adds:
o Nellymoser, Musepack SV8 audio decode
- new patches:
o ad_faad-restrict_aac_probe_adts.diff: restrict data probe prior to
libfaad init allowing additional AAC streams to successfully decode
o demux_lavf_02_remove_mov_preferred.diff: set legacy 3GP/3G2/MP4/MOV
demuxer (libmpdemux) to default for backward compatibility until lavf
demuxer's performance can be assessed
o reduce_spurious_logging.diff: combine vo_fonts_warn_msg_level.diff &
vd_incompatible_vo_warn_msg_level.diff, add ffmpeg logging initialization
to mencoder.c
o mplayer_demux_real.patch: addresses security advisories
oCERT-2008-013 / CVE-2008-3827, the potential for heap overflow
within the real demuxer.
For more information, see:
<http://www.ocert.org/advisories/ocert-2008-013.html>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3827>
- updated patches:
o mencoder_03_joint_av_eof.diff: combine
mencoder_03_audio_eof.diff/mencoder_09_joint_audio_video_eof.diff
o mencoder_06_asf_pts_passing.diff, mencoder_07_demux_update_pts.diff,
mencoder_08_decoded_before_dups.diff, mov_08_tkhd_matrix_scale.diff,
mov_09_increase_sample_pts_size.diff: rename based on patch reduction
o mov-ignore_reference_trak.diff: ignore tracks that reference external
files, avoiding potential decode errors caused by non-existent data
- removed patches:
o flv_02_r12184_LE_codec.diff, flv_handle_bad_header.diff,
mencoder_06_conf.diff, mov_07_missing_std.diff: rendered unnecessary
Bug Fixes:
- FE2_FILTER_CROP, do not require both right and bottom to be set, consistent
w/documentation.
- FE2_FILTER_FRAMERATE/FE2_FILTER_PNGEX, fix bug causing duration to be
applied to incorrect frame resulting in incorrect transitions. seen esp.
w/text frames.
- (Linux) mencoder-muxer_ogg.c, discard frames w/neg. timestamps. allows
transcoding of certain WMV files to mov,mp4,etc., though in some cases
sync issues may still remain.
- (Linux) mencoder-muxer_ogg.c, treat 0 byte audio frame as indication of eos
avoiding transcode failure due to ErrFileIO w/certain clips.
NOTE: in some cases this is caused by multi-trak audio (MOV), the trailing
audio will be missing as this is currently unsupported.
- (Linux) FE2_CODEC_VP6, fix chroma problem caused by input w/an odd width
- (Windows) flvsplit.dll (v1.6.2.0) expose FFDSHOW compatible Nellymoser
output pin allowing audio to be transcoded
+-----+
2008-07-16 v8.0.12.0 Linux
2008-07-16 v8.0.12.0_DEMO Linux
2008-07-16 v8.0.12.0 Windows COM v0.0.5.0
2008-07-16 v8.0.12.0_DEMO Windows COM v0.0.5.0
Features:

- FE2_MUXER_3GP/FE2_MUXER_3G2/FE2_MUXER_MOV/FE2_MUXER_MP4, add FE2_ISOMEDIA_FASTSTART parameter allowing 'moov' box placement to be influenced, enabling progressive download.
- FE2_MUXER_FLV, add support for FE2_CODEEC_AAC/FE2_CODEEC_AACPLUS. See API documentation for limitations.

Bug Fixes:

- FE2_MUXER_3GP/FE2_MUXER_3G2/FE2_MUXER_MOV/FE2_MUXER_MP4, avoid using fixed framerate as video timebase. avoids failure should observed disagree w/reported, esp. w/variable framerate input.
- FE2_CODEEC_H263/FE2_CODEEC_H263_BASELINE, should configuration of 2nd pass fail attempt a single pass avoiding outright encode failure.
- FE2_FILTER_OVERLAY, fix crash when using gray scale PNG w/bit depth < 8
- (Linux) mencoder,
 - new patches:
 - o asfhdr_correct_movielenh.diff: correct duration calculation by using double precision as well as correct units for pre-roll (ms).
 - updated files/patches:
 - o muxer_ogg.c,oggpcm.c: re-base timestamps guaranteeing one stream starting at 0. avoids unnecessary startup delays and at times A/V sync issues esp. w/ASF/WMV.
- (Linux) upgraded to SWIG-1.3.35 for language binding generation. Fixes issue seen w/PHP language bindings, from swig-1.3.35/CHANGES:
 - [PHP5] If ZTS is enabled, release <module>_globals_id in MSHUTDOWN to avoid PHP interpreter crash on shutdown. This solution was suggested here: <http://bugs.php.net/bug.php?id=40985>

Other language bindings are unaffected.

- (Linux) flixd, all components respect TMPDIR env. var. if set
- (Windows) flixengine_com.jar, resolve issue resulting in leaked interface handle as well as causing input file to remain locked should encode not be called.
- (Windows) flvdecvp6.dll (v1.4.0.0) fix incorrect -- top rather than bottom -- crop w/height non-multiple of 16
- (Windows) On2QTSrc.dll (v1.3.0.2),
 - o properly handle clips w/odd widths avoiding output stride problems. ('vertically skewed' output)
 - o allow muxed MPEG-1 movies to be rendered by traditional filters duplicating behavior of <=8.0.10.2

+-----+

2008-05-16 v8.0.11.0 Linux
 2008-05-16 v8.0.11.0_DEMO Linux
 2008-05-16 v8.0.11.0 Windows COM v0.0.4.0
 2008-05-16 v8.0.11.0_DEMO Windows COM v0.0.4.0

Features:

- FE2_CODEEC_VP6, updated encoder. provides increased quality across profiles, esp. in high motion sequences.

Bug Fixes:

- upgraded to libpng-1.2.27.
 Along w/various bug fixes addresses CVE-2008-1382.
 For more information, see:
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1382>
<http://www.libpng.org/pub/png/libpng.html>
- (Linux) mencoder,
 - new patches:
 - o mov_06_invalid_audio_size.diff - disable audio tracks w/less data than reported in track header. avoids encoding timeout (Flix2_Errno -2)
 - o mov_08_tkhd_matrix_scale.diff - set aspect ratio based on track's transformation matrix
 - o mov_09_increase_sample_pts_size.diff - avoid timestamp rollover on long material or material in which timestamps increase w/large scale.
 - updated patches:
 - o codecs_conf_custom.diff - prefer FFmpeg DV decoder over qdv for DVSD contained w/in AVI files.
 - o mencoder_08_demux_update_pts.diff - restrict change to ASF/WMV files which it is intended to fix. avoids potential sync issues w/other media.
 - o mov_01_edl_kf_search.diff
- (Windows) On2QTSrc.dll (v1.2.7.1), improved support for:
 - o multi-track video files

```

    o audio streams w/either >2 channels or samplerate >48kHz
+-----+
2008-04-17 v8.0.10.2 Linux
2008-04-17 v8.0.10.2_DEMO Linux
2008-04-17 v8.0.10.2 Windows COM v0.0.4.0
2008-04-17 v8.0.10.2_DEMO Windows COM v0.0.4.0
Bug Fixes:
- FE2_MUXER_FLV, fix bug introduced in v8.0.10.1, in to changes made to pad
  clips w/truncated audio, resulting in canSeekToEnd metadata field being set to
  0 in most cases.
- adjust default encode timeout added in v8.0.9.0 (Windows)/v8.0.10.0 (Linux)
  to 60000ms from 15000ms. helps avoid spurious failures b/w passes while under
  heavy load. the FLIX_HUNG_ENCODE_TIMEOUT env. var may be used for further
  adjustment. value is in ms, 0=no timeout.
- FE2_CODEC_VP6ALPHA, add pixel normalize to help avoid loss of transparency
  seen when re-encoding alpha content.
- FE2_CODEC_VP6, fix codec bug causing HD clips at high bitrates (>~5000KiB)
  under VBR_2PASSControl to greatly overshoot the data rate.
- correct appl. of forced FE2_FILTER_RESAMPLE due to sample rate/channel
  restrictions in output. prior versions would fail should the user specify one,
  but not the other (required) param.
- (Linux) in addition to encode timeout modifications above, modify
  mencoder std(out|err) output handling to avoid spurious timeouts caused by
  heavy logging on either.
- (Linux) libogg, check all memory allocation return values returning an error
  to caller on failure. this coupled w/FLIX_OGG_PHYMEM_PCTMAX further increases
  flixd stability w/malformed input.
- (Linux) libogg, allow audio timestamp to be forwarded to the application
- (Linux) mencoder,
- new patches:
  o asfhdr_use_best_stream.diff - selects best (highest bitrate) available
    stream for decode. fixes poor output quality in certain ASF/WMV files.
  o avi_check_idxflags.diff - better AVIINDEXENTRY flag support, fixing clip
    truncation (AVI) in certain cases.
  o codecs_conf_custom.diff - combines local codecs.conf changes to one patch
  o demux_lavf_add_dv_preferred.diff - improved DV file support
  o flv_handle_bad_header.diff - upon discovery enables stream even if header
    does not report it. addresses missing audio/video in clips of this nature.
  o flv_02_rl2184_LE_codec.diff - back port of PCM fix which correctly
    identifies endianness fixing static output
  o mp3lib_increase_maxframesize.diff - improves success rate of clips
    containing MP3 streams (avi,mp3,etc.). previous versions failing
    w/Flix2_Errno(-2,2) or, w/mp3 files, no indication at all.
  o mencoder_07_asf_pts_passing.diff, mencoder_08_demux_update_pts.diff,
    demux_asf_use_avg_fps.diff - collectively these patches address sync issues
    w/certain ASF/WMV files by forwarding the decoded timestamps.
  o mencoder_09_joint_audio_video_eof.diff - holds encoding completion until
    both streams have finished. fixes clip truncation in certain cases.
  o mencoder_10_decoded_before_dups.diff - waits to discard leading duplicate
    frames until a successful decode occurs. addresses video streams reporting
    dimensions of 0x0 in certain cases.
  o mov_07_missing_stsd.diff - skip malformed stream descriptions allowing
    transcode to complete successfully w/MOV files of this type.
- updated patches:
  o mov_03_moof_fragments.diff - changes for file format conformance,
    improving success rate of select MOV files.
- removed patches:
  o codecs_ffwmav2_ffsvq3_buggy.diff, hdlv1_use_libmpeg2.diff,
    vp6[01]_use_vfw.diff, wmv_remove_yv12.diff - replaced by
    codecs_conf_custom.diff
  o dec_audio_dmo_pts_passing.diff - replaced
    w/mencoder_08_demux_update_pts.diff
- (Linux) upgraded to libpng-1.2.23. issues referenced under Windows apply.
- (Windows) modify end of stream check w/animated image files (e.g., GIF)
  avoiding hang.
NB: current implementation ignores any loop count information stored w/in the
file and encodes supplied frames only once.

```

- (Windows) fix timeout when attempting to encode certain WMV files containing 'MSS2' video, created by Microsoft Office LiveMeeting 2005 among others.
- (Windows) on license failure, when running as a service, a message will be written to the Application Event Log. avoids hang when attempting to display a message box to a non-visible window station.

NB: the above assumes the user account the service is run as has write access to same. see: <http://support.microsoft.com/kb/842209>

- (Windows) upgraded to libpng-1.2.25

This addresses a crash bug (v1.2.21) when reading the ICC-profile chunk, iCCP (CVE-2007-5267). In addition, versions 1.2.20 and earlier have a number of potential crash-bugs due to out-of-bounds reads in certain chunk-handlers; MITRE has collectively assigned them the identifiers CVE-2007-5266, CVE-2007-5268 and CVE-2007-5269.

+-----+

2007-12-14 v8.0.10.1 Linux
 2007-12-14 v8.0.10.1_DEMO Linux
 2007-12-14 v8.0.10.1 Windows COM v0.0.4.0
 2007-12-14 v8.0.10.1_DEMO Windows COM v0.0.4.0

Bug Fixes:

- FE2_MUXER_FLV, pad clips w/truncated audio stream allowing playback to fully complete, avoiding potential hang in many third party players.
- FE2_CODEC_H264, add preliminary iPhone support. see group_codec_h264.html#h264_appledevices in the API documentation for further details.
- FE2_FILTER_PNGEX, fix regression causing lost thumbnails under 2 pass, w/clips misreporting their duration. observed duration is now used.
- FE2_FILTER_ADAPTIVE_DEINTERLACE, correct placement in filter chain fixing a case where FE2_FILTER_SCALE could precede it depending on _AddFilter order, resulting in less than optimal output.
- FE2_CODEC_AMR_NB, correct bitrate adjustment to avoid possible codec open failure due to an unsupported value, evidenced by Flix2_Errno==-7 post encode.
- (linux) mencoder, new patches:
 - o codecs_ffwmav2_ffsvq3_buggy.diff - prefer wma dmo to avoid audio corruption (static,etc.) across transitions in asf/wmv files. evident w/clips made w/Windows Movie Maker among others.
 - o dec_audio_dmo_pts_passing.diff - addresses sync issues w/various asf/wmv files.
 - o mencoder_05_mov_lavf_pts_var_fps.diff, mov_05_ffmpeg_bitrate.diff - addresses sync issues w/various mov files.
 - o mencoder_06_conf.diff - allows use of a system-wide mencoder.conf as described in the mencoder man page under 'CONFIGURATION FILES'.
- (linux) mencoder, modified mov_01_edl_kf_search.diff to use both the audio and video edit lists if available. addresses various sync issues seen w/mov files.
- (linux) mencoder, enable hdlv support
- (windows) qtsource.dll (v1.2.4.0) now preserves the alpha channel when encoding using FE2_CODEC_VP6ALPHA.
- (windows) fix some DirectShow related memory leaks
- (windows) flvsplit.dll (v1.5.5.2), give preference to sampling rate contained w/in mp3 data itself rather than relying on flv tags. addresses data timeout (ErrFileIO) w/same.

+-----+

2007-10-08 v8.0.10.0 Linux
 2007-10-08 v8.0.10.0_DEMO Linux
 2007-10-08 v8.0.10.0 Windows COM v0.0.4.0
 2007-10-08 v8.0.10.0_DEMO Windows COM v0.0.4.0

Features:

- add FE2_MUXER_3G2
 - NOTE: Only available if this feature has been added to your license. Contact sales@on2.com for further details.
- add FE2_MUXER_MOV, FE2_MUXER_MP4
 - Renamed header file: muxers/3gp.h -> muxer/isomedia.h
- add FE2_CODEC_AAC, FE2_CODEC_AACPLUS, FE2_CODEC_H264
 - New header files: codecs/{aac,h264}.h
- add FE2_VP6_PROFILE, allows selection of VP6-S in addition to (default) VP6-E. See API documentation for further details.

- add FE2_LAME_RC_MODE, allows selection of ABR/VBR in addition to legacy (default) CBR.
- (linux) update mencoder snapshot to r24143.

A full list of differences can be obtained via the subversion logs (svn://svn.mplayerhq.hu/mplayer/trunk), notably addresses:

- o AC3 setup regression introduced prior to r22906 causing loss of audio in output
- o some MJPEG related decode errors

new patches:

- o mencoder_03_audio_eof.diff - avoids truncated (video) output due to missing audio track data
- o mencoder_05_lavf_var_fps.diff - addresses sync/duration issues seen w/variable framerate flv files.
- o ffmpeg_correct_dimensions.diff - addresses cropped output in MPG/MP4/MOV files
- o mov_01_edl_kf_search.diff - updated to avoid video frame duplication (entire video track sans audio appearing after clip completion)
- o demux_mpg_short_video_fix.diff - correctly reports duration on extremely short clips (<1s)

- (windows) add ColdFusion CGI sample

Bug Fixes:

- FE2_CODEEC_VP6, avoid output artifacts w/files encoded using FE2_VP6_CONCURRENCY and FE2_VP6_TEMPORAL_RESAMPLING caused by keyframes not being produced at FE2_VP6_KFFREQ.
- attempt to close input file at encode completion. avoids potential hang when dealing w/corrupt input (often seen under windows, esp. files decoded by QuickTime). additionally, under windows, obviates forced garbage collection upon successful completion, previously necessary to guarantee timely destruction of the COM object thereby closing the input. note that should the encode fail due to a hung filter at close the open file handle will remain.
- (linux) add input timeout similar to windows (8.0.9.0) to avoid corrupt files from hanging encode thread.
- (windows) update flvdecvp6.dll, addresses decoder memory leak
- (windows) compile flxengine_com.jar w/-source 1.5 option allowing for use under older versions of Java w/o a recompile.

Note: com4j, used to generate flxengine_com.jar, utilizes 1.5 features therefore further compatibility changes are unlikely.

+-----+

2007-07-27 v8.0.9.1 Linux
 2007-07-27 v8.0.9.1_DEMO Linux
 2007-07-27 v8.0.9.1 Windows COM v0.0.3.0
 2007-07-27 v8.0.9.1_DEMO Windows COM v0.0.3.0

Bug Fixes:

- FE2_CODEEC_VP6, should initialization of FE2_VP6_CONCURRENCY=1 fail due to memory constraints, fall back to 0.
- FE2_FILTER_PNGEX, allow multiple instances as was possible w/8.0.8.2.
- FE2_FILTER_PNGEX, avoid generating (many) additional PNGs when used in conjunction w/FE2_FILTER_FRAMERATE.
- (linux) flxld, enable --interface to accept numeric IPv4 address or device name
- (windows) update qtsource.dll to v1.0.2.2, fixes out of sync transcode due to clips w/variable framerate and/or non-zero audio start time.
- (windows) fix bug causing audio decode failure in WMV files (observed under win2k3)

+-----+

2007-06-21 v8.0.9.0 Linux
 2007-06-21 v8.0.9.0_DEMO Linux
 2007-06-21 v8.0.9.0 Windows COM v0.0.3.0
 2007-06-21 v8.0.9.0_DEMO Windows COM v0.0.3.0

Features:

- FE2_CODEEC_VP6, add FE2_VP6_CONCURRENCY parameter. Allows the encode process to take advantage of multiple cores/processors yielding a potentially significant gain in performance. See the API documentation for further details.
- FE2_FILTER_SCALE, add support for reserved values [-3,0] to allow aspect ratio to be maintained. See the API documentation for further details

regarding their use.

- (windows) add Java command line sample. See the API documentation for usage notes.

Bug fixes:

- FE2_MUXER_SWF, allow audio only SWF creation.
- encoding_status_PercentComplete, avoid reporting >50% on first pass should duration be misreported.
- upgraded to libpng-1.2.18
This addresses a NULL-pointer-dereference vulnerability involving palette images with a malformed tRNS chunk (i.e., one with a bad CRC value). This vulnerability is given the identifiers VU#684664 and CVE-2007-2445 by CERT and MITRE, respectively.
For more information, see:
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2445>
- FE2_FILTER_PNGEX, should end-of-stream be encountered and not all requested PNGs have been written (due to misreported duration) output 1 final PNG.
- FE2_FILTER_PNGEX, when using FE2_PNGEX_AUTO_EXPORT_COUNT and clip duration cannot be determined output the first FE2_PNGEX_AUTO_EXPORT_COUNT frame(s), as opposed to none at all.
- (linux) mencoder, add preliminary support for movie fragments contained in file formats based on ISO-14496 Base File Format, most notably 3GPP2 (.3g2) files. addresses clip truncation.
- (linux) mencoder, add patch (avi_forceidx_fallback.diff) forcing index regeneration should the reported number of frames not match the number of index entries. addresses various duration issues for this format.
- (linux) mencoder, update muxer_ogg to allow delayed stream setup. addresses missing video in certain mpeg based streams when info is unavailable prior to decode.
- (linux) remove scale=0:0 mencoder filter as prescaling interlaced material results in output artifacts. FE2_FILTER_SCALE can now be used to achieve similar results.
- (windows) fix color convert bug causing video encode to fail (ErrEncodeV) should the source have an odd width.
- (windows) add decode timeout to avoid hung encode session due to corrupt input file. most visibly addresses hangs caused by wmv decoders supplied with windows media player 9/10, the latter being the release currently available under win2k3. Flix2_Errno will be set to -2 in this case. this change only adds a timeout for data production by DirectShow and will not correct hangs related to destroying the decode graph (currently seen with certain QuickTime files).
- (windows) FE2_FILTER_PNGEX, enable support for '/' delimited paths. fixes incorrect PNG name generation resulting in misplacement or outright failure should FE2_PNGEX_(OUTPUT_DIRECTORY|FILENAME_PREFIX) be left unset and the output filename be '/' delimited.

+-----+

2007-04-13 v8.0.8.2 Linux
2007-04-13 v8.0.8.2_DEMO Linux
2007-04-13 v8.0.8.2 Windows COM v0.0.2.0
2007-04-13 v8.0.8.2_DEMO Windows COM v0.0.2.0

Features:

- add optimized scaling code to FE2_FILTER_SCALE improving encode time 25-35% depending on the input. greatest gains are seen when scaling large input to comparatively smaller size, e.g., 720x480->256x192.
- (linux) updated mencoder snapshot to r22906.
A full list of differences can be obtained via the subversion logs (svn://svn.mplayerhq.hu/mplayer/trunk), notably addresses:
o buffer overflow in DirectShow loader.
For more information, see:
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1246>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1387>
o artifacts when transcoding flvs w/height a non-multiple of 16
o framerate issues (>actual causing fast playback) w/certain input flvs
o inability to load drvc.so and/or atrc.so.6.0 (Real binary codecs) necessary for certain Real input clips. This build now selectively statically links prerequisite libraries leaving base system libraries as dynamic. requirements are similar to flixd so should not affect the binary's ability to load. should this become an issue and a fully static binary is

required, the sources are available via: <http://www.on2.com/gpl/mplayer>

Bug fixes:

- fix crash in FE2_FILTER_RESAMPLE caused by the demuxer producing 0 byte audio frames. this was only observed under windows but had the potential of causing a crash on either platform.
- create key frame when navigation cue point(s) are set with FE2_FLV_CUEPT_NAV
- fix encoder hang when specifying an explicit framerate w/FE2_FILTER_FRAMERATE. in addition to the fix made in v8.0.8.1 there was another case when using either FE2_FILTER_ADAPTIVE_DEINTERLACE or FE2_FILTER_CROP, or should the input frame need to be extended or color converted (windows only), that could cause a hang to occur.
- (linux) revert LAME library to 3.96.1. w/certain input files 3.97 currently produces encode errors.
- (linux,mencoder) allow mean framerate patch to only increase output framerate. avoids artifacts when decoding certain h264 clips (mov, etc).
- (windows) update flvsplit.dll to v1.5.3.0, ignores flv header information indicating presence of audio, manually inspecting the data, allowing for audio extraction from malformed files.
- (windows) update qtsource.dll to v1.0.2.0, addresses incorrect framerate calculation causing file to be transcoded at low framerate.
- (windows) forward base timestamp from the demuxer through the codec. allows for material w/non-zero start times to remain in sync after transcode.
- (windows) fix color conversion bug producing green bar in videos having an input width a non-multiple of 4.

+-----+

2007-03-09 v8.0.8.1 Windows COM v0.0.2.0
 2007-03-09 v8.0.8.1_DEMO Windows COM v0.0.2.0
 2007-03-09 v8.0.8.1 Linux
 2007-03-09 v8.0.8.1_DEMO Linux

Bug fixes:

- fix crash when removing unnecessary (i.e., desired==input) FE2_FILTER_RESAMPLE
- fix encoder hang when specifying an explicit framerate > source with FE2_FILTER_FRAMERATE
- (windows) fix bug causing loss of audio at input end. amt. varied based on the clip w/as much as 8s loss possible.
- (linux) updated 64bit libflxengine2.so to allow for use on platforms whose binutils and/or dynamic linker/loader do not have .gnu.hash support, as evidenced by output similar to:
 '/usr/local/lib64/libflxengine2.so: file not recognized: File format not recognized'

+-----+

2007-02-26 v8.0.8.0 Windows COM v0.0.2.0
 2007-02-26 v8.0.8.0_DEMO Windows COM v0.0.2.0
 2007-02-26 v8.0.8.0 Linux
 2007-02-26 v8.0.8.0_DEMO Linux

Features:

- upgrade to lame-3.97
- add FE2_LAME_QUALITY parameter to FE2_CODEC_LAME
- added muxer interface
 - New API calls: Flix2_MuxerGetParam/AsStr, Flix2_MuxerSetParam/AsStr
 Flix2_AddMuxer/Flix2_RemoveMuxer
 - New header files: muxer_constants.h, muxers/3gp.h,
 muxers/flv.h, muxers/swf.h
 - NOTE: This update has deprecated many of the muxer related functions in (swf|video)_options.h. Consult the deprecated list in the API documentation as they will be removed in a future release.
- add FE2_MUXER_3GP. See the API documentation for limitations.
 - NOTE: Only available if this feature has been added to your license. Contact sales@on2.com for further details.
- add FE2_CODEC_H263_BASELINE for use in conjunction with the 3gpp muxer.
- add FE2_CODEC_AMR_NB for use in conjunction with the 3gpp muxer.
- (windows) add ASP, ASP.NET and VB(6|.NET) samples
- (linux) add libtheora-1.0alpha7 (<http://www.theora.org/>) to mencoder build
- (linux) add support for FLIX_OGG_PHYMEM_PCTMAX environment variable used to control individual ogg demuxer's max memory usage to avoid a segmentation fault due to unchecked memory allocations within

libogg. The default set in flixd's start script is 70. See the flixd man page for further details.

Bug fixes:

- modify FE2_FILTER_RESAMPLE to use FFmpeg based filter. addresses sync issues when resampling longer content (esp. non-integer conversions, e.g., 48->44.1kHz) as well as the issue of giving precedence to the left channel when converting stereo to mono.
- fix bug causing audio and video to be improperly interleaved in certain cases, mostly when dealing with clips that were variable framerate (wmv). a video frame with a timestamp much greater than the current audio would be inserted followed by a large amount of (earlier timestamped) audio causing playback and FMS issues.
- fix bug in FE2_CODEC_LAME causing the frame prior to the final block to be duplicated. depending on the content this may have produced an audible distortion in the output. in addition, break final mp3 block into individual frames to avoid storing one (potentially large) block at the end of the file.
- fix bug, most evident in the demo version, that could cause a crash should the crop filter be used to adjust the width of the input.
- remove video height restriction from overlay filter. prior versions required the height to be divisible by 2.
- (windows) On2QTSOURCE.dll,v1.0.1.3
 - o updated to handle the mov transformation matrix allowing for the correct output size/viewport.
 - o avoid potential hang on shutdown, seen most often when using FE2_CUT_STOP_SEC or explicitly disabling stream.
- (windows) add support for 8bit audio input (converted to 16 as linux)
- (windows) fix bug related to shared 1st pass file w/VP6 2pass resulting in incorrect output when running concurrent encodes
- (windows) fix bug causing the alpha channel to appear inverted when using CODEC_VP6ALPHA.
- (linux) avoid flixd hang due to unconstrained mencoder error output on open

+-----+

2006-12-13 v8.0.7.1 Windows COM v0.0.1.0

2006-12-13 v8.0.7.1_DEMO Windows COM v0.0.1.0

2006-12-13 v8.0.7.1_DEMO Linux

2006-12-13 v8.0.7.1 Linux

- fix bug in cut filter when setting FE2_CUT_STOP_SEC and using FE2_CUT_USE_SEEK along with a non-zero FE2_CUT_START_SEC causing the output to have an incorrect duration. The stop time is now adjusted accordingly should the seek succeed.
- (linux) add 64bit version of libflixengine2.so, the client-side rpc lib, to allow for 64bit language bindings to be built.
NOTE: Support has yet to be added to the installer.
libflixengine2.so must be extracted and installed by hand.
From a running installer the file is found under:
\$HOME/flix-engine-installer.<pid>/.flix-engine-installation-files/test
ing/lib64
Install the desired language binding sources but 'skip' the build during install to avoid errors.
Tested under FC6 running on an amd64x2. As this is currently under development, support will not be provided for installing this library.
- fix bug in resample filter (upsampling) causing clicks/pops in the output. rather than out ranging, in this case the problem stemmed from over reading.
- (windows) add FE2_PNGEX_(WIDTH|HEIGHT) to the COM interface
- upgraded to libpng-1.2.14. This fixes a bug where a specially crafted PNG file could cause a crash. For more information, see:
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5793>

+-----+

2006-11-30 v8.0.7.0 Windows COM v0.0.0.1

2006-11-30 v8.0.7.0_DEMO Windows COM v0.0.0.1

- (windows) correct errant reference count within On2QTSOURCE.dll (QuickTime source filter) allowing the filter to be properly unloaded, releasing references to the input file

2006-11-29 v8.0.7.0_DEMO Linux

2006-11-29 v8.0.7.0 Linux

- (linux) add mencoder patch to calculate mean framerate w/in mov and related

```

        files. Addresses issue w/3gp files reporting incorrect (and often
        low) values.
- add FE2_PNGEX_(WIDTH|HEIGHT) parameters to PNG exporter allowing the image
  to be scaled
- add FE2_PNGEX_COMPRESSION_LEVEL parameter to PNG exporter
- add support for frame rate filter, FE2_FILTER_FRAMERATE, via the filter api.
  New header file: filters/framerate.h
  NOTE: This has deprecated:
      video_options_(G|S)et(VideoFramerateAsDouble|
                           DecimateValue|UseSourceFramerate)
- (windows) fix object creation to allow for instances of IFlix to be created
  across multiple threads. close references to flixengine.lic when
  they are unneeded, addresses file descriptor leak.
- correct frame rate calculated by frame rate filter when using decimate.
  previous versions would correctly decimate the input, but report the source
  frame rate, adversely affecting video quality.
- (windows) correct PNG export filter to accept '\\' as the path name
  separator. fixes a potential crash when either
      FE2_PNGEX_DIRECTORY or FE2_PNGEX_FILENAME_PREFIX were left unset.
- add FE2_CUT_USE_SEEK parameter. If set and FE2_CUT_START_SEC is non-zero
  attempts to seek the input instead of decoding/discarding all leading data,
  improving processing time. This is now the default. Set to 0 for legacy
  behavior.
- fix bug in resample filter causing 'clicks'/'pops' in the output due to
  out ranging of the samples
- fix bug in resample filter causing distorted output
  when the number of input and output channels differed
+-----+
2006-10-24 v8.0.6.0_DEMO Windows
2006-10-20 v8.0.6.0_DEMO Linux
2006-10-20 v8.0.6.0 Linux
- (linux) mencoder
  o snapshot updated to r20146, adds (most notably):
    o VP6 flv import
    o QCELP support
  o add amr_[nw]b support
  o patch to support mov files w/audio samplerate of 5512Hz
    (produced by various cellphones)
  o correct mencoder identify patch to provide more correct clip info
    if available. addresses incorrect/unknown stream duration being
    reported with some mpeg files.
- (linux) when installing java bindings, symlink javadoc to the doc install
  directory, default: /usr/local/share/doc/on2
- (linux) sync ffmpeg snapshot to that used under windows (r5990)
+-----+
2006-09-29 v8.0.6.0 Windows COM v0.0.0.0
- fix possible encoder crash when transcoding low framerate clips
- upgraded to libpng 1.2.12
- fix bad chroma in png's generated from input of odd width/height
- exposed FE2_CODEC_PCM to complete audio_options_SetFlvAudioFormat deprecation
- add support for brightness/contrast/hue/saturation filter, FE2_FILTER_BCHS,
  via the filter api
  New header file: filters/bchs.h
  NOTE: This has deprecated:
      editor_options_(G|S)et(Brightness|Contrast|Hue|Saturation) and
      editor_options_(G|S)etUse(Brightness|Contrast|Hue|Saturation)
- add resample filter, FE2_FILTER_RESAMPLE
  New header file: filters/resample.h
  NOTE: This has deprecated: audio_options_(G|S)et(Samplingrate|Stereo)
- fix misleading 100 returned from encoding_status_PercentComplete before
  encoding starts
- fix small memory leak when using FE2_PNGEX_EXPORT_TIME_STRING
- (windows) initial release of COM interface

```

Chapter 5

Upgrade Notes

This page pertains only to Flix Engine users upgrading from a version earlier than 8.0.6.0.

5.1 Page Contents

- [Overview](#)
- [Important: COM ID](#)
- [Flix Engine Improvements](#)
 - [Exception Handling](#)
 - [PNG Thumbnails](#)
 - [Concurrent Encodes](#)
 - [Sane Naming Conventions](#)
 - [New COM Interface is Deprecation-Aware](#)
 - [Audio Decoding Doesn't Require Audio Hardware](#)
- [API Reference Shortcuts](#)
- [Functions with No Equivalent in the New Flix Engine](#)

5.2 Overview

Version 8.0.6.0 represents the first release of a new, next-generation Flix Engine. The new Flix Engine **is not backwards-compatible with 8.0.0.x and earlier versions**. Earlier versions descended from a purchased, third-party codebase, whose purpose included desktop application development.

Flix Engine Version No.	COM Object Version No. (see the Change Log)	Codebase	Product	Release Date
<= 8.0.0.x		WildForm	Flix Engine	
8.0.6.0	0.0.0.0	On2 Technologies	new Flix Engine	2006-09-29
8.0.7.0	0.0.0.1	On2 Technologies	new Flix Engine	2006-11-30
...

The new Flix Engine is licensed as a server-based, enterprise-grade digital multimedia transcoding engine, with special emphasis on Flash Video creation via On2's advanced VP6 video encoder. It *may not* be compiled or linked into desktop applications.

The new Flix Engine API obsoletes or deprecates functionality geared toward desktop application development, and makes **significant improvements in performance, stability, and usability**.

The best approach to migration is to re-code to the new API. Depending on how you've implemented Flix Engine in the past, it may be realistic to modify your existing routines *once you're familiar with the new Flix Engine*.

5.3 Important: COM ID

COM Program ID (ProgID) has changed to On2.FlixEngine

5.4 Flix Engine Improvements

5.4.1 Exception Handling

The legacy Flix Engine exposed few methods for catching and handling errors, and had a bad habit of silently failing. The new Engine returns an improved range of error codes and has improved logging and debugging hooks. We also provide a new function for returning the current error state of the engine (`flixfengine_com::IFlix::errno_()`).

5.4.2 PNG Thumbnails

The legacy Flix Engine could produce thumbnail images of the first video frame only, in JPEG format. Unfortunately, JPEG is a lossy format, and transcoding to other formats from JPEG may incur further loss. Also, the first frame in a video is rarely a good representation of the content.

Thumbnail images are now produced in PNG (Portable Network Graphics) format. PNG employs lossless compression, supports alpha channel compositing, and is easily transcoded to other formats. Users may now specify which frames to thumbnail.

5.4.3 Concurrent Encodes

The new Flix Engine has stable support for concurrent encoding. Some users may still prefer to queue encoding jobs via their own routines, but the convenience of concurrent encoding is now available.

5.4.4 Sane Naming Conventions

We have refactored the new Flix Engine's internal naming conventions, to restore some logic and remove cruft.

5.4.5 New COM Interface is Deprecation-Aware

Some features of 8.0.0.x are obsolete under the new Flix Engine ([see below](#)), others are deprecated and will be removed in a future version. The new Engine's COM interface exposes none of the deprecated functions, so re-coding to the new Engine protects your work.

5.4.6 Audio Decoding Doesn't Require Audio Hardware

The legacy Flix Engine was sometimes unable to decode audio on systems without a hardware audio device. That limitation has been corrected in the new Flix Engine.

5.5 API Reference Shortcuts

The Flix Engine API Reference is presented in the familiar and thorough [Doxygen](#) HTML format. Below are links to documentation pages of particular interest to upgraders.

- [COM Interface Front Page](#)

Flix Engine for Windows is exposed solely via its COM interface. Function-level detail is included for reference purposes; obviously, functions may not be called directly.

- [IFlixPlgn Interface](#)

Access Codec and Filter functions via the [flixengine_com::IFlixPlgn](#) Interface. *Note: Please be aware that, while called similarly, filters and codecs are separate and discrete.*

- [Complete Function Listing](#) (alphabetical)

Note: This listing includes deprecated functions, which will be removed from the API in a future release and are not exposed by the COM interface. Each deprecated function is clearly marked, and cites a replacement method. Also: [Deprecated Function Listing](#)

- [VP6 Video Encoding Settings](#)
- [Video Framerate Filter](#)
- [Video Scale \(Resize\) Filter](#)
- [MP3 Audio Encoding Settings \(LAME\)](#)
- [Audio Resample Filter](#)
- [Overlay \(Watermark\) Video Filter](#)
- [PNG Image Export \(Thumbnail\) Filter](#)

5.6 Functions with No Equivalent in the New Flix Engine

In re-engineering the API, we identified groups of functions in the legacy Flix Engine API that were artifacts, and not germane to Flix Engine's purpose of developing server-based Flash Video encoding applications. Listed below are legacy functions that have no equivalent in the new Flix Engine.

IWF_FlixCOMEncoderEvents Interface *OnEncodingFinished, OnEncodingMessage, OnChange, OnEncoderProgress*

Flix Engine no longer fires these events, as they exceed the core API's intended scope. Similar functionality may be added in a future release. Developers may easily replicate this functionality with custom routines.

IWF_FlixCOMEncoder Interface *EnableDirectShow8Importer, DisableDirectShow8Importer, EnableQuicktimeImporter, DisableQuicktimeImporter, EnableQTMLInitialization, DisableQTMLInitialization, LoadSettings*

Flix Engine may no longer be compiled into desktop applications for redistribution, thus the above functions are unneeded and have not been replicated in the new Flix Engine. Functionality similar to *LoadSettings* may be added in a future release.

WF_EncodingStatus Interface *GetPlayerRamUsage, GetAverageQuality, GetMaximumQuality, GetMinimumQuality*

The above encoding status functions have not been replicated in the new Flix Engine. In our judgment they did not return useful data.

WF_Exports Interface *GetOutputEmailPath, SetOutputEmailPath, GetOutputHtmlPath, SetOutputHtmlPath, GetOutputMacProjectorPath, SetOutputMacProjectorPath, GetOutputMp3Path, SetOutputMp3Path, GetOutputWavPath, SetOutputWavPath, GetOutputWinProjectorPath, SetOutputWinProjectorPath, GetOutputHtmlPath, SetOutputHtmlPath, GetExportHtml, SetExportHtml,*

GetOutputMacProjectorPath, SetOutputMacProjectorPath, GetExportMacProjector, SetExportMacProjector, GetOutputWinProjectorPath, SetOutputWinProjectorPath, GetExportWinProjector, SetExportWinProjector

The above functions were artifacts of Flix Engine's history as a desktop application development tool, and have not been replicated in the new Flix Engine.

WF_PlayerMaker Interface *GetLoadMovieUrlForExisting, SetLoadMovieUrlForExisting, GetUseLoadMovieForExisting, SetUseLoadMovieForExisting, GetOutputSwfPathForExisting, SetOutputSwfPathForExisting, SetNewFilePrefix, AutoExport, IsAutoExportSet, GetLoadMovieUrlForNew, SetLoadMovieUrlForNew, GetUseLoadMovieForNew, SetUseLoadMovieForNew, GetOutputSwfPathForNew, SetOutputSwfPathForNew, GetCustomBackgroundColor, SetCustomBackgroundColor, GetUseCustomBgColor, SetUseCustomBgColor, GetSkinFile, SetSkinFile*

We determined that the above player creation functions were not being used by Flix Engine customers, because players constructed this way look out-dated and unappealing. These functions have not been replicated in the new Flix Engine.

WF_Vectorize Interface *GetColorOptions, SetColorOptions, GetMonochromeColor, SetMonochromeColor, GetColorThreshold, SetColorThreshold, GetDontFillShapes, SetDontFillShapes, AddOption, DeleteOption, GetCurrentOptions, GetBlurThreshold, SetBlurThreshold, GetDetail, SetDetail, GetFitCurvesThreshold, SetFitCurvesThreshold, GetFitLinesThreshold, SetFitLinesThreshold, GetPeakReductionThreshold, SetPeakReductionThreshold, GetStragglersThreshold, SetStragglersThreshold, GetRemoveJaggedEdgesThreshold, SetRemoveJaggedEdgesThreshold, GetRemoveRightAnglesThreshold, SetRemoveRightAnglesThreshold, GetMultiplePasses, SetMultiplePasses, GetImageQuality, SetImageQuality*

The above functions produced a little-used novelty effect, transforming input video elements into simple polygons. They have not been replicated in the new Flix Engine.

Chapter 6

Samples

LINUX

Sample usage of the [Flix Engine COM](#) interface. The easiest way to test your installation is to run the [C++ command line](#) sample or the command line sample of your language of choice. By default samples are installed in `%PROGRAMFILES%\On2 Flix Engine (Demo)\samples`.

- ASP
 - [CGI](#)
- [ASP .NET](#)
 - [CGI](#)
- C++
 - [Command Line](#)
- [C#](#)
 - [Command Line](#)
- [ColdFusion](#)
 - [CGI](#)
- [Java](#)
 - [Command Line](#)
- [Perl](#)
 - [Command Line](#)
 - [CGI](#)
- [PHP](#)
 - [Command Line](#)
 - [CGI](#)
- [Visual Basic](#)
 - [Command Line](#)
- [Visual Basic .NET](#)
 - [Command Line](#)

See also:

Flix Engine COM [usage notes](#) for important information regarding updates between versions.

6.1 CGI

This example consists of 2 parts: [flix2_sample.asp](#) and [process_sample.asp](#). [flix2_sample.asp](#) searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's COM interfaces.

This example requires that the server process user accounts have any necessary access to the input and output directories. Depending on your web server configuration, you may need to give the IUSR_MACHINENAME and IWAM_MACHINENAME accounts or the NETWORK SERVICE account write access to the output directory.

To use this example navigate to [flix2_sample.asp](#) in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to [process_sample.asp](#) which runs the encode loop.

Default file locations used by the scripts:

Input : *C:/inetpub/flixmedia/in* (\$indir in [flix2_sample.asp](#))
Overlay : *C:/inetpub/flixmedia/overlay* (\$overlaydir in [flix2_sample.asp](#))
Output : *C:/inetpub/flixmedia/out* (\$outdir in [process_sample.asp](#))

6.1.1 flix2_sample.asp

```
<!--METADATA TYPE="typelib" UUID="AB5CF70C-0851-409A-97F2-C68F7058EA50" NAME="On2
.FlixEngine Type Library" -->
<%
'=====
'
' Copyright (c) On2 Technologies Inc. All Rights Reserved.
'
'-----
'
' File:          $Workfile$
'               $Revision$
'
' Last Update:  $DateUTC$
'-----
%>
<%

prefix      = "C:\inetpub\"
indir       = prefix & "flixmedia\in\"
overlaydir  = prefix & "flixmedia\overlay\"
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/str
ict.dtd">
<html lang="en-US">
<head>
<title>Flix2 CGI Sample - ASP</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<script type="text/javascript">
function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
    }
}
```

```

        i++;
    }
}

function toggle_expand(_this)
{
    _this.parentNode.className= (_this.parentNode.className=='expanded') ?
        'collapsed' : 'expanded';
    document.getElementById('showall').checked=false;
}

function set_table_visible(ctable,visible)
{
    var table= document.getElementById(ctable);

    if(table) {
        if(visible && table.className == 'hidden') {
            table.className= '';
        } else if(!visible && table.className == '') {
            table.className= 'hidden';
        }
    }
}

function hide_tablelist(vistablename,tablelist)
{
    var table= document.getElementById(vistablename);

    if(table&&table.className=='hidden') {
        table.className= '';
    }

    var i=0;
    while(tablelist[i]) {
        if(tablelist[i]!=vistablename) {
            var hiddentable= document.getElementById(tablelist[i]);

            set_table_visible(tablelist[i],false);
            /*clear down any values set in the hidden table to avoid posting
            unnecessary values*/
            clear_value(hiddentable.getElementsByTagName('input'));
            clear_value(hiddentable.getElementsByTagName('select'));
        }
        i++;
    }
}

function set_acodec_visible(ctable)
{
    var acodecs= new Array('aactable','aacplustable','amrnhtable','lametable','pcm
        table','vorbistable');
    hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
    var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8tabl
        e');
    hide_tablelist(ctable,vcodecs);
}

function set_muxer_visible(mtable)
{
    var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','
        tg2table','tgptable','webmtable');
    hide_tablelist(mtable,muxers);
}

```

```

function clear_value(list)
{
    var i=0;
    while(list[i]) {
        if(list[i].type=='checkbox') { list[i++].checked=false; }
        else { list[i++].value=''; }
    }
}

function toggle_ftable(ftable,enabled)
{
    var table= document.getElementById(ftable);

    if(table) {
        table.className = enabled ? '' : 'disabled';
        if(!enabled) {
            clear_value(table.getElementsByTagName('input'));
            clear_value(table.getElementsByTagName('select'));
        }
    }
}

function reset_tables()
{
    var table_list= document.getElementsByTagName('table');
    var i=0;
    while(table_list[i]) {
        if (table_list[i].id.length > 7 &&
            table_list[i].id.substring(0,7) == 'filter_') {
            table_list[i].className= 'disabled';
        } else if (table_list[i].id.indexOf('table') != -1) {
            table_list[i].className= 'hidden';
        }
        i++;
    }
}

</script>

<style type="text/css">
<!--
html {
    font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
    font-size: 100%;
    color: rgb(56,56,56);
    background-color: rgb(236,236,236);
    border-style: solid;
    border-color: rgb(236,236,236);
}

body {
    text-align: center;
    margin: 0 auto;
}

div.content {
    color: rgb(56,56,56);
    background-color: rgb(246,246,246);
    text-align: left;
    margin: 0 auto;
    width: 80%;
    min-width: 768px;
    max-width: 932px;
    border-width: 0 1px;
    border-color: rgb(144,144,144);
    border-style: solid;

```

```

}

div.content:after {
    content: "";
    color: inherit;
    background-color: rgb(250,250,250);
    border-top: 1px solid rgb(144,144,144);
    height: 20px;
    width: 100%;
    display: block;
}

fieldset table, fieldset {display: none;}
fieldset.expanded, fieldset.collapsed {display: block;}

/*first is fallback for IE*/
fieldset.expanded table {display: block;}
fieldset.expanded table {display: table;}

h1 {
    font-family: sans-serif;
    font-size: 150%;
    font-weight: normal;
    text-align: left;
    letter-spacing: -1px;
    color: rgb(74,74,74);
    background-color: inherit;
    margin: 0;
}

a {
    color: rgb(74,74,74);
    background-color: transparent;
}

label {
    font-size: 75%;
}

fieldset {
    font-size: 75%;
    line-height: 130%;
    padding: 0;
    margin: 20px;
    border: none;
}

fieldset.expanded {
    color: inherit;
    background-color: rgb(252,252,252);
    border-style: solid;
    border-width: 1px;
    border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
}

legend {
    padding: 0 5px;
    border-left: 10px solid rgb(217,217,217);
    cursor: pointer;
}

legend:hover {text-decoration: underline;}
fieldset.expanded legend {
    font-size: 150%;
    font-weight: bold;
    letter-spacing: -1px;
    background: transparent;
    margin-left: 12px;
}

```

```

        border-right: 10px solid rgb(217,217,217);
        display: block;
    }
    table {
        font-size: 100%;
        border-spacing: 0;
        /*border-collapse: collapse;*/
        width: 100%;
    }

    th:before { display: none; }
    th, td {
        width: 50%;
        vertical-align: top;
        padding: 2px 3px;
        border-width: 1px 0;
        border-style: solid;
        border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
    }
    tr:first-child>* {border-top-color: rgb(252,252,252);}
    th {
        font-weight: normal;
        text-align: left;
        padding: 2px 2px 2px 5px;
    }
    input[type] {
        font-family: monospace;
        font-size: 100%;
        color: rgb(56,56,56);
        background-color: inherit;
    }
    [type="text"], [type="number"] {
        margin-right: 13px;
        width: 222px;
    }

    [type="checkbox"].filter {
        margin-left: 4px
    }
    [type="checkbox"] {
        margin-left: 13px
    }
    [type="button"], [type="submit"] {
        font-size: 1em;
        margin: 0 2px 0 13px;
    }

    fieldset {
        font-size: 75%;
        margin: 20px 10px;
    }
    fieldset input[type="text"], fieldset input[type="number"] {
        width: 95% !important;
        margin: 0;
        display: block;
    }
    fieldset.expanded table.hidden {
        display: none;
    }

    table.disabled {
        color: rgb(176,176,176);
    }
    -->
</style>
</head>

```

```

<body>
  <div class='content'>
    <noscript>
      <p>This page requires javascript be enabled.</p>
    </noscript>

    <hr>
    <h1>Flix CGI Sample</h1>

    <p><small>flix2_sample.cgi version 1.9</small></p>
    <h4>Instructions</h4>
    <ul>
      <li>In this sample you must choose a source file and an output file.<br>
        If you leave all the other options blank then the sample will not call
        the corresponding Flix Engine function and the default will be used.<br>
        When done please press the "Start Encode" button at the bottom of the page.<br>
      <li>Mouse over a function name to see its default, if applicable.
      <li>Current source file directory: <%=indir%>
    </ul>

    <p>
      <label><input type="checkbox" id='showall' onclick='showall(this.checked)''>Show
        all</label>
    </p>

    <form action="process_sample.asp" method="post">

    <!-- ##SOURCE FILE##### -->

    <hr>
    <fieldset class='expanded' id="srcfile">
    <legend onclick='toggle_expand(this)''>Source File</legend>
    <table>

    <tr>
      <td>
        <%
          'Get the file names from indir
          Dim FSO
          Set FSO = CreateObject("Scripting.FileSystemObject")
          If (FSO.FolderExists(indir) = True) Then
            Dim inFolder
            Set inFolder = FSO.getFolder(indir)
            'Code snippet for sorted directory listing
            fileCount = inFolder.files.count
            Dim fileNames()
            ReDim fileNames(fileCount)
            fileCounter = 0
            For each file in inFolder.files
              fileCounter = fileCounter + 1
              fileNames(fileCounter) = file.name
            Next
            For fileOne = 1 To fileCount
              For fileTwo = (fileOne + 1) To fileCount
                If strComp(fileNames(fileOne),fileNames(fileTwo),0)=1 Then
                  buffer = fileNames(fileTwo)
                  fileNames(fileTwo) = fileNames(fileOne)
                  fileNames(fileOne) = buffer
                end if
              Next
            Next

            Dim siz
            siz = Ubound(fileNames)-1
            If (siz>15) Then siz= 15

```



```

'Output the sorted options
Response.Write("<select name='setInputFile' size='" & siz & "'>" & vbCrLf)
selected = False
Dim filename
For Each fileName In fileNames
    If (Len(fileName) > 0) Then
        Response.Write("<option ")
        If (selected = False) Then
            Response.Write("selected ")
            selected = True
        End If
        Response.Write("value=""" & indir & fileName & """" & fileName & "</option>" & vbCrLf)
    End If
Next
Response.Write("</select>" & vbCrLf)
Else
    Response.Write("WARNING couldn't open " & indir & ": FolderExists() returned FALSE<br>" & vbCrLf)
End If
%>

</td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE#####
-->
<hr>
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)'/>Output File</legend>
<table>

<tr>
<td>
<input type="text" name="setOutputFile" value="cgi-asp-out.flv">
</td>
</tr>

</table>
</fieldset>

<!-- ##MAIN OPTIONS#####
### -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)'/>Main Options</legend>
<table>

<tr>
<th><abbr title="Default: FALSE">setOverwriteExistingFiles</abbr></th>
<td>
<select name="setOverwriteExistingFiles">
<option value=""></option>
<option value="<%=on2true%>">TRUE</option>
<option value="<%=on2false%>">FALSE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: TRUE">setExportAudio</abbr></th>
<td>
<select name="setExportAudio">

```

```

<option value=""></option>
<option value="<%=on2true%>">TRUE</option>
<option value="<%=on2false%>">FALSE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: TRUE">setExportVideo</abbr></th>
<td>
<select name="setExportVideo">
<option value=""></option>
<option value="<%=on2true%>">TRUE</option>
<option value="<%=on2false%>">FALSE</option>
</select>
</td>
</tr>

</table>
</fieldset>

<!-- ##CODECS#####
### -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this) '>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
<td>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
onfocus="set_vcodec_visible('vp8table') ">
<abbr title="For use with WebM">FE2_CODEC_VP8</abbr>&nbsp;
</label>
<br>

<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
onfocus="set_vcodec_visible('vp6table') ">
<abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr>&nbsp;
</label>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
onfocus="set_vcodec_visible('vp6atable') ">
<abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr>&nbsp;
</label>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_H263'
onfocus="set_vcodec_visible('h263table') ">
<abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr>&nbsp;
</label>
<br>

<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
onfocus="set_vcodec_visible('h263table') ">
<abbr title="For use with 3GP">FE2_CODEC_H263_BASELINE</abbr>&nbsp;
</label>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_H264'
onfocus="set_vcodec_visible('h264table') ">
<abbr title="For use with 3GP/3G2/MOV/MP4">FE2_CODEC_H264</abbr>&nbsp;
</label>

<!-- VP6 codec parameters -->

```

```

<table id='vp6table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_VP6_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_KFINTTYPE">
        <option value=""></option>
        <option value="<%=MAX_KEYFRAMES%>">MAX_KEYFRAMES</option>
        <option value="<%=FIXED_KEYFRAMES%>">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_VP6_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP6_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_RC_MODE">
        <option value=""></option>
        <option value="<%=VBR_2PASSControl%>">VBR_2PASSControl</option>
        <option value="<%=CBR_2PASSControl%>">CBR_2PASSControl</option>
        <option value="<%=VBR_1PASSControl%>">VBR_1PASSControl</option>
        <option value="<%=CBR_1PASSControl%>">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_CXMODE">
        <option value=""></option>
        <option value="<%=COMPRESSMODE_GOOD%>">COMPRESSMODE_GOOD</option>
        <option value="<%=COMPRESSMODE_BEST%>">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: VP6_E">FE2_VP6_PROFILE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_PROFILE">
        <option value=""></option>
        <option value="<%=VP6_E%>">VP6_E</option>
        <option value="<%=VP6_S%>">VP6_S</option>
      </select>
    </td>
  </tr>

  <tr><th><b>Advanced Settings:</b></th></tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_CONCURRENCY'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>

```

```

    <td><input type='text' name='codec:setParam:FE2_VP6_UNDERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_TEMPORAL_RESAMPLING</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_TEMPORAL_DOWN_WATERMARK</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
  <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_2PASS_MIN_SECTION'></td>

```

```

</tr>

<tr>
  <th><abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_2PASS_MAX_SECTION'></td>
</tr>

</table>
<!-- END - VP6 codec parameters -->

<!-- VP6A codec parameters -->
<table id='vp6atable' class='hidden'>
  <tr>
    <th><abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 68kbps (15% of default 448kbps)">
      FE2_VP6A_ALPHA_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6A_KFINTTYPE">
        <option value=""></option>
        <option value="%=MAX_KEYFRAMES%">MAX_KEYFRAMES</option>
        <option value="%=FIXED_KEYFRAMES%">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_VP6A_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6A_RC_MODE">
        <option value=""></option>
        <option value="%=VBR_2PASSControl%">VBR_2PASSControl</option>
        <option value="%=CBR_2PASSControl%">CBR_2PASSControl</option>
        <option value="%=VBR_1PASSControl%">VBR_1PASSControl</option>
        <option value="%=CBR_1PASSControl%">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6A_CXMODE">
        <option value=""></option>
        <option value="%=COMPRESSMODE_GOOD%">COMPRESSMODE_GOOD</option>
        <option value="%=COMPRESSMODE_BEST%">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr><th><b>Advanced Settings:</b></th></tr>

  <tr>

```

```

    <th><abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_MIN_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_ALPHA_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MIN_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_SHARPNESS'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_NOISE_REDUCTION'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_TEMPORAL_RESAMPLING</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_TEMPORAL_DOWN_WATERMARK</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'>
        </td>
</tr>

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>   <th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_STREAM_PREBUFFER'></td>   <th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr> </th>  <input type='text' name='codec:setParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>   <th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>   <th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_2PASS_MIN_SECTION'></td>   <th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_2PASS_MAX_SECTION'></td>   <th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>  <input type='text' name='codec:setParam:FE2_H263_BITRATE'></td>   <th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>  <select name="codec:setParam:FE2_H263_KFINTTYPE"> <option value=""></option> <option value="<%=MAX_KEYFRAMES%>">MAX_KEYFRAMES</option> <option value="<%=FIXED_KEYFRAMES%>">FIXED_KEYFRAMES</option> </select> | <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown"> FE2_H263_KFFREQ</abbr></th>  <input type='text' name='codec:setParam:FE2_H263_KFFREQ'></td>   <th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>  <select name="codec:setParam:FE2_H263_RC_MODE"> <option value=""></option> <option value="<%=VBR_2PASSControl%>">VBR_2PASSControl</option> <option value="<%=CBR_2PASSControl%>">CBR_2PASSControl</option> | | | | | | | | | | | | | | | | | | |

```

```

        <option value="<%=VBR_1PASSControl%>">VBR_1PASSControl</option>
        <option value="<%=CBR_1PASSControl%>">CBR_1PASSControl</option>
    </select>
</td>
</tr>

<tr>
    <th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H263_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H263_MIN_Q'></td>
</tr>
</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
    <tr>
        <th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_H264_BITRATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
            FE2_H264_KFFREQ</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_H264_KFFREQ'></td>
    </tr>

    <tr>
        <th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
        <td>
            <select name="codec:setParam:FE2_H264_RC_MODE">
                <option value=""></option>
                <option value="<%=VBR_1PASSControl%>">VBR_1PASSControl</option>
                <option value="<%=CBR_1PASSControl%>">CBR_1PASSControl</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
        <td>
            <select name="codec:setParam:FE2_H264_PROFILE">
                <option value=""></option>
                <option value="<%=BASE_H264PROFILE%>">BASE_H264PROFILE</option>
                <option value="<%=MAIN_H264PROFILE%>">MAIN_H264PROFILE</option>
                <option value="<%=HIGH_H264PROFILE%>">HIGH_H264PROFILE</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_H264_B_FRAME_RATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: Dependent on profile selection, see API docs. Valid
            Range [0,5]">FE2_H264_SPEED</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_H264_SPEED'></td>
    </tr>
</table>

```



```

<!-- END - H264 codec parameters -->

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_KFINTTYPE">
        <option value=""></option>
        <option value="<%=MAX_KEYFRAMES%>">MAX_KEYFRAMES</option>
        <option value="<%=FIXED_KEYFRAMES%>">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_VP8_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_RC_MODE">
        <option value=""></option>
        <option value="<%=VBR_2PASSControl%>">VBR_2PASSControl</option>
        <option value="<%=CBR_2PASSControl%>">CBR_2PASSControl</option>
        <option value="<%=VBR_1PASSControl%>">VBR_1PASSControl</option>
        <option value="<%=CBR_1PASSControl%>">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_CXMODE">
        <option value=""></option>
        <option value="<%=COMPRESSMODE_GOOD%>">COMPRESSMODE_GOOD</option>
        <option value="<%=COMPRESSMODE_BEST%>">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_THREADS'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_PROFILE'></td>
  </tr>

  <tr><th><b>Advanced Settings:</b></th></tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_LAG'></td>
  </tr>

```

```

<tr>
  <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_UNDERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_OVERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
  <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_ALTREF'></td>
</tr>

```

```
|  |  |
| --- | --- |
| <th><abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th> | <td><input type='text' name='codec:setParam:FE2_VP8_AR_MAX_FRAMES'></td> |
| <th><abbr title="">FE2_VP8_AR_TYPE</abbr></th> | <td><input type='text' name='codec:setParam:FE2_VP8_AR_TYPE'></td> |
| <th><abbr title="">FE2_VP8_AR_STRENGTH</abbr></th> | <td><input type='text' name='codec:setParam:FE2_VP8_AR_STRENGTH'></td> |
| <th><abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th> | <td><input type='text' name='codec:setParam:FE2_VP8_MB_STATIC_THRESHOLD'></td> |
| <th><abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th> | <td><input type='text' name='codec:setParam:FE2_VP8_TOKEN_PARTITIONS'></td> |


<!-- END - VP8 codec parameters -->

</td>
</tr> <!-- END - video codecs -->



```

```

        onfocus="set_acodec_visible('amrnhtable') ">
        <abbr title="For use with 3GP">FE2_CODECEC_AMR_NB</abbr>&nbsp;
    </label>

<!-- AMR_NB codec parameters -->
<table id='amrnhtable' class='hidden'>
    <tr>
        <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_AMR_BITRATE'></td>
    </tr>

</table>
<!-- END - AMR_NB codec parameters -->

<!-- AAC codec parameters -->
<table id='aactable' class='hidden'>
    <tr>
        <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_AAC_BITRATE'></td>
    </tr>

</table>
<!-- END - AAC codec parameters -->

<!-- AACPLUS codec parameters -->
<table id='aacplustable' class='hidden'>
    <tr>
        <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_AACPLUS_BITRATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: disabled (aacPlus v1)">
            FE2_AACPLUS_PARAMETRIC_STEREO</abbr></th>
        <td>
            <select name="codec:setParam:FE2_AACPLUS_PARAMETRIC_STEREO">
                <option value=""></option>
                <option value="0">disable (aacPlus v1)</option>
                <option value="1">enable (aacPlus v2)</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - AACPLUS codec parameters -->

<!-- LAME codec parameters -->
<table id='lametable' class='hidden'>
    <tr>
        <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_LAME_BITRATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_LAME_QUALITY'></td>
    </tr>

    <tr>
        <th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
        <td>
            <select name="codec:setParam:FE2_LAME_RC_MODE">
                <option value=""></option>
                <option value="<%=LAME_CBR%>">LAME_CBR</option>
                <option value="<%=LAME_ABR%>">LAME_ABR</option>
                <option value="<%=LAME_VBR_rh%>">LAME_VBR_rh</option>
                <option value="<%=LAME_VBR_mtrh%>">LAME_VBR_mtrh</option>
            </select>
        </td>
    </tr>
</table>

```

```

        </td>
      </tr>
    </table>
    <!-- END - LAME codec parameters -->

    <!-- PCM codec parameters -->
    <table id='pcmtable' class='hidden'>
      <tr>
        <th>(FE2_CODEEC_PCM defines no parameters)</th>
      </tr>
    </table>
    <!-- END - PCM codec parameters -->

    <!-- VORBIS codec parameters -->
    <table id='vorbistable' class='hidden'>
      <tr>
        <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VORBIS_BITRATE'></td>
      </tr>
    </table>
    <!-- END - VORBIS codec parameters -->

  </td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS#####
    ### -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this)''>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_CUT' value='avfilter:'
        onchange="toggle_ftable('filter_cut',this.checked)">
      FE2_FILTER_CUT
    </label>

    <table id='filter_cut' class='disabled'>
      <tr>
        <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CUT_START_SEC'></td>
      </tr>

      <tr>
        <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CUT_STOP_SEC'></td>
      </tr>

      <tr>
        <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>
        <td>
          <select name="filter:setParam:FE2_CUT_USE_SEEK">
            <option value=""></option>
            <option value="<%=on2false%>">FALSE</option>
            <option value="<%=on2true%>">TRUE</option>
          </select>
        </td>
      </tr>
    </table>
  </td>
</tr>

```

```

        </tr>

    </table>

    </td>
</tr>
<!-- END - CUT filter parameters -->

<tr><th><b>Video Filters</b></th></tr>

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:'
            onchange="toggle_ftable('filter_adaptive_deinterlace',this.checked)">
        FE2_FILTER_ADAPTIVE_DEINTERLACE
    </label>

    <table id='filter_adaptive_deinterlace' class='disabled'>
    <tr>
    <th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</abbr>
        <br></th>
    <td>
        <select name="filter:setParam:FE2_ADAPTIVE_DEINTERLACE_MODE">
        <option value=""></option>
        <option value="<%=DEINTERLACE_NONE%>">DEINTERLACE_NONE</option>
        <option value="<%=DEINTERLACE_1_2_1_BLUR%>">DEINTERLACE_1_2_1_BLUR</option>
        <option value="<%=DEINTERLACE_DROP_FIELD%>">DEINTERLACE_DROP_FIELD</option>
        <option value="<%=DEINTERLACE_ADAPTIVE%>">DEINTERLACE_ADAPTIVE</option>
        </select>
    </td>
    </tr>
    </table>

    </td>
</tr>
<!-- END - ADAPTIVE DEINTERLACE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_BCHS' value='vfilter:'
            onchange="toggle_ftable('filter_bchs',this.checked)">
        FE2_FILTER_BCHS
    </label>

    <table id='filter_bchs' class='disabled'>
    <tr>
    <th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_BCHS_BRIGHTNESS'></td>
    </tr>
    <tr>
    <th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_BCHS_CONTRAST'></td>
    </tr>
    <tr>
    <th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_BCHS_HUE'></td>
    </tr>
    <tr>
    <th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_BCHS_SATURATION'></td>
    </tr>
    </table>

```

```

</td>
</tr>
<!-- END - BCHS filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_BLUR' value='vfilter:'
onchange="toggle_fhtable('filter_blur',this.checked) ">
FE2_FILTER_BLUR
</label>

<table id='filter_blur' class='disabled'>
<tr>
<th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
<td>
<select name="filter:setParam:FE2_BLUR_FILTER">
<option value=""></option>
<option value="<%=BLUR_LOWPASS%>">BLUR_LOWPASS</option>
<option value="<%=BLUR_GAUSS%>">BLUR_GAUSS</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
<td>
<select name="filter:setParam:FE2_BLUR_MASKSIZE">
<option value=""></option>
<option value="<%=MASK_3x3%>">MASK_3x3</option>
<option value="<%=MASK_5x5%>">MASK_5x5</option>
</select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_CROP' value='vfilter:'
onchange="toggle_fhtable('filter_crop',this.checked) ">
FE2_FILTER_CROP
</label>

<table id='filter_crop' class='disabled'>
<tr>
<th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_TOP'></td>
</tr>

<tr>
<th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_BOTTOM'></td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_LEFT'></td>
</tr>

<tr>

```

```

        <th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CROP_RIGHT'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - CROP filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_DENOISE' value='vfilter:'
            onchange="toggle_ftable('filter_denoise',this.checked) ">
        FE2_FILTER_DENOISE
    </label>

    <table id='filter_denoise' class='disabled'>
        <tr>
            <th><abbr title="Default: 0. Range: [0.0,1.0]">FE2_DENOISE_NOISE_LEVEL</abbr>
            </th>
            <td><input type='text' name='filter:setParam:FE2_DENOISE_NOISE_LEVEL'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_FRAMERATE' value='vfilter:'
            onchange="toggle_ftable('filter_framerate',this.checked) ">
        FE2_FILTER_FRAMERATE
    </label>

    <table id='filter_framerate' class='disabled'>
        <tr>
            <th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAM
            ERATE_DECIMATE</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_FRAMERATE_DECIMATE'></td>
        </tr>

        <tr>
            <th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FR
            AMERATE_FPS</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_FRAMERATE_FPS'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_MIRROR' value='vfilter:'
            onchange="toggle_ftable('filter_mirror',this.checked) ">
        FE2_FILTER_MIRROR
    </label>

    <table id='filter_mirror' class='disabled'>

```



```

<tr>
  <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
  <td>
    <select name="filter:setParam:FE2_MIRROR_HORIZONTAL">
      <option value=""></option>
      <option value="<%=on2false%>">FALSE</option>
      <option value="<%=on2true%>">TRUE</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
  <td>
    <select name="filter:setParam:FE2_MIRROR_VERTICAL">
      <option value=""></option>
      <option value="<%=on2false%>">FALSE</option>
      <option value="<%=on2true%>">TRUE</option>
    </select>
  </td>
</tr>
</table>

</td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_OVERLAY' value='vfilter:'
        onchange="toggle_ftable('filter_overlay',this.checked)">
      FE2_FILTER_OVERLAY
    </label>

    <table id='filter_overlay' class='disabled'>
      <tr>
        <th>
          <abbr title="Currently searching <%=overlaydir%> for overlay images">FE2_OVE
            RLAY_FILE</abbr>
        </th>
        <td>
          <%
'Get the file names from overlaydir
'Several objects here, such as FSO, are being reused from parsing the indir fi
  lenames
'Set FSO = CreateObject("Scripting.FileSystemObject")
If (FSO.FolderExists(overlaydir) = True) Then
  Dim ovrlFolder
  Set ovrlFolder = FSO.getFolder(overlaydir)
  'Code snippet for sorted directory listing
  fileCount = ovrlFolder.files.count
  'Set fileNames = Nothing
  ReDim fileNames(fileCount)
  fileCounter = 0
  For each file in ovrlFolder.files
    fileCounter = fileCounter + 1
    fileNames(fileCounter) = file.name
  Next
  For fileOne = 1 To fileCount
    For fileTwo = (fileOne + 1) To fileCount
      If strComp(fileNames(fileOne),fileNames(fileTwo),0)=1 Then
        buffer = fileNames(fileTwo)
        fileNames(fileTwo) = fileNames(fileOne)
        fileNames(fileOne) = buffer
      end if
    
```

```

        Next
    Next

    'Output the sorted options
    Response.Write("<select name=\""filter:setParamAsStr:FE2_OVERLAY_FILE\"" &
vbLf)
    Response.Write("<option value=\""\"></option>" & vbLf)
    For Each fileName In fileNames
        If ( (InStr(1, fileName, ".bmp", 1) > 0) Or (InStr(1, fileName, ".png",
1) > 0)) Then
            Response.Write("<option value='" & overlaydir & fileName & "'" & fil
eName & "</option>" & vbLf)
        End If
    Next
    Response.Write("</select>" & vbLf)
    Response.Write("<br>default: None, must be programmatically set to" _
        & " absolute path of overlay input file, e.g., '" _
        & overlaydir & "overlay.png'" & vbLf)
Else
    Response.Write("WARNING couldn't open " & overlaydir & ": FolderExists() re
turned FALSE<br>" & vbLf)
End If
%>

</td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
<td>
<select name="filter:setParam:FE2_OVERLAY_MASK_XY">
<option value=""></option>
<option value="<%=on2false%>">FALSE</option>
<option value="<%=on2true%>">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_Y'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
<td>
<select name="filter:setParam:FE2_OVERLAY_MASK_RGB">
<option value=""></option>
<option value="<%=on2false%>">FALSE</option>
<option value="<%=on2true%>">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_R'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_G'></td>
</tr>
<tr>

```

```

    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_B'></td>
</tr>

<tr>
<th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
<td>
    <select name="filter:setParam:FE2_OVERLAY_POS">
        <option value=""></option>
        <option value="<%=FE2_OVERLAY_POS_MODE_TOPLEFT%>">
            FE2_OVERLAY_POS_MODE_TOPLEFT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_BOTLEFT%>">
            FE2_OVERLAY_POS_MODE_BOTLEFT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_CENTER%>">
            FE2_OVERLAY_POS_MODE_CENTER</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_TOPRIGHT%>">
            FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_BOTRIGHT%>">
            FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_XY%>">FE2_OVERLAY_POS_MODE_XY</optio
            n>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_Y'></td>
</tr>

</table>

</td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_PNGEX' value='vfilter:'
            onchange="toggle_ftable('filter_pngex',this.checked)">
        FE2_FILTER_PNGEX
    </label>

    <table id='filter_pngex' class='disabled'>
    <tr>
        <th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</
            abbr></th>
        <td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'
            ></td>
    </tr>

    <tr>
        <th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
        <td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_PREFIX'>
            </td>
    </tr>

    <tr>
        <th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
        <td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'>
            </td>
    </tr>

```

```

</tr>

<tr>
<th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_WIDTH'></td>
</tr>

<tr>
<th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_HEIGHT'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>

<td>
<select name="filter:setParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">
<option value=""></option>
<option value="<%=on2false%>">FALSE</option>
<option value="<%=on2true%>">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
<td>
<select name="filter:setParam:FE2_PNGEX_ENABLE_ALPHA">
<option value=""></option>
<option value="<%=on2false%>">FALSE</option>
<option value="<%=on2true%>">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="interval in ms; Default: disabled">
FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_EXPORT_INTERVAL'></td>
</tr>

<tr>
<th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">
FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
<td><input type='text' name='filter:setParamAsStr:
FE2_PNGEX_EXPORT_TIME_STRING'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
<td>
<select name="filter:setParam:FE2_PNGEX_EXPORT_CUE_POINTS">
<option value=""></option>
<option value="<%=FE2_PNGEX_CP_ALL%>">All cue points (FE2_PNGEX_CP_ALL)</option>
<option value="<%=FE2_PNGEX_CP_NAV%>">Only navigation cue points (
FE2_PNGEX_CP_NAV)</option>
<option value="<%=FE2_PNGEX_CP_EVENT%>">Only event cue points (
FE2_PNGEX_CP_EVENT)</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">
FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>

```

```

        <td><input type='text' name='filter:setParam:FE2_PNGEX_COMPRESSION_LEVEL'></td>
    </tr>

    <tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>

    <tr>
        <th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>
        <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></td>
    </tr>

    <tr>
        <th><abbr title="start time in ms; Default: 0">
            FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
        <td><input type='text' name='filter:setParam:
            FE2_PNGEX_AUTO_EXPORT_START_TIME'></td>
    </tr>

    <tr>
        <th><abbr title="stop time in ms; Default: <clip length>">
            FE2_PNGEX_AUTO_EXPORT_END_TIME</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_END_TIME'>
            </td>
    </tr>

    <tr>
        <th><abbr title="Default: 0">FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD</abbr></th>
        <td><input type='text' name='filter:setParam:
            FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - PNGEX filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_ROTATE' value='vfilter:'
                onchange="toggle_ftable('filter_rotate',this.checked)">
            FE2_FILTER_ROTATE
        </label>

        <table id='filter_rotate' class='disabled'>
            <tr>
                <th><abbr title="Default: 0. valid: {0,90,180,270}">FE2_ROTATE_ANGLE</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_ROTATE_ANGLE'></td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - ROTATE filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_SCALE' value='vfilter:'
                onchange="toggle_ftable('filter_scale',this.checked)">
            FE2_FILTER_SCALE
        </label>

```

```

<table id='filter_scale' class='disabled'>
  <tr>
    <th><abbr title="Default: input image width">FE2_SCALE_WIDTH</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_SCALE_WIDTH'></td>
  </tr>

  <tr>
    <th><abbr title="Default: input image height">FE2_SCALE_HEIGHT</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_SCALE_HEIGHT'></td>
  </tr>
</table>

</td>
</tr>
<!-- END - SCALE filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_SHARPEN' value='vfilter:'
        onchange="toggle_ftable('filter_sharpen',this.checked)">
      FE2_FILTER_SHARPEN
    </label>

    <table id='filter_sharpen' class='disabled'>
      <tr>
        <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
      </tr>
    </table>

  </td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_HIGHPASS' value='afilter:'
        onchange="toggle_ftable('filter_highpass',this.checked)">
      FE2_FILTER_HIGHPASS
    </label>

    <table id='filter_highpass' class='disabled'>
      <tr>
        <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_HIGHPASS_Q'></td>
      </tr>

      <tr>
        <th>FE2_HIGHPASS_CUTOFF</th>
        <td><input type='text' name='filter:setParam:FE2_HIGHPASS_CUTOFF'></td>
      </tr>
    </table>

  </td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'

```

```

        name='FE2_FILTER_LOWPASS' value='afilter:'
        onchange="toggle_ftable('filter_lowpass',this.checked)">
FE2_FILTER_LOWPASS
</label>

<table id='filter_lowpass' class='disabled'>
<tr>
<th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>
<td><input type='text' name='filter:setParam:FE2_LOWPASS_Q'></td>
</tr>

<tr>
<th>FE2_LOWPASS_CUTOFF</th>
<td><input type='text' name='filter:setParam:FE2_LOWPASS_CUTOFF'></td>
</tr>
</table>

</td>
</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_RESAMPLE' value='afilter:'
onchange="toggle_ftable('filter_resample',this.checked)">
FE2_FILTER_RESAMPLE
</label>

<table id='filter_resample' class='disabled'>
<tr>
<th><abbr title="Default: 0">FE2_RESAMPLE_RATE</abbr></th>
<td><input type='text' name='filter:setParam:FE2_RESAMPLE_RATE'></td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_RESAMPLE_CHANNELS</abbr></th>
<td><input type='text' name='filter:setParam:FE2_RESAMPLE_CHANNELS'></td>
</tr>
</table>

</td>
</tr>
<!-- END - RESAMPLE filter parameters -->

</table>
</fieldset>

<!-- ##MUXERS#####
### -->

<hr>
<fieldset class='collapsed' id="muxers">
<legend onclick='toggle_expand(this)'>Muxers</legend>
<table>

<tr>
<td>
<label>
<input type='radio' name='muxer:' value='FE2_MUXER_3GP'
onfocus="set_muxer_visible('tgptable')">
FE2_MUXER_3GP&nbsp;
</label>
<label>
<input type='radio' name='muxer:' value='FE2_MUXER_3G2'
onfocus="set_muxer_visible('tg2table')">
FE2_MUXER_3G2&nbsp;

```

```

</label>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MOV'
    onfocus="set_muxer_visible('movtable') ">
    FE2_MUXER_MOV    
</label>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MP4'
    onfocus="set_muxer_visible('mp4table') ">
    FE2_MUXER_MP4    
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FLV'
    onfocus="set_muxer_visible('flvtable',true) ">
    FE2_MUXER_FLV    
</label>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_SWF'
    onfocus="set_muxer_visible('swftable',true) ">
    FE2_MUXER_SWF
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
    onfocus="set_muxer_visible('fxmtable',true) ">
    FE2_MUXER_FXM    
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
    onfocus="set_muxer_visible('webmtable',true) ">
    FE2_MUXER_WEBM    
</label>
</td>
</tr>

<tr>
  <td>
    <!-- 3GP muxer parameters -->
    <table id='tgptable' class='hidden'>
      <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
          <select name="muxer:setParam:FE2_3GP_FASTSTART">
            <option value=""></option>
            <option value="<%=on2false%>">FALSE</option>
            <option value="<%=on2true%>">TRUE</option>
          </select>
        </td>
      </tr>
    </table>
    <!-- END - 3GP muxer parameters -->

    <!-- 3G2 muxer parameters -->
    <table id='tg2table' class='hidden'>
      <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
          <select name="muxer:setParam:FE2_3G2_FASTSTART">
            <option value=""></option>
            <option value="<%=on2false%>">FALSE</option>
            <option value="<%=on2true%>">TRUE</option>
          </select>
        </td>
      </tr>
    </table>

```



```

        </td>
    </tr>
</table>
<!-- END - 3G2 muxer parameters -->

<!-- FLV muxer parameters -->
<table id='flvtable' class='hidden'>
    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'"
            >
            FE2_FLV_CUEPT_EVENT</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
    </tr>

    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'"
            >
            FE2_FLV_CUEPT_NAV</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_NAV'></td>
    </tr>

    <tr>
        <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'"
            >
            FE2_FLV_CUEPT_PARAM</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
    </tr>

    <tr>
        <th><abbr title="Select specific metadata entries to enable. Default for each
            item is provided."
            >
            FE2_FLV_METADATA_ENABLE</abbr></th>
        <td>
            <table id='flv_metadata_enable' class=''>
                <tr>
                    <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr
                        title="Default: Enabled">MD_DURATION</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                        DURATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' value='<%=MD_
                        DURATION%'></td>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr
                        title="Default: Enabled">MD_DATASIZE</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                        DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' value='<%=MD_
                        DATASIZE%'></td>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><ab
                        br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                        AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' value='<%=
                        MD_AUDIO_SIZE%'></td>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><ab
                        br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                        VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' value='<%=
                        MD_VIDEO_SIZE%'></td>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'

```

```

><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' v
alue='<%=MD_AUDIO_DATARATE%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'
><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' v
alue='<%=MD_VIDEO_DATARATE%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'>
<abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' val
ue='<%=MD_AUDIO_CODECID%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'>
<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' val
ue='<%=MD_VIDEO_CODECID%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr ti
tle="Default: Enabled">MD_WIDTH</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' value='<%=MD_WIDTH%
>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' value='<%=MD_HEIG
HT%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' value='<%=M
D_FRAMERATE%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' value
='<%=MD_CANSEEKTOEND%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'>
<abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' val

```

```

ue='<%=MD_LASTTIMESTAMP%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP' value='<%=MD_LASTKEYFRAMETIMESTAMP%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION' value='<%=MD_LASTKEYFRAMELOCATION%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' value='<%=MD_KEYFRAMES%>'></td>
</tr>
</table>
</td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
FE2_FLV_METADATA_DISABLE</abbr></th>
<td>
<table id='flv_metadata_disable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abbr title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' value='<%=MD_DURATION%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' value='<%=MD_DATASIZE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><abbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' value='<%=MD_AUDIO_SIZE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' value='

```

```

<%=MD_VIDEO_SIZE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE' value='<%=MD_AUDIO_DATARATE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE' value='<%=MD_VIDEO_DATARATE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' value='<%=MD_AUDIO_CODECID%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' value='<%=MD_VIDEO_CODECID%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabled">MD_WIDTH</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' value='<%=MD_WIDTH%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enabled">MD_HEIGHT</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' value='<%=MD_HEIGHT%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' value='<%=MD_FRAMERATE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' value='<%=MD_CANSEEKTOEND%>'></td>
</tr>

```

```

        <tr>
            <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP'
><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' v
alue='<%=MD_LASTTIMESTAMP%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETI
MESTAMP'><abbr title="Default: Enabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYF
RAMETIMESTAMP' value='<%=MD_LASTKEYFRAMETIMESTAMP%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Enabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='<%=MD_LASTKEYFRAMELOCATION%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><ab
br title="Default: Enabled">MD_KEYFRAMES</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES' value='<
%=MD_KEYFRAMES%>'></td>
        </tr>
    </table>
</td>
</tr>
</table>
<!-- END - FLV muxer parameters -->

<!-- FXM muxer parameters -->
<table id='fxmtable' class='hidden'>
    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0' "
>
            FE2_FXM_CUEPT_EVENT</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_EVENT'></td>
    </tr>

    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.
0' "
>
            FE2_FXM_CUEPT_NAV</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_NAV'></td>
    </tr>

    <tr>
        <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...' "
>
            FE2_FXM_CUEPT_PARAM</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM'></td>
    </tr>

    <tr>
        <th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
            FE2_FXM_METADATA_ENABLE</abbr></th>
        <td>
            <table id='fxm_metadata_enable' class=''>

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Enabled">MD_DURATION</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ DURATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION' value='<%=MD_ DURATION%>'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Enabled">MD_DATASIZE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' value='<%=MD_ DATASIZE%>'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><ab br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' value='<%= MD_AUDIO_SIZE%>'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><ab br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' value='<%= MD_VIDEO_SIZE%>'></td> </tr>  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' ><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' v alue='<%=MD_AUDIO_DATARATE%>'></td> </tr>  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' ><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' v alue='<%=MD_VIDEO_DATARATE%>'></td> </tr>  |  |  |  |  |  | | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'> <abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' val ue='<%=MD_AUDIO_CODECID%>'></td> </tr>  |  |  |  | | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'> <abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' val ue='<%=MD_VIDEO_CODECID%>'></td> </tr>  | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr ti tle="Default: Enabled">MD_WIDTH</abbr></label></th> | | --- | | | | | | | | | | | | | | | | |

```

```

        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' value='<%=MD_WIDTH%
>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' value='<%=MD_HEIG
HT%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' value='<%=M
D_FRAMERATE%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' value
='<%=MD_CANSEEKTOEND%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'>
<abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' val
ue='<%=MD_LASTTIMESTAMP%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIM
ESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRA
METIMESTAMP' value='<%=MD_LASTKEYFRAMETIMESTAMP%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOC
ATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAM
ELOCATION' value='<%=MD_LASTKEYFRAMELOCATION%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abb
r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' value='<%=M
D_KEYFRAMES%>'></td>
    </tr>
</table>
</td>
</tr>

<tr>

```

```

<th><abbr title="Select specific metadata entries to enable. Default for each
    item is provided.">
        FE2_FXM_METADATA_DISABLE</abbr></th>
<td>
    <table id='fxm_metadata_disable' class=''>
        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abbr
                title="Default: Enabled">MD_DURATION</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _DURATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' value='<%=M
                D_DURATION%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abbr
                title="Default: Enabled">MD_DATASIZE</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' value='<%=M
                D_DATASIZE%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><a
                bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' value='
                <%=MD_AUDIO_SIZE%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><a
                bbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' value='
                <%=MD_VIDEO_SIZE%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE
                '><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'
                value='<%=MD_AUDIO_DATARATE%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE
                '><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'
                value='<%=MD_VIDEO_DATARATE%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'
                '><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' v
                alue='<%=MD_AUDIO_CODECID%>'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'
                '><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
                _VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' v
                alue='<%=MD_VIDEO_CODECID%>'></td>
        </tr>
    </table>

```



```

</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr t
itle="Default: Enabled">MD_WIDTH</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_WIDTH' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' value='<%=MD_WIDT
H%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr
title="Default: Enabled">MD_HEIGHT</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' value='<%=MD_HE
IGHT%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><ab
br title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' value='<%
=MD_FRAMERATE%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'>
<abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' val
ue='<%=MD_CANSEEKTOEND%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'
><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' v
alue='<%=MD_LASTTIMESTAMP%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETI
MESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label><
/th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYF
RAMETIMESTAMP' value='<%=MD_LASTKEYFRAMETIMESTAMP%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='<%=MD_LASTKEYFRAMELOCATION%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><ab
br title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' value='<%
=MD_KEYFRAMES%>'></td>
</tr>

```

```

        </table>
    </td>
</tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>
    <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_MOV_FASTSTART">
                <option value=""></option>
                <option value="<%=on2false%>">FALSE</option>
                <option value="<%=on2true%>">TRUE</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - MOV muxer parameters -->

<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
    <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_MP4_FASTSTART">
                <option value=""></option>
                <option value="<%=on2false%>">FALSE</option>
                <option value="<%=on2true%>">TRUE</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
    <tr>
        <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_WIDTH'></td>
    </tr>

    <tr>
        <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_HEIGHT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_FRAMERATE'></td>
    </tr>

    <tr>
        <th>FE2_SWF_LOOP_COUNT</th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_LOOP_COUNT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
    </tr>

    <tr>
        <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
        <td>
            <select name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>

```

```

        <option value=""></option>
        <option value="_self">_self</option>
        <option value="_blank">_blank</option>
        <option value="_parent">_parent</option>
        <option value="_top">_top</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</a
    bbr></th>
<td>
    <select name='muxer:setParam:FE2_SWF_EMBEDDED_URL_TYPE'>
        <option value=""></option>
        <option value="<%=EmbeddedUrlIsGetUrl%>">EmbeddedUrlIsGetUrl</option>
        <option value="<%=EmbeddedUrlIsLoadMovie%>">EmbeddedUrlIsLoadMovie</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="e.g. n0=v0& n1=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
<td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
</tr>

<tr><th><b>Preloader Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
<td>
    <select name='muxer:setParam:FE2_SWF_PRELOAD_TYPE'>
        <option value=""></option>
        <option value="<%=SwfPreloaderNone%>">SwfPreloaderNone</option>
        <option value="<%=SwfFixedPreloader%>">SwfFixedPreloader</option>
        <option value="<%=SwfAdaptivePreloader%>">SwfAdaptivePreloader</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
<th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FA
    CTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTIO
    N</abbr></th>
<td>
    <select name='muxer:setParam:FE2_SWF_ON_START_OPTION'>
        <option value=""></option>
        <option value="<%=SwfOnMovieStartAutomatically%>">SwfOnMovieStartAutomatica
            ly</option>
        <option value="<%=SwfOnMovieStartOnClick%>">SwfOnMovieStartOnClick</option>
        <option value="<%=SwfOnMovieStartWait%>">SwfOnMovieStartWait</option>
        <option value="<%=SwfOnMovieStartEmbedSTOP%>">SwfOnMovieStartEmbedSTOP</opti
            on>
    </select>

```

```

    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_START_BLANK_FRAME'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_START_WAIT_SEC'></td>
  </tr>

  <tr><th><b>End Settings:</b></th><td></td></tr>

  <tr>
    <th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
    <td>
      <select name='muxer:setParam:FE2_SWF_ON_END_OPTION'>
        <option value=""></option>
        <option value="<%=SwfOnMovieEndNothing%>">SwfOnMovieEndNothing</option>
        <option value="<%=SwfOnMovieEndSTOP%>">SwfOnMovieEndSTOP</option>
        <option value="<%=SwfOnMovieEndLoop%>">SwfOnMovieEndLoop</option>
        <option value="<%=SwfOnMovieEndUnload%>">SwfOnMovieEndUnload</option>
        <option value="<%=SwfOnMovieEndLoadMovie%>">SwfOnMovieEndLoadMovie</option>
      </select>
    </td>
  </tr>

  <tr>
    <th>FE2_SWF_ON_END_URL</th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ON_END_URL'></td>
  </tr>
</table>
<!-- END - SWF muxer parameters -->

<!-- WEBM muxer parameters -->
<table id='webmtable' class='hidden'>
  <tr>
    <th>(FE2_MUXER_WEBM defines no parameters)</th>
  </tr>
</table>
<!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM##### -->
<hr>
<p>
  <input type="submit" value="Start Encode">
  <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>

```

6.1.2 process_sample.asp

```

<%
'=====
'
' Copyright (c) On2 Technologies Inc. All Rights Reserved.

```

```

'
'-----
'   File:           $Workfile$
'                   $Revision$
'
'   Last Update: $DateUTC$
'-----

'process_sample.asp
' Receive a form via post from flix2_sample.asp, treating each name=value pair
' as a function/param pair.
' These map to the Flix Engine COM API and each function that has a valid
' param will be called.
' Once setup is complete, calls encode() to produce an output file located in out
  dir

Option Explicit

'Global Variables
Dim flix, codec, filter, muxer
Dim prefix, outdir

prefix = "C:\Inetpub\"
outdir = prefix & "flixmedia\out\"

' Increase script timeout to allow encodes of long files to complete,
' barring any further server side timeout settings
Server.ScriptTimeout = 43200 ' = 12 hour timeout
Response.Buffer = False

Response.Write(vbLf & vbLf _
    & "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" _
    & "http://www.w3.org/TR/html4/strict.dtd">" & vbLf _
    & "<html><head><title>Flix CGI Process Sample - ASP</title>" & vb
    Lf _
    & "<meta http-equiv='Content-Type' content='text/html; charset
    =iso-8859-1'">" _
    & "</head><body>" & vbLf)
Response.Write("<hr><p>process_sample.asp version 1.2<br>" & vbLf)

LoadEngine()

On Error Resume Next
' verify outdir's presence and accessibility
Dim objFSO, tempFile
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set tempFile = objFSO.CreateTextFile(outdir & "temp.dat", True)
If Not IsObject(tempFile) Or Not objFSO.FileExists(outdir & "temp.dat") Then
    Response.Write("<p>*****<br>" & vbLf)
    Response.Write("<b>WARNING</b>: <i>' " & outdir & "'</i>" _
        & " MUST exist and be writeable by" _
        & " <i>flixengine_com.dll</i>.<br>" & vbLf)

    Response.Write("Please make <i>' " & outdir & "'</i> accessible or modify" _
        & " the <tt>outdir</tt> value defined in '" _
        & Server.MapPath(Request.ServerVariables("SCRIPT_NAME")) _
        & "'<br>" & vbLf)
    Response.Write("*****</p>" & vbLf)
End If
tempFile.Close
'Attempt to delete the temp file, if possible
objFSO.DeleteFile(outdir & "temp.dat")
On Error GoTo 0

ParseParams()

```

```

Encode()
Finish(True)
'End of Main Code Block

'Prints the status codes of the Flix Engine object
Sub PrintEncoderStatus()
    Dim flixerr, syserr
    flixerr = flix.flixerrno()
    syserr = flix.syserrno()
    Response.Write("<p>Encoder Status<br>" & vbCrLf)
    Response.Write("&nbsp;  flix.getEncoderState: " & flix.getEncoderState() & "<br>" & vbCrLf)
    Response.Write("&nbsp;  flix.errno_: flixerrno:" & flixerr _
        & " syserrno:" & syserr & "</p>" & vbCrLf)
End Sub

'Adds the Flix Engine function and return value to the HTML table
'and, if necessary, exits with an error.
Sub ProcessHR(funcname, sc)
    Dim flixerr, syserr, flixsc, hr
    hr = Err.number
    flixsc = flix.sc
    flixerr = flix.flixerrno
    syserr = flix.syserrno

    Response.Write("<td align=""center"">" & sc & "</td>" & vbCrLf)

    If ((hr < 0) Or (sc > 0)) Then
        Response.Write("</tr></table><br>" & vbCrLf)
        Response.Write("*** Error calling " & funcname & ", sc = " & sc _
            & ", hr = " & hr & "; " & "flix.sc = " & flixsc _
            & " flix.errno_( " & flixerr & ", " & syserr & " )" _
            & vbCrLf)
        Finish(False)
    End If
    Response.Write("</tr>" & vbCrLf)
End Sub

'Parses parameters which have been passed as form items from flix2_sample.asp
Sub ParseParams()
    Response.Write("<table border='1' cellpadding='5'>" _
        & "<caption>Flix Function Calls</caption>" & vbCrLf _
        & "<tr><th>Function Name</th><th>Return Value</th></tr>" _
        & vbCrLf)

    Dim x
    Dim name, value
    For x = 1 to Request.Form.Count
        name = Request.Form.Key(x)
        If (Len( CStr( Request.Form.Item(name))) > 0) Then
            value = Request.Form.Item(name)
            'Response.Write(name & " = " & value & "<br>" & vbCrLf)
            If (name = "setOutputFile") Then
                Call SimpleSet(name, outdir & value)
            ElseIf ((name = "vcodec:") Or (name = "acodec:")) Then
                Call InitCodec(value)
            ElseIf ((InStr(1, value, "afilter:") > 0) _
                Or (InStr(1, value, "vfilter:") > 0)) Then
                Call InitFilter(name)
            ElseIf (name = "muxer:") Then
                Call InitMuxer(value)
            Else
                Dim temp
                temp = Split(name, ":", -1)
                If (temp(0) = "codec") Then
                    Call CodecInterface(temp(1), temp(2), value)
                ElseIf (temp(0) = "filter") Then

```

```

        Call FilterInterface(temp(1), temp(2), value)
    ElseIf (temp(0) = "muxer") Then
        Call MuxerInterface(temp(1), temp(2), value)
    Else
        Call SimpleSet(name, value)
    End If
End If
End If
Next 'Item in Form
End Sub

'Displays error information and exits.
'To add error information to the HTML table, use Sub ProcessHR.
Sub ReportFlixError(funcName)
    Response.Write("A Flix Engine error occurred while executing " & funcName &
        ".<br>" & vbCrLf)
    Response.Write("hr: " & Err.number & " flixerrno: " & flix.flixerrno _
        & " syserrno: " & flix.syserrno & "<br>" & vbCrLf)
    Err.Clear
    Finish(False)
End Sub

'Creates the Flix Engine Object and displays information about the object
Sub LoadEngine()
    On Error Resume Next

    '
    ' retrieve the main engine interface, IFlix
    '
    Set flix = Server.CreateObject("On2.FlixEngine")
    If ((IsObject(flix) = False) Or (Err.number < 0)) Then
        Response.Write("<br>Server.CreateObject ("&"On2.FlixEngine"&") failed!<br>" &
            vbCrLf & vbCrLf)
        Response.Write("hr: " & Hex(Err.number) & "<br>" & vbCrLf)
        Dim srv
        Set srv = Request.ServerVariables("SERVER_SOFTWARE")
        If Err.number = &H80040154 And _
            InStr(srv, "IIS") And Mid(srv, InStrRev(srv, "/")+1) >= 5.0 Then
            ' the associated description is valid only under IIS<5.0 as it
            ' references an obsolete metabase property:
            ' http://support.microsoft.com/kb/233968
            Response.Write("description: This error indicates IIS is configured " &
                _
                "to run in 64-bit mode and is therefore unable to run " &
                _
                "32-bit applications.<br>See <a href=" & _
                "http://support.microsoft.com/kb/895976">" & _
                "http://support.microsoft.com/kb/895976</a>" & _
                " for further details." & vbCrLf)
        Else
            Response.Write("description: " & Err.description & vbCrLf)
        End If
        Response.End
    End If

    'Print Flix Version information
    Response.Write("Flix Engine COM library. Flix Engine v" & flix.version() & vbCrLf)
    If (flix.flixerrno <> 0) Then
        ReportFlixError("flix.version()")
    End If
    Response.Write(" COM v" & flix.com_version() & "<br>" & vbCrLf & vbCrLf)
    If (flix.flixerrno <> 0) Then
        ReportFlixError("flix.com_version()")
    End If
    Response.Write(Replace(flix.copyright(), vbCrLf, "<br>" & vbCrLf) & "<br>" & vbCrLf
        & vbCrLf)

```

```

    If (flix.flixerrno <> 0) Then
        ReportFlixError("flix.copyright()")
    End If
End Sub

'Finishes the HTML page and exits the application
'If printStatus = True, the Flix encoder status will be displayed.
Sub Finish(printStatus)
    'Release all references to "Set" global objects
    Set codec = Nothing
    Set filter = Nothing

    If IsObject(flix) Then
        If (printStatus) Then
            PrintEncoderStatus()
        End If
        If (flix.isEncoderRunning = 1) Then
            flix.stopEncoding
            If (flix.flixerrno <> 0) Then
                Response.Write("flix.stopEncoding Error: " & flix.flixerrno() & "
<br>" & vbCrLf)
            ElseIf (flix.isEncoderRunning = 1) Then
                Response.Write("Flix failed to stop encoding, but did not report
an error!<br>" & vbCrLf)
            End If
        End If
        Set flix = Nothing
    End If

    If Not (flix Is Nothing) Then
        Response.Write("Setting flix = Nothing failed!<BR>")
    End If

    'Close the html page
    Response.Write("</body></html>" & vbCrLf)
    Response.End
End Sub

'Calls an IFlix function with a single parameter
Sub SimpleSet(func, arg)
    On Error Resume Next

    Response.Write("<tr><td>flix." & func & "(" & arg & " )</td>" & vbCrLf)
    Execute("flix." & func & "(arg)")
    Call ProcessHR("flix." & func, flix.sc)
End Sub

'Adds a codec, creating an IFlixPlgn object.
Sub InitCodec(name)
    On Error Resume Next

    'if name is a codec name, e.g. FE2_CODEEC_VP6, add an instance
    'we'll assume all IFlixPlgn::setParam's relate to this codec until we
    'hit the next codec name
    Response.Write("<tr><td>flix.addCodec( " & name & " )</td>" & vbCrLf)
    Set codec = flix.addCodec(Eval("flix." & name))
    Call ProcessHR("flix.addCodec", flix.sc)
End Sub

'Calls a codec-related IFlixPlgn function.
'Assumes that a codec has been added in InitCodec.
Sub CodecInterface(funcname, name, value)
    On Error Resume Next

    Response.Write("<tr><td>codec." & funcname & "(" & name & ", " & _
        & value & " )</td>" & vbCrLf)
    Execute("Call codec." & funcname & "(flix." & name & ", value)")

```



```

    Call ProcessHR("codec." & funcname, codec.sc)
End Sub

'Adds a filter, creating an IFlixPlgn object.
Sub InitFilter(name)
    On Error Resume Next

    'if name is a filter name, e.g. FE2_FILTER_CUT, add an instance
    'we'll assume all IFlixPlgn::setParam's relate to this filter until we
    'hit the next filter name
    Response.Write("<tr><td>flix.addFilter( " & name & " )</td>" & vbLf)
    Set filter = flix.addFilter(Eval("flix." & name))
    Call ProcessHR("flix.addFilter", flix.sc)
End Sub

'Calls a filter-related IFlixPlgn function.
'Assumes that a filter has been added in InitFilter.
Sub FilterInterface(funcname, name, value)
    On Error Resume Next

    Response.Write("<tr><td>filter." & funcname & "( " & name & ", " _
        & value & " )</td>" & vbLf)
    Execute("Call filter." & funcname & "(flix." & name & ", value)")
    Call ProcessHR("filter." & funcname, filter.sc)
End Sub

'Adds a muxer, creating an IFlixPlgn object.
Sub InitMuxer(name)
    On Error Resume Next

    'if name is a muxer name, e.g. FE2_MUXER_FLV, add an instance
    'we'll assume all IFlixPlgn::setParam's relate to this muxer until we
    'hit the next muxer name
    Response.Write("<tr><td>flix.addMuxer( " & name & " )</td>" & vbLf)
    Set muxer = flix.addMuxer(Eval("flix." & name))
    Call ProcessHR("flix.addMuxer", flix.sc)
End Sub

'Calls a muxer-related IFlixPlgn function.
'Assumes that a muxer has been added in InitMuxer.
Sub MuxerInterface(funcname, name, value)
    On Error Resume Next

    Response.Write("<tr><td>muxer." & funcname & "( " & name & ", " _
        & value & " )</td>" & vbLf)
    Execute("Call muxer." & funcname & "(flix." & name & ", value)")
    Call ProcessHR("muxer." & funcname, muxer.sc)
End Sub

'Starts the encode and, using a simple loop, prints encoding updates.
Sub Encode
    On Error Resume Next

    '
    ' start the encode
    '
    Response.Write("<tr><td>flix.encode( )</td>" & vbLf)
    flix.encode
    Call ProcessHR("flix.encode", flix.sc)

    '
    ' retrieve the encoding status interface, IEncodingStatus
    '
    Response.Write("<tr><td>flix.encodingStatus( )</td>" & vbLf)
    Dim encstatus
    set encstatus = flix.encodingStatus()
    Call ProcessHR("flix.encodingStatus", 0)

```

```

Response.Write("</table>" & vbCrLf)

Response.Write("<p>Encoding...(video frames encoded, percent complete). " _
               & "Total frames will reset when doing 2pass.<br>" & vbCrLf)

'Start the progress updates
'NOTE: The progress updates in this sample are implemented with a
'primitive timer loop, which is CPU intensive. An alternate solution
'would be recommended for production purposes, probably using a sleep
'function or timer object.
Dim isRunning
Dim seconds
Dim startTime
seconds = 0
startTime = Timer
Do
    'Correct times for encodes that traverse one midnight
    If (startTime > Timer) Then startTime = startTime - (60 * 60 * 24)
    If ((Timer - startTime) >= seconds) Then
        isRunning = flx.isEncoderRunning
        Response.Write("(" & encStatus.totalFrames)
        Response.Write(", " & encStatus.percentComplete & "%)<br>" & vbCrLf)
        seconds = seconds + 1 'update every 1 second
    End If
    Loop Until ((isRunning = 0) or (Response.IsClientConnected = False))
    Response.Write("<br>Done!</p>" & vbCrLf)
End Sub
%>

```

6.2 ASP.NET

The ASP .NET example requires that the Microsoft .NET Framework (1.1|2.0) be installed. Should you receive an error dialog containing the text "The application failed to initialize ...", it is likely that you do not have the Framework installed. This component is located on Windows Update under "Software, Optional".

6.3 CGI

This example consists of 2 parts: [flix2_sample.aspx](#) and [process_sample.aspx](#). [flix2_sample.aspx](#) searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's COM interfaces.

The ASP .NET example requires that the Microsoft .NET Framework (1.1 or higher) be installed. The .NET Framework component is located on Windows Update under "Software, Optional".

This example also requires that the server process user account has any necessary access to the input and output directories. Therefore, you may need to give the ASPNET account write access to the output directory.

To install this example on your web server, copy the file `<InstallDir>\sample\asp.net\bin\Interop.flixengine_com.dll` to your `<WebServerRoot>\bin` directory, which is usually `C:\Inetpub\wwwroot\bin`. Then copy the `<InstallDir>\sample\asp.net` directory to your web server root.

To use this example navigate to [flix2_sample.aspx](#) in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to [process_sample.aspx](#) which runs the encode loop.

Default file locations used by the scripts:

Input : *C:/Inetpub/flixmedia/in* (\$indir in [flix2_sample.aspx](#))
Overlay : *C:/Inetpub/flixmedia/overlay* (\$overlaydir in [flix2_sample.aspx](#))
Output : *C:/Inetpub/flixmedia/out* (\$outdir in [process_sample.aspx](#))

6.3.1 flix2_sample.aspx

```
<!--
'=====
'
' Copyright (c) On2 Technologies Inc. All Rights Reserved.
'
'-----
'
' File:           $Workfile$
'               $Revision$
'
' Last Update: $DateUTC$
'
'-----
-->
<%@ Page Language="VB" Strict = "true"%>
<%@ Import Namespace=System.IO %>
<%@ Assembly Name="Interop.flixengine_com"%>
<%@ Import Namespace="flixengine_com" %>
<%@ Import Namespace="flixengine_com.FE2_AudioBitrates" %>
<%@ Import Namespace="flixengine_com.FE2_AudioSamplingrates" %>
<%@ Import Namespace="flixengine_com.FE2_CompressMode" %>
<%@ Import Namespace="flixengine_com.FE2_CuePointType" %>
<%@ Import Namespace="flixengine_com.FE2_EmbeddedUrlType" %>
<%@ Import Namespace="flixengine_com.FE2_ExportedVideoType" %>
<%@ Import Namespace="flixengine_com.FE2_OverlayPositionMode" %>
<%@ Import Namespace="flixengine_com.FE2_PNGExCuePtMode" %>
<%@ Import Namespace="flixengine_com.FE2_SwfOnEndOptions" %>
<%@ Import Namespace="flixengine_com.FE2_SwfOnStartOptions" %>
<%@ Import Namespace="flixengine_com.FE2_SwfPreloaderOptions" %>
<%@ Import Namespace="flixengine_com.FE2_VideoBitrateControls" %>
<%@ Import Namespace="flixengine_com.FE2_VideoKeyframeTypes" %>
<%@ Import Namespace="flixengine_com._on2bool" %>
<%@ Import Namespace="flixengine_com.blurfilter" %>
<%@ Import Namespace="flixengine_com.deinterlacemode" %>
<%@ Import Namespace="flixengine_com.flv_metadata" %>
<%@ Import Namespace="flixengine_com.h264profile" %>
<%@ Import Namespace="flixengine_com.lame_rcmode" %>
<%@ Import Namespace="flixengine_com.masksiz" %>
<%@ Import Namespace="flixengine_com.vp6profile" %>

<script runat="server">
    Dim prefix      As String = "C:\Inetpub\"
    Dim indir       As String = prefix & "flixmedia\in\"
    Dim overlaydir  As String = prefix & "flixmedia\overlay\"
</script>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/str
ict.dtd">
<html lang="en-US">
<head>
<title>Flix2 CGI Sample - ASP.NET</title>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<script type="text/javascript">
function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
        i++;
    }
}

function toggle_expand(_this)
{
    _this.parentNode.className= (_this.parentNode.className=='expanded') ?
        'collapsed' : 'expanded';
    document.getElementById('showall').checked=false;
}

function set_table_visible(ctable,visible)
{
    var table= document.getElementById(ctable);

    if(table) {
        if(visible && table.className == 'hidden') {
            table.className= '';
        } else if(!visible && table.className == '') {
            table.className= 'hidden';
        }
    }
}

function hide_tablelist(vistablename,tablelist)
{
    var table= document.getElementById(vistablename);

    if(table&&table.className=='hidden') {
        table.className= '';

        var i=0;
        while(tablelist[i]) {
            if(tablelist[i]!=vistablename) {
                var hiddentable= document.getElementById(tablelist[i]);

                set_table_visible(tablelist[i],false);
                /*clear down any values set in the hidden table to avoid posting
                unnecessary values*/
                clear_value(hiddentable.getElementsByTagName('input'));
                clear_value(hiddentable.getElementsByTagName('select'));
            }
            i++;
        }
    }
}

function set_acodec_visible(ctable)
{
    var acodecs= new Array('aactable','aacplustable','amrnhtable','lametable','pcm
        table','vorbistable');
    hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
    var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8tabl
        e');
    hide_tablelist(ctable,vcodecs);
}

```

```

}

function set_muxer_visible(mtable)
{
    var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','
        tg2table','tgptable','webmtable');
    hide_tablelist(mtable,muxers);
}

function clear_value(list)
{
    var i=0;
    while(list[i]) {
        if(list[i].type=='checkbox') { list[i++].checked=false; }
        else { list[i++].value=''; }
    }
}

function toggle_ftable(ftable,enabled)
{
    var table= document.getElementById(ftable);

    if(table) {
        table.className = enabled ? '' : 'disabled';
        if(!enabled) {
            clear_value(table.getElementsByTagName('input'));
            clear_value(table.getElementsByTagName('select'));
        }
    }
}

function reset_tables()
{
    var table_list= document.getElementsByTagName('table');
    var i=0;
    while(table_list[i]) {
        if (table_list[i].id.length > 7 &&
            table_list[i].id.substring(0,7) == 'filter_') {
            table_list[i].className= 'disabled';
        } else if (table_list[i].id.indexOf('table') != -1) {
            table_list[i].className= 'hidden';
        }
        i++;
    }
}

</script>

<style type="text/css">
<!--
html {
    font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
    font-size: 100%;
    color: rgb(56,56,56);
    background-color: rgb(236,236,236);
    border-style: solid;
    border-color: rgb(236,236,236);
}

body {
    text-align: center;
    margin: 0 auto;
}

div.content {
    color: rgb(56,56,56);
    background-color: rgb(246,246,246);
}

```

```

        text-align: left;
        margin: 0 auto;
        width: 80%;
        min-width: 768px;
        max-width: 932px;
        border-width: 0 1px;
        border-color: rgb(144,144,144);
        border-style: solid;
    }

    div.content:after {
        content: "";
        color: inherit;
        background-color: rgb(250,250,250);
        border-top: 1px solid rgb(144,144,144);
        height: 20px;
        width: 100%;
        display: block;
    }

    fieldset table, fieldset {display: none;}
    fieldset.expanded, fieldset.collapsed {display: block;}

    /*first is fallback for IE*/
    fieldset.expanded table {display: block;}
    fieldset.expanded table {display: table;}

    h1 {
        font-family: sans-serif;
        font-size: 150%;
        font-weight: normal;
        text-align: left;
        letter-spacing: -1px;
        color: rgb(74,74,74);
        background-color: inherit;
        margin: 0;
    }

    a {
        color: rgb(74,74,74);
        background-color: transparent;
    }

    label {
        font-size: 75%;
    }

    fieldset {
        font-size: 75%;
        line-height: 130%;
        padding: 0;
        margin: 20px;
        border: none;
    }

    fieldset.expanded {
        color: inherit;
        background-color: rgb(252,252,252);
        border-style: solid;
        border-width: 1px;
        border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
    }

    legend {
        padding: 0 5px;
        border-left: 10px solid rgb(217,217,217);
        cursor: pointer;
    }

```

```

}
legend:hover {text-decoration: underline;}
fieldset.expanded legend {
    font-size: 150%;
    font-weight: bold;
    letter-spacing: -1px;
    background: transparent;
    margin-left: 12px;
    border-right: 10px solid rgb(217,217,217);
    display: block;
}
table {
    font-size: 100%;
    border-spacing: 0;
    /*border-collapse: collapse;*/
    width: 100%;
}

th:before { display: none; }
th, td {
    width: 50%;
    vertical-align: top;
    padding: 2px 3px;
    border-width: 1px 0;
    border-style: solid;
    border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
}
tr:first-child>* {border-top-color: rgb(252,252,252);}
th {
    font-weight: normal;
    text-align: left;
    padding: 2px 2px 2px 5px;
}
input[type] {
    font-family: monospace;
    font-size: 100%;
    color: rgb(56,56,56);
    background-color: inherit;
}
[type="text"], [type="number"] {
    margin-right: 13px;
    width: 222px;
}

[type="checkbox"].filter {
    margin-left: 4px
}
[type="checkbox"] {
    margin-left: 13px
}
[type="button"], [type="submit"] {
    font-size: 1em;
    margin: 0 2px 0 13px;
}

fieldset {
    font-size: 75%;
    margin: 20px 10px;
}
fieldset input[type="text"], fieldset input[type="number"] {
    width: 95% !important;
    margin: 0;
    display: block;
}
fieldset.expanded table.hidden {
    display: none;
}

```

```

        table.disabled {
            color: rgb(176,176,176);
        }
    -->
</style>
</head>

<body>
    <div class='content'>
        <noscript>
            <p>This page requires javascript be enabled.</p>
        </noscript>

        <hr>
        <h1>Flix CGI Sample</h1>

        <p><small>flix2_sample.cgi version 1.9</small></p>
        <h4>Instructions</h4>
        <ul>
            <li>In this sample you must choose a source file and an output file.<br>
            If you leave all the other options blank then the sample will not call
            the corresponding Flix Engine function and the default will be used.<br>
            When done please press the "Start Encode" button at the bottom of the page.<br>
            <li>Mouse over a function name to see its default, if applicable.
            <li>Current source file directory: <%=indir%>
        </ul>

        <p>
            <label><input type="checkbox" id='showall' onclick='showall(this.checked)'/>Show
            all</label>
        </p>

        <form action="process_sample.aspx" method="post">

        <!-- ##SOURCE FILE##### -->

        <hr>
        <fieldset class='expanded' id="srcfile">
        <legend onclick='toggle_expand(this)'/>Source File</legend>
        <table>

        <tr>
            <td>
                <%
                    'Get the file names from indir
                    Dim inDirectory As DirectoryInfo = New DirectoryInfo(indir)
                    If (inDirectory.Exists) Then
                        'Code snippet for sorted directory listing
                        Dim fileInfos() As FileInfo = inDirectory.GetFiles()
                        Dim buffer As FileInfo
                        Dim fileOne, fileTwo As Integer
                        For fileOne = 0 To UBound(fileInfos)
                            For fileTwo = (fileOne + 1) To UBound(fileInfos)
                                If strComp(fileInfos(fileOne).Name,fileInfos(fileTwo).Name,0)=1 Then
                                    buffer = fileInfos(fileTwo)
                                    fileInfos(fileTwo) = fileInfos(fileOne)
                                    fileInfos(fileOne) = buffer
                                end if
                            Next
                        Next

                        Dim siz As Integer = Ubound(fileInfos)+1
                        If (siz>15) Then siz= 15

```



```

'Output the sorted options
Response.Write("<select name='setInputFile' size='" & siz & "'>" & vbCrLf)
Dim selected As Boolean = False
Dim inFile As FileInfo
For Each inFile In fileInfos
    If (Len(inFile.Name) > 0) Then
        Response.Write("<option ")
        If (selected = False) Then
            Response.Write("selected ")
            selected = True
        End If
        Response.Write("value=""" & indir & inFile.Name & """">" _
            & inFile.Name & "</option>" & vbCrLf)
    End If
Next
Response.Write("</select>" & vbCrLf)
Else
    Response.Write("WARNING couldn't open " & indir _
        & ": FolderExists() returned FALSE<br>" & vbCrLf)
End If
%>

</td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE#####
-->
<hr>
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)''>Output File</legend>
<table>

<tr>
<td>
<input type="text" name="setOutputFile" value="cgi-aspnet-out.flv">
</td>
</tr>

</table>
</fieldset>

<!-- ##MAIN OPTIONS#####
### -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)''>Main Options</legend>
<table>

<tr>
<th><abbr title="Default: FALSE">setOverwriteExistingFiles</abbr></th>
<td>
<select name="setOverwriteExistingFiles">
<option value=""></option>
<option value="<%=on2true%>">TRUE</option>
<option value="<%=on2false%>">FALSE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: TRUE">setExportAudio</abbr></th>
<td>
<select name="setExportAudio">
<option value=""></option>

```

```

<option value="<%=on2true%>">TRUE</option>
<option value="<%=on2false%>">FALSE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: TRUE">setExportVideo</abbr></th>
<td>
<select name="setExportVideo">
<option value=""></option>
<option value="<%=on2true%>">TRUE</option>
<option value="<%=on2false%>">FALSE</option>
</select>
</td>
</tr>

</table>
</fieldset>

<!-- ##CODECS#####
    ## -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this) '>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
<td>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
onfocus="set_vcodec_visible('vp8table') ">
<abbr title="For use with WebM">FE2_CODEC_VP8</abbr>&nbsp;
</label>
<br>

<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
onfocus="set_vcodec_visible('vp6table') ">
<abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr>&nbsp;
</label>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
onfocus="set_vcodec_visible('vp6atable') ">
<abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr>&nbsp;
</label>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_H263'
onfocus="set_vcodec_visible('h263table') ">
<abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr>&nbsp;
</label>
<br>

<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
onfocus="set_vcodec_visible('h263table') ">
<abbr title="For use with 3GP">FE2_CODEC_H263_BASELINE</abbr>&nbsp;
</label>
<label>
<input type='radio' name='vcodec:' value='FE2_CODEC_H264'
onfocus="set_vcodec_visible('h264table') ">
<abbr title="For use with 3GP/3G2/MOV/MP4">FE2_CODEC_H264</abbr>&nbsp;
</label>

<!-- VP6 codec parameters -->
<table id='vp6table' class='hidden'>

```

```

<tr>
<th><abbr title="Default: 448kbps">FE2_VP6_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6_KFINTTYPE</abbr></th>
<td>
<select name="codec:setParam:FE2_VP6_KFINTTYPE">
<option value=""></option>
<option value="<%=MAX_KEYFRAMES%>">MAX_KEYFRAMES</option>
<option value="<%=FIXED_KEYFRAMES%>">FIXED_KEYFRAMES</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
FE2_VP6_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6_KFFREQ'></td>
</tr>

<tr>
<th><abbr title="Default: VBR_2PASSControl">FE2_VP6_RC_MODE</abbr></th>
<td>
<select name="codec:setParam:FE2_VP6_RC_MODE">
<option value=""></option>
<option value="<%=VBR_2PASSControl%>">VBR_2PASSControl</option>
<option value="<%=CBR_2PASSControl%>">CBR_2PASSControl</option>
<option value="<%=VBR_1PASSControl%>">VBR_1PASSControl</option>
<option value="<%=CBR_1PASSControl%>">CBR_1PASSControl</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6_CXMODE</abbr></th>
<td>
<select name="codec:setParam:FE2_VP6_CXMODE">
<option value=""></option>
<option value="<%=COMPRESSMODE_GOOD%>">COMPRESSMODE_GOOD</option>
<option value="<%=COMPRESSMODE_BEST%>">COMPRESSMODE_BEST</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: VP6_E">FE2_VP6_PROFILE</abbr></th>
<td>
<select name="codec:setParam:FE2_VP6_PROFILE">
<option value=""></option>
<option value="<%=VP6_E%>">VP6_E</option>
<option value="<%=VP6_S%>">VP6_S</option>
</select>
</td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
<th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6_CONCURRENCY'></td>
</tr>

<tr>
<th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6_UNDERSHOOT_PCT'></td>

```

```

</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_TEMPORAL_RESAMPLING</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_TEMPORAL_DOWN_WATERMARK</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
  <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_2PASS_MIN_SECTION'></td>
</tr>

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6_2PASS_MAX_SECTION'></td>   <abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_BITRATE'></td>   <abbr title="Default: 68kbps (15% of default 448kbps)"> FE2_VP6A_ALPHA_BITRATE</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_ALPHA_BITRATE'></td>   <abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>  <select name="codec:setParam:FE2_VP6A_KFINTTYPE"> <option value=""></option> <option value="<%=MAX_KEYFRAMES%>">MAX_KEYFRAMES</option> <option value="<%=FIXED_KEYFRAMES%>">FIXED_KEYFRAMES</option> </select> | <abbr title="Default: 12*fps or 360 frames if the framerate is unknown"> FE2_VP6A_KFFREQ</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_KFFREQ'></td>   <abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>  <select name="codec:setParam:FE2_VP6A_RC_MODE"> <option value=""></option> <option value="<%=VBR_2PASSControl%>">VBR_2PASSControl</option> <option value="<%=CBR_2PASSControl%>">CBR_2PASSControl</option> <option value="<%=VBR_1PASSControl%>">VBR_1PASSControl</option> <option value="<%=CBR_1PASSControl%>">CBR_1PASSControl</option> </select> | <abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>  <select name="codec:setParam:FE2_VP6A_CXMODE"> <option value=""></option> <option value="<%=COMPRESSMODE_GOOD%>">COMPRESSMODE_GOOD</option> <option value="<%=COMPRESSMODE_BEST%>">COMPRESSMODE_BEST</option> </select> | <abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th> | | | | | | | | | | |

```

```

        <td><input type='text' name='codec:setParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_MIN_Q</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_MIN_Q'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_ALPHA_MIN_Q</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MIN_Q'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_MAX_Q</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_MAX_Q'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_MAX_Q</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MAX_Q'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_SHARPNESS</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_SHARPNESS'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_NOISE_REDUCTION</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_NOISE_REDUCTION'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_TEMPORAL_RESAMPLING</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
    </tr>

    <tr>
        <th><abbr title="Default: bits per pixel dependent (see API docs)">
            FE2_VP6A_TEMPORAL_DOWN_WATERMARK</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'>
        </td>
    </tr>

    <tr>

```

```

    <th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PREBUFFER'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr>
    </th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>
  </tr>

  <tr>
    <th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MIN_SECTION'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MAX_SECTION'></td>
  </tr>
</table>
<!-- END - VP6A codec parameters -->

<!-- H263 codec parameters -->
<table id='h263table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H263_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_H263_KFINTTYPE">
        <option value=""></option>
        <option value="<%=MAX_KEYFRAMES%">MAX_KEYFRAMES</option>
        <option value="<%=FIXED_KEYFRAMES%">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_H263_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H263_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_H263_RC_MODE">
        <option value=""></option>
        <option value="<%=VBR_2PASSControl%">VBR_2PASSControl</option>
        <option value="<%=CBR_2PASSControl%">CBR_2PASSControl</option>
        <option value="<%=VBR_1PASSControl%">VBR_1PASSControl</option>
      </select>
    </td>
  </tr>
</table>

```

```

        <option value="%=CBR_1PASSControl%">CBR_1PASSControl</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MAX_Q'></td>
</tr>

<tr>
<th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MIN_Q'></td>
</tr>

</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
<tr>
<th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
FE2_H264_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_KFFREQ'></td>
</tr>

<tr>
<th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
<td>
<select name="codec:setParam:FE2_H264_RC_MODE">
<option value=""></option>
<option value="%=VBR_1PASSControl%">VBR_1PASSControl</option>
<option value="%=CBR_1PASSControl%">CBR_1PASSControl</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
<td>
<select name="codec:setParam:FE2_H264_PROFILE">
<option value=""></option>
<option value="%=BASE_H264PROFILE%">BASE_H264PROFILE</option>
<option value="%=MAIN_H264PROFILE%">MAIN_H264PROFILE</option>
<option value="%=HIGH_H264PROFILE%">HIGH_H264PROFILE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_B_FRAME_RATE'></td>
</tr>

<tr>
<th><abbr title="Default: Dependent on profile selection, see API docs. Valid
Range [0,5]">FE2_H264_SPEED</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_SPEED'></td>
</tr>

</table>
<!-- END - H264 codec parameters -->

```



```

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_KFINTTYPE">
        <option value=""></option>
        <option value="<%=MAX_KEYFRAMES%>">MAX_KEYFRAMES</option>
        <option value="<%=FIXED_KEYFRAMES%>">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_VP8_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_RC_MODE">
        <option value=""></option>
        <option value="<%=VBR_2PASSControl%>">VBR_2PASSControl</option>
        <option value="<%=CBR_2PASSControl%>">CBR_2PASSControl</option>
        <option value="<%=VBR_1PASSControl%>">VBR_1PASSControl</option>
        <option value="<%=CBR_1PASSControl%>">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_CXMODE">
        <option value=""></option>
        <option value="<%=COMPRESSMODE_GOOD%>">COMPRESSMODE_GOOD</option>
        <option value="<%=COMPRESSMODE_BEST%>">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_THREADS'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_PROFILE'></td>
  </tr>

  <tr><th><b>Advanced Settings:</b></th></tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_LAG'></td>
  </tr>

  <tr>

```

```

    <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_UNDERSHOOT_PCT'></td>
</tr>

<tr>
    <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_OVERSHOOT_PCT'></td>
</tr>

<tr>
    <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_MIN_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
    <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
    <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
    <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
    <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
    <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
    <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>
    <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

<tr>
    <th><abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_ALTREF'></td>
</tr>

<tr>

```

```

        <th><abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP8_AR_MAX_FRAMES'></td>
    </tr>

    <tr>
        <th><abbr title="">FE2_VP8_AR_TYPE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP8_AR_TYPE'></td>
    </tr>

    <tr>
        <th><abbr title="">FE2_VP8_AR_STRENGTH</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP8_AR_STRENGTH'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP8_MB_STATIC_THRESHOLD'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP8_TOKEN_PARTITIONS'></td>
    </tr>
</table>
<!-- END - VP8 codec parameters -->

</td>
</tr> <!-- END - video codecs -->

<tr><th><b>Audio Codecs</b></th></tr>
<tr>
    <td>
        <label>
            <input type='radio' name='acodec:' value='FE2_CODEC_VORBIS'
                onfocus="set_acodec_visible('vorbistable') ">
            <abbr title="For use with WebM">FE2_CODEC_VORBIS</abbr>&nbsp;
        </label>
        <br>

        <label>
            <input type='radio' name='acodec:' value='FE2_CODEC_AAC'
                onfocus="set_acodec_visible('aactable') ">
            <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AAC</abbr>&nbsp;
        </label>
        <label>
            <input type='radio' name='acodec:' value='FE2_CODEC_AACPLUS'
                onfocus="set_acodec_visible('aacplustable') ">
            <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AACPLUS</abbr>&nbsp;
        </label>
        <label>
            <input type='radio' name='acodec:' value='FE2_CODEC_LAME'
                onfocus="set_acodec_visible('lametable') ">
            <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_LAME</abbr>&nbsp;
        </label>
        <label>
            <input type='radio' name='acodec:' value='FE2_CODEC_PCM'
                onfocus="set_acodec_visible('pcmtable') ">
            <abbr title="For use with FLV/SWF">FE2_CODEC_PCM</abbr>&nbsp;
        </label>
        <br>

        <label>
            <input type='radio' name='acodec:' value='FE2_CODEC_AMR_NB'
                onfocus="set_acodec_visible('amrntable') ">

```

```

    <abbr title="For use with 3GP">FE2_CODEC_AMR_NB</abbr>&nbsp;
  </label>

  <!-- AMR_NB codec parameters -->
  <table id='amrnhtable' class='hidden'>
    <tr>
      <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
      <td><input type='text' name='codec:setParam:FE2_AMR_BITRATE'></td>
    </tr>
  </table>
  <!-- END - AMR_NB codec parameters -->

  <!-- AAC codec parameters -->
  <table id='aactable' class='hidden'>
    <tr>
      <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
      <td><input type='text' name='codec:setParam:FE2_AAC_BITRATE'></td>
    </tr>
  </table>
  <!-- END - AAC codec parameters -->

  <!-- AACPLUS codec parameters -->
  <table id='aacplustable' class='hidden'>
    <tr>
      <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
      <td><input type='text' name='codec:setParam:FE2_AACPLUS_BITRATE'></td>
    </tr>

    <tr>
      <th><abbr title="Default: disabled (aacPlus v1)">
        FE2_AACPLUS_PARAMETRIC_STEREO</abbr></th>
      <td>
        <select name="codec:setParam:FE2_AACPLUS_PARAMETRIC_STEREO">
          <option value=""></option>
          <option value="0">disable (aacPlus v1)</option>
          <option value="1">enable (aacPlus v2)</option>
        </select>
      </td>
    </tr>
  </table>
  <!-- END - AACPLUS codec parameters -->

  <!-- LAME codec parameters -->
  <table id='lametable' class='hidden'>
    <tr>
      <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
      <td><input type='text' name='codec:setParam:FE2_LAME_BITRATE'></td>
    </tr>

    <tr>
      <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
      <td><input type='text' name='codec:setParam:FE2_LAME_QUALITY'></td>
    </tr>

    <tr>
      <th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
      <td>
        <select name="codec:setParam:FE2_LAME_RC_MODE">
          <option value=""></option>
          <option value="%=LAME_CBR%">LAME_CBR</option>
          <option value="%=LAME_ABR%">LAME_ABR</option>
          <option value="%=LAME_VBR_rh%">LAME_VBR_rh</option>
          <option value="%=LAME_VBR_mtrh%">LAME_VBR_mtrh</option>
        </select>
      </td>
    </tr>
  </table>

```

```

    </tr>
</table>
<!-- END - LAME codec parameters -->

<!-- PCM codec parameters -->
<table id='pcmtable' class='hidden'>
  <tr>
    <th>(FE2_CODEEC_PCM defines no parameters)</th>
  </tr>
</table>
<!-- END - PCM codec parameters -->

<!-- VORBIS codec parameters -->
<table id='vorbistable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VORBIS_BITRATE'></td>
  </tr>
</table>
<!-- END - VORBIS codec parameters -->

</td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS#####
    ## -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this)'>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_CUT' value='avfilter:'
        onchange="toggle_ftable('filter_cut',this.checked)">
      FE2_FILTER_CUT
    </label>

    <table id='filter_cut' class='disabled'>
      <tr>
        <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CUT_START_SEC'></td>
      </tr>

      <tr>
        <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CUT_STOP_SEC'></td>
      </tr>

      <tr>
        <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>
        <td>
          <select name="filter:setParam:FE2_CUT_USE_SEEK">
            <option value=""></option>
            <option value="<%=on2false%>">FALSE</option>
            <option value="<%=on2true%>">TRUE</option>
          </select>
        </td>
      </tr>
    </table>
  </td>
</tr>

```

```

</table>

</td>
</tr>
<!-- END - CUT filter parameters -->

<tr><th><b>Video Filters</b></th></tr>

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:'
onchange="toggle_ftable('filter_adaptive_deinterlace',this.checked)">
FE2_FILTER_ADAPTIVE_DEINTERLACE
</label>

<table id='filter_adaptive_deinterlace' class='disabled'>
<tr>
<th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</abbr></th>
<td>
<select name="filter:setParam:FE2_ADAPTIVE_DEINTERLACE_MODE">
<option value=""></option>
<option value="<%=DEINTERLACE_NONE%>">DEINTERLACE_NONE</option>
<option value="<%=DEINTERLACE_1_2_1_BLUR%>">DEINTERLACE_1_2_1_BLUR</option>
<option value="<%=DEINTERLACE_DROP_FIELD%>">DEINTERLACE_DROP_FIELD</option>
<option value="<%=DEINTERLACE_ADAPTIVE%>">DEINTERLACE_ADAPTIVE</option>
</select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - ADAPTIVE DEINTERLACE filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_BCHS' value='vfilter:'
onchange="toggle_ftable('filter_bchs',this.checked)">
FE2_FILTER_BCHS
</label>

<table id='filter_bchs' class='disabled'>
<tr>
<th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th>
<td><input type='text' name='filter:setParam:FE2_BCHS_BRIGHTNESS'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
<td><input type='text' name='filter:setParam:FE2_BCHS_CONTRAST'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
<td><input type='text' name='filter:setParam:FE2_BCHS_HUE'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
<td><input type='text' name='filter:setParam:FE2_BCHS_SATURATION'></td>
</tr>
</table>

</td>

```

```

</tr>
<!-- END - BCHS filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_BLUR' value='vfilter:'
onchange="toggle_ftable('filter_blur',this.checked)">
FE2_FILTER_BLUR
</label>

<table id='filter_blur' class='disabled'>
<tr>
<th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
<td>
<select name="filter:setParam:FE2_BLUR_FILTER">
<option value=""></option>
<option value="<%=BLUR_LOWPASS%>">BLUR_LOWPASS</option>
<option value="<%=BLUR_GAUSS%>">BLUR_GAUSS</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
<td>
<select name="filter:setParam:FE2_BLUR_MASKSIZE">
<option value=""></option>
<option value="<%=MASK_3x3%>">MASK_3x3</option>
<option value="<%=MASK_5x5%>">MASK_5x5</option>
</select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_CROP' value='vfilter:'
onchange="toggle_ftable('filter_crop',this.checked)">
FE2_FILTER_CROP
</label>

<table id='filter_crop' class='disabled'>
<tr>
<th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_TOP'></td>
</tr>

<tr>
<th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_BOTTOM'></td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_LEFT'></td>
</tr>

<tr>
<th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>

```

```

        <td><input type='text' name='filter:setParam:FE2_CROP_RIGHT'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - CROP filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
    name='FE2_FILTER_DENOISE' value='vfilter:'
    onchange="toggle_ftable('filter_denoise',this.checked)">
    FE2_FILTER_DENOISE
</label>

<table id='filter_denoise' class='disabled'>
<tr>
<th><abbr title="Default: 0. Range: [0.0,1.0]">FE2_DENOISE_NOISE_LEVEL</abbr>
</th>
<td><input type='text' name='filter:setParam:FE2_DENOISE_NOISE_LEVEL'></td>
</tr>
</table>

</td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
    name='FE2_FILTER_FRAMERATE' value='vfilter:'
    onchange="toggle_ftable('filter_framerate',this.checked)">
    FE2_FILTER_FRAMERATE
</label>

<table id='filter_framerate' class='disabled'>
<tr>
<th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAM
    ERATE_DECIMATE</abbr></th>
<td><input type='text' name='filter:setParam:FE2_FRAMERATE_DECIMATE'></td>
</tr>

<tr>
<th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FR
    AMERATE_FPS</abbr></th>
<td><input type='text' name='filter:setParam:FE2_FRAMERATE_FPS'></td>
</tr>
</table>

</td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
    name='FE2_FILTER_MIRROR' value='vfilter:'
    onchange="toggle_ftable('filter_mirror',this.checked)">
    FE2_FILTER_MIRROR
</label>

<table id='filter_mirror' class='disabled'>
<tr>

```



```

        <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
        <td>
            <select name="filter:setParam:FE2_MIRROR_HORIZONTAL">
                <option value=""></option>
                <option value="<%=on2false%>">FALSE</option>
                <option value="<%=on2true%>">TRUE</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
        <td>
            <select name="filter:setParam:FE2_MIRROR_VERTICAL">
                <option value=""></option>
                <option value="<%=on2false%>">FALSE</option>
                <option value="<%=on2true%>">TRUE</option>
            </select>
        </td>
    </tr>
</table>

</td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_OVERLAY' value='vfilter:'
                onchange="toggle_ftable('filter_overlay',this.checked)">
            FE2_FILTER_OVERLAY
        </label>

        <table id='filter_overlay' class='disabled'>
            <tr>
                <th>
                    <abbr title="Currently searching <%=overlaydir%> for overlay images">FE2_OVE
                        RLAY_FILE</abbr>
                </th>
                <td>
                    <%
'Get the file names from overlaydir
Dim overlayDirectory As DirectoryInfo = New DirectoryInfo(overlaydir)
If (overlayDirectory.Exists) Then
    'Code snippet for sorted directory listing
    Dim fileInfos() As FileInfo = overlayDirectory.GetFiles()
    Dim fileOne, fileTwo as Integer
    Dim buffer as FileInfo
    For fileOne = 1 To UBound(fileInfos)
        For fileTwo = (fileOne + 1) To UBound(fileInfos)
            If strComp(fileInfos(fileOne).Name,fileInfos(fileTwo).Name,0)=1 Then
                buffer = fileInfos(fileTwo)
                fileInfos(fileTwo) = fileInfos(fileOne)
                fileInfos(fileOne) = buffer
            end if
        Next
    Next

    'Output the sorted options
    Response.Write("<select name=""filter:setParamAsStr:FE2_OVERLAY_FILE"">" _
        & vbLf)
    Response.Write("<option value="" ""></option>" & vbLf)
    Dim overlayFile As FileInfo
    For Each overlayFile In fileInfos
        If ( (InStr(1, overlayFile.Name, ".bmp", CompareMethod.Text) > 0) _

```

```

        Or (InStr(1, overlayFile.Name, ".png", CompareMethod.Text) > 0)) Then

        Response.Write("<option value='" & overlaydir & overlayFile.Name _
                        & "'>" & overlayFile.Name & "</option>" & vbCrLf)
    End If
Next
Response.Write("</select>" & vbCrLf)
Response.Write("<br>default: None, must be programmatically set to" _
                & " absolute path of overlay input file, e.g., '" _
                & overlaydir & "overlay.png'" & vbCrLf)
Else
    Response.Write("WARNING couldn't open " & overlaydir _
                    & ": FolderExists() returned FALSE<br>" & vbCrLf)
End If
%>

</td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
<td>
<select name="filter:setParam:FE2_OVERLAY_MASK_XY">
<option value=""></option>
<option value="<%=on2false%>">FALSE</option>
<option value="<%=on2true%>">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_Y'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
<td>
<select name="filter:setParam:FE2_OVERLAY_MASK_RGB">
<option value=""></option>
<option value="<%=on2false%>">FALSE</option>
<option value="<%=on2true%>">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_R'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_G'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_B'></td>
</tr>

<tr>
<th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
<td>
<select name="filter:setParam:FE2_OVERLAY_POS">

```

```

        <option value=""></option>
        <option value="<%=FE2_OVERLAY_POS_MODE_TOPLEFT%>">
FE2_OVERLAY_POS_MODE_TOPLEFT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_BOTLEFT%>">
FE2_OVERLAY_POS_MODE_BOTLEFT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_CENTER%>">
FE2_OVERLAY_POS_MODE_CENTER</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_TOPRIGHT%>">
FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_BOTRIGHT%>">
FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
        <option value="<%=FE2_OVERLAY_POS_MODE_XY%>">FE2_OVERLAY_POS_MODE_XY</optio
n>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_Y'></td>
</tr>

</table>

</td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
<td>
<label>
    <input type='checkbox' class='filter'
        name='FE2_FILTER_PNGEX' value='vfilter:'
        onchange="toggle_ftable('filter_pngex',this.checked)">
FE2_FILTER_PNGEX
</label>

<table id='filter_pngex' class='disabled'>
<tr>
<th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</
abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'
></td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_PREFIX'>
</td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'>
</td>
</tr>

<tr>
<th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_WIDTH'></td>
</tr>

<tr>

```

```

    <th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_PNGEX_HEIGHT'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>

<td>
    <select name="filter:setParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">
    <option value=""></option>
    <option value="<%=on2false%>">FALSE</option>
    <option value="<%=on2true%>">TRUE</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
<td>
    <select name="filter:setParam:FE2_PNGEX_ENABLE_ALPHA">
    <option value=""></option>
    <option value="<%=on2false%>">FALSE</option>
    <option value="<%=on2true%>">TRUE</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="interval in ms; Default: disabled">
    FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_EXPORT_INTERVAL'></td>
</tr>

<tr>
<th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">
    FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
<td><input type='text' name='filter:setParamAsStr:
    FE2_PNGEX_EXPORT_TIME_STRING'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
<td>
    <select name="filter:setParam:FE2_PNGEX_EXPORT_CUE_POINTS">
    <option value=""></option>
    <option value="<%=FE2_PNGEX_CP_ALL%>">All cue points (FE2_PNGEX_CP_ALL)</opt
    ion>
    <option value="<%=FE2_PNGEX_CP_NAV%>">Only navigation cue points (
    FE2_PNGEX_CP_NAV)</option>
    <option value="<%=FE2_PNGEX_CP_EVENT%>">Only event cue points (
    FE2_PNGEX_CP_EVENT)</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">
    FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_COMPRESSION_LEVEL'></t
    d>
</tr>

<tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>

<tr>
<th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>

```

```

        <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></td>
    </tr>

    <tr>
        <th><abbr title="start time in ms; Default: 0">
            FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
        <td><input type='text' name='filter:setParam:
            FE2_PNGEX_AUTO_EXPORT_START_TIME'></td>
    </tr>

    <tr>
        <th><abbr title="stop time in ms; Default: <clip length">
            FE2_PNGEX_AUTO_EXPORT_END_TIME</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_END_TIME'>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: 0">FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD</abbr></th>
        <td><input type='text' name='filter:setParam:
            FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - PNGEX filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_ROTATE' value='vfilter:'
            onchange="toggle_ftable('filter_rotate',this.checked)">
        FE2_FILTER_ROTATE
    </label>

    <table id='filter_rotate' class='disabled'>
        <tr>
            <th><abbr title="Default: 0. valid: {0,90,180,270}">FE2_ROTATE_ANGLE</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_ROTATE_ANGLE'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - ROTATE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_SCALE' value='vfilter:'
            onchange="toggle_ftable('filter_scale',this.checked)">
        FE2_FILTER_SCALE
    </label>

    <table id='filter_scale' class='disabled'>
        <tr>
            <th><abbr title="Default: input image width">FE2_SCALE_WIDTH</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_SCALE_WIDTH'></td>
        </tr>

        <tr>

```

```

        <th><abbr title="Default: input image height">FE2_SCALE_HEIGHT</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_SCALE_HEIGHT'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - SCALE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_SHARPEN' value='vfilter:'
            onchange="toggle_ftable('filter_sharpen',this.checked)">
        FE2_FILTER_SHARPEN
    </label>

    <table id='filter_sharpen' class='disabled'>
        <tr>
            <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
        </tr>
    </table>

</td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_HIGHPASS' value='afilter:'
            onchange="toggle_ftable('filter_highpass',this.checked)">
        FE2_FILTER_HIGHPASS
    </label>

    <table id='filter_highpass' class='disabled'>
        <tr>
            <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_HIGHPASS_Q'></td>
        </tr>

        <tr>
            <th>FE2_HIGHPASS_CUTOFF</th>
            <td><input type='text' name='filter:setParam:FE2_HIGHPASS_CUTOFF'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_LOWPASS' value='afilter:'
            onchange="toggle_ftable('filter_lowpass',this.checked)">
        FE2_FILTER_LOWPASS
    </label>

    <table id='filter_lowpass' class='disabled'>
        <tr>
            <th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>

```

```

        <td><input type='text' name='filter:setParam:FE2_LOWPASS_Q'></td>
    </tr>

    <tr>
        <th>FE2_LOWPASS_CUTOFF</th>
        <td><input type='text' name='filter:setParam:FE2_LOWPASS_CUTOFF'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_RESAMPLE' value='a'filter:'
                onchange="toggle_ftable('filter_resample',this.checked)">
            FE2_FILTER_RESAMPLE
        </label>

        <table id='filter_resample' class='disabled'>
            <tr>
                <th><abbr title="Default: 0">FE2_RESAMPLE_RATE</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_RESAMPLE_RATE'></td>
            </tr>

            <tr>
                <th><abbr title="Default: 0">FE2_RESAMPLE_CHANNELS</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_RESAMPLE_CHANNELS'></td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - RESAMPLE filter parameters -->

</table>
</fieldset>

<!-- ##MUXERS#####
    ### -->
<hr>
<fieldset class='collapsed' id="muxers">
<legend onclick='toggle_expand(this)'>Muxers</legend>
<table>

<tr>
    <td>
        <label>
            <input type='radio' name='muxer:' value='FE2_MUXER_3GP'
                onfocus="set_muxer_visible('tgptable') ">
            FE2_MUXER_3GP&nbsp;
        </label>
        <label>
            <input type='radio' name='muxer:' value='FE2_MUXER_3G2'
                onfocus="set_muxer_visible('tg2table') ">
            FE2_MUXER_3G2&nbsp;
        </label>
        <label>
            <input type='radio' name='muxer:' value='FE2_MUXER_MOV'
                onfocus="set_muxer_visible('movtable') ">
            FE2_MUXER_MOV&nbsp;
        </label>
        <label>
            <input type='radio' name='muxer:' value='FE2_MUXER_MP4'

```

```

        onfocus="set_muxer_visible('mp4table') ">
    FE2_MUXER_MP4&nbsp;
</label>
<br>

<label>
    <input type='radio' name='muxer:' value='FE2_MUXER_FLV'
        onfocus="set_muxer_visible('flvtable',true) ">
    FE2_MUXER_FLV&nbsp;
</label>
<label>
    <input type='radio' name='muxer:' value='FE2_MUXER_SWF'
        onfocus="set_muxer_visible('swftable',true) ">
    FE2_MUXER_SWF
</label>
<br>

<label>
    <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
        onfocus="set_muxer_visible('fxmtable',true) ">
    FE2_MUXER_FXM&nbsp;
</label>
<br>

<label>
    <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
        onfocus="set_muxer_visible('webmtable',true) ">
    FE2_MUXER_WEBM&nbsp;
</label>
</td>
</tr>

<tr>
<td>
<!-- 3GP muxer parameters -->
<table id='tgptable' class='hidden'>
    <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_3GP_FASTSTART">
                <option value=""></option>
                <option value="<%=on2false%>">FALSE</option>
                <option value="<%=on2true%>">TRUE</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - 3GP muxer parameters -->

<!-- 3G2 muxer parameters -->
<table id='tg2table' class='hidden'>
    <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_3G2_FASTSTART">
                <option value=""></option>
                <option value="<%=on2false%>">FALSE</option>
                <option value="<%=on2true%>">TRUE</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - 3G2 muxer parameters -->

<!-- FLV muxer parameters -->
<table id='flvtable' class='hidden'>
    <tr>

```



```

<th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0' "
>
    FE2_FLV_CUEPT_EVENT</abbr></th>
<td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
</tr>

<tr>
<th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.
0' "
    FE2_FLV_CUEPT_NAV</abbr></th>
<td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_NAV'></td>
</tr>

<tr>
<th><abbr title="e.g. 'cuept_name& n0=v0& n1=v1...' "
    FE2_FLV_CUEPT_PARAM</abbr></th>
<td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
    FE2_FLV_METADATA_ENABLE</abbr></th>
<td>
<table id='flv_metadata_enable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr
title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
DURATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' value='<%=MD_
DURATION%'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr
title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' value='<%=MD_
DATASIZE%'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><ab
br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' value='<%=
MD_AUDIO_SIZE%'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><ab
br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' value='<%=
MD_VIDEO_SIZE%'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'
><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' v
alue='<%=MD_AUDIO_DATARATE%'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'

```

```

><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' v
alue='<%=MD_VIDEO_DATARATE%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'>
<abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' val
ue='<%=MD_AUDIO_CODECID%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'>
<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' val
ue='<%=MD_VIDEO_CODECID%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr ti
tle="Default: Enabled">MD_WIDTH</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' value='<%=MD_WIDTH%
>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' value='<%=MD_HEIG
HT%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' value='<%=M
D_FRAMERATE%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' value
='<%=MD_CANSEEKTOEND%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'>
<abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' val
ue='<%=MD_LASTTIMESTAMP%>'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIM
ESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></
th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_

```

```

LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRA
METIMESTAMP' value='<%=MD_LASTKEYFRAMETIMESTAMP%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOC
ATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAM
ELOCATION' value='<%=MD_LASTKEYFRAMELOCATION%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abb
r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' value='<%=M
D_KEYFRAMES%>'></td>
</tr>
</table>
</td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
FE2_FLV_METADATA_DISABLE</abbr></th>
<td>
<table id='flv_metadata_disable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abb
r title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_
_DURATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' value='<%=M
D_DURATION%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abb
r title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_
_DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' value='<%=M
D_DATASIZE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><a
bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_
_AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' value='
<%=MD_AUDIO_SIZE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><a
bbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_
_VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' value='
<%=MD_VIDEO_SIZE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE
'><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_
_AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'

```

```

value='<%=MD_AUDIO_DATARATE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE' value='<%=MD_VIDEO_DATARATE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' value='<%=MD_AUDIO_CODECID%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' value='<%=MD_VIDEO_CODECID%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabled">MD_WIDTH</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' value='<%=MD_WIDTH%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enabled">MD_HEIGHT</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' value='<%=MD_HEIGHT%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' value='<%=MD_FRAMERATE%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' value='<%=MD_CANSEEKTOEND%>'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' value='<%=MD_LASTTIMESTAMP%>'></td>
</tr>

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAME_TIMESTAMP'><abbr title="Default: Enabled">MD_LASTKEYFRAME_TIMESTAMP</abbr></label></th>  ☐| <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAME_LOCATION'><abbr title="Default: Enabled">MD_LASTKEYFRAME_LOCATION</abbr></label></th>  ☐| <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: Enabled">MD_KEYFRAMES</abbr></label></th>  ☐  </td> </tr> </table> <!-- END - FLV muxer parameters -->  <!-- FXM muxer parameters -->   | <th> &lt;th&gt;&lt;abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'"&gt; FE2_FXM_CUEPT_NAV&lt;/abbr&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_NAV" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="e.g. 'cuept_name&amp; n0=v0&amp; n1=v1...'"&gt; FE2_FXM_CUEPT_PARAM&lt;/abbr&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> </td></th></td></tr></td></th> | <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'"> FE2_FXM_CUEPT_NAV</abbr></th> <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_NAV" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="e.g. 'cuept_name&amp; n0=v0&amp; n1=v1...'"&gt; FE2_FXM_CUEPT_PARAM&lt;/abbr&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> </td></th></td></tr></td> | <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_NAV" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="e.g. 'cuept_name&amp; n0=v0&amp; n1=v1...'"&gt; FE2_FXM_CUEPT_PARAM&lt;/abbr&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> </td></th></td></tr> | <th> &lt;th&gt;&lt;abbr title="e.g. 'cuept_name&amp; n0=v0&amp; n1=v1...'"&gt; FE2_FXM_CUEPT_PARAM&lt;/abbr&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> </td></th> | <th><abbr title="e.g. 'cuept_name& n0=v0& n1=v1...'"> FE2_FXM_CUEPT_PARAM</abbr></th> <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> </td> | <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> | <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th> | <th><abbr title="Select specific metadata entries to enable. Default for each item is provided."> FE2_FXM_METADATA_ENABLE</abbr></th> <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> | <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> | <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th> | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Enabled">MD_DURATION</abbr></label></th> <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td> | <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> | |----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------| | <th> &lt;th&gt;&lt;abbr title="e.g. 'cuept_name&amp; n0=v0&amp; n1=v1...'"&gt; FE2_FXM_CUEPT_PARAM&lt;/abbr&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> </td></th>                                                                                                                                                                                                                                                                                  | <th><abbr title="e.g. 'cuept_name& n0=v0& n1=v1...'"> FE2_FXM_CUEPT_PARAM</abbr></th> <td> <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr> </td>                                                                                                                                                                                                                                                                                          | <input &gt;&lt;="" <="" name="muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM" td&gt;="" tr="" type="text"/> <tr> <td> <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th></td></tr>                                                                                                                                                                                                                                                                   | <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th>                                                                                                                                                                                                                                                                     | <th><abbr title="Select specific metadata entries to enable. Default for each item is provided."> FE2_FXM_METADATA_ENABLE</abbr></th> <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td>                                                                                                                                                                                                                                                              | <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table>                                                                                                                                                                                                                                                                                                              | <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th>                                                                                                                                                                                                                                                                | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Enabled">MD_DURATION</abbr></label></th> <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td>                                                                                                                                                                                                                                                                           | <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/>                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                            | | <th> &lt;th&gt;&lt;abbr title="Select specific metadata entries to enable. Default for each item is provided."&gt; FE2_FXM_METADATA_ENABLE&lt;/abbr&gt;&lt;/th&gt; <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td> </th>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <th><abbr title="Select specific metadata entries to enable. Default for each item is provided."> FE2_FXM_METADATA_ENABLE</abbr></th> <td> <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table> </td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <table &gt;="" <tr="" class="" id="fxm_metadata_enable"> <td> <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th></td></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <th> &lt;th&gt;&lt;label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'&gt;&lt;abbr title="Default: Enabled"&gt;MD_DURATION&lt;/abbr&gt;&lt;/label&gt;&lt;/th&gt; <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td></th>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Enabled">MD_DURATION</abbr></label></th> <td> <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/> </td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <input &gt;&lt;="" <="" id="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" name="muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION" td&gt;="" tr="" type="checkbox" value="&lt;%=MD_DURATION%&gt;"/>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                            | | | | | | | | | |

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Enabled">MD_DATASIZE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' value='<%=MD_ DATASIZE%'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><ab br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' value='<% =MD_AUDIO_SIZE%'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><ab br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' value='<% =MD_VIDEO_SIZE%'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' ><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' v alue='<%=MD_AUDIO_DATARATE%'></td> </tr>  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' ><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' v alue='<%=MD_VIDEO_DATARATE%'></td> </tr>  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'> <abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' val ue='<%=MD_AUDIO_CODECID%'></td> </tr>  |  |  |  |  |  | | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'> <abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' val ue='<%=MD_VIDEO_CODECID%'></td> </tr>  |  |  |  | | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr ti tle="Default: Enabled">MD_WIDTH</abbr></label></th>  <input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ WIDTH' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' value='<%=MD_WIDTH% '></td> </tr>  | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr t itle="Default: Enabled">MD_HEIGHT</abbr></label></th> | | --- | | | | | | | | | | | | | | | | |

```

```

        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' value='<%=MD_HEIG
HT%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' value='<%=M
D_FRAMERATE%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' value
='<%=MD_CANSEEKTOEND%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'>
<abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' val
ue='<%=MD_LASTTIMESTAMP%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIM
ESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></
th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRA
METIMESTAMP' value='<%=MD_LASTKEYFRAMETIMESTAMP%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOC
ATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAM
ELOCATION' value='<%=MD_LASTKEYFRAMELOCATION%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abb
r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' value='<%=M
D_KEYFRAMES%>'></td>
    </tr>
</table>
</td>
</tr>

<tr>
    <th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
        FE2_FXM_METADATA_DISABLE</abbr></th>
    <td>
        <table id='fxm_metadata_disable' class=''>
            <tr>
                <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abb
r title="Default: Enabled">MD_DURATION</abbr></label></th>

```

```

        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_DURATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' value='<%=M
D_DURATION%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abbr
r title="Default: Enabled">MD_DATASIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' value='<%=M
D_DATASIZE%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><a
bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' value='
<%=MD_AUDIO_SIZE%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><a
bbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' value='
<%=MD_VIDEO_SIZE%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE
'><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'
value='<%=MD_AUDIO_DATARATE%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE
'><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'
value='<%=MD_VIDEO_DATARATE%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'
><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' v
alue='<%=MD_AUDIO_CODECID%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'
><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' v
alue='<%=MD_VIDEO_CODECID%>'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr t
itle="Default: Enabled">MD_WIDTH</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_WIDTH' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' value='<%=MD_WIDT
H%>'></td>

```



```

    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr
title="Default: Enabled">MD_HEIGHT</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' value='<%=MD_HE
IGHT%>'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><ab
br title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' value='<%=
MD_FRAMERATE%>'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'>
<abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' val
ue='<%=MD_CANSEEKTOEND%>'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'
><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' v
alue='<%=MD_LASTTIMESTAMP%>'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETI
MESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label><
/th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYF
RAMETIMESTAMP' value='<%=MD_LASTKEYFRAMETIMESTAMP%>'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></t
h>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='<%=MD_LASTKEYFRAMELOCATION%>'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><ab
br title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' value='<%=
MD_KEYFRAMES%>'></td>
    </tr>
  </table>
</td>
</tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>

```

```

<tr>
  <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
  <td>
    <select name="muxer:setParam:FE2_MOV_FASTSTART">
      <option value=""></option>
      <option value="<%=on2false%>">FALSE</option>
      <option value="<%=on2true%>">TRUE</option>
    </select>
  </td>
</tr>
</table>
<!-- END - MOV muxer parameters -->

<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
  <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
      <select name="muxer:setParam:FE2_MP4_FASTSTART">
        <option value=""></option>
        <option value="<%=on2false%>">FALSE</option>
        <option value="<%=on2true%>">TRUE</option>
      </select>
    </td>
  </tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
  <tr>
    <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_WIDTH'></td>
  </tr>

  <tr>
    <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_HEIGHT'></td>
  </tr>

  <tr>
    <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_FRAMERATE'></td>
  </tr>

  <tr>
    <th>FE2_SWF_LOOP_COUNT</th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_LOOP_COUNT'></td>
  </tr>

  <tr>
    <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
  </tr>

  <tr>
    <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
    <td>
      <select name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>
        <option value=""></option>
        <option value="_self">_self</option>
        <option value="_blank">_blank</option>
        <option value="_parent">_parent</option>
        <option value="_top">_top</option>
      </select>
    </td>
  </tr>
</table>

```

```

<tr>
  <th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</a
    bbr></th>
  <td>
    <select name='muxer:setParam:FE2_SWF_EMBEDDED_URL_TYPE'>
      <option value=""></option>
      <option value="<%=EmbeddedUrlIsGetUrl%>">EmbeddedUrlIsGetUrl</option>
      <option value="<%=EmbeddedUrlIsLoadMovie%>">EmbeddedUrlIsLoadMovie</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="e.g. n0=v0&nl=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
  <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
</tr>

<tr><th><b>Preloader Settings:</b></th><td></td></tr>

<tr>
  <th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
  <td>
    <select name='muxer:setParam:FE2_SWF_PRELOAD_TYPE'>
      <option value=""></option>
      <option value="<%=SwfPreloaderNone%>">SwfPreloaderNone</option>
      <option value="<%=SwfFixedPreloader%>">SwfFixedPreloader</option>
      <option value="<%=SwfAdaptivePreloader%>">SwfAdaptivePreloader</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
  <td><input type='text' name='muxer:setParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
  <td><input type='text' name='muxer:setParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
  <th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTION</abbr></th>
  <td>
    <select name='muxer:setParam:FE2_SWF_ON_START_OPTION'>
      <option value=""></option>
      <option value="<%=SwfOnMovieStartAutomatically%>">SwfOnMovieStartAutomatical
        ly</option>
      <option value="<%=SwfOnMovieStartOnClick%>">SwfOnMovieStartOnClick</option>
      <option value="<%=SwfOnMovieStartWait%>">SwfOnMovieStartWait</option>
      <option value="<%=SwfOnMovieStartEmbedSTOP%>">SwfOnMovieStartEmbedSTOP</opti
        on>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
  <td><input type='text' name='muxer:setParam:FE2_SWF_START_BLANK_FRAME'></td>
</tr>

```

```

<tr>
  <th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
  <td><input type='text' name='muxer:setParam:FE2_SWF_START_WAIT_SEC'></td>
</tr>

<tr><th><b>End Settings:</b></th><td></td></tr>

<tr>
  <th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
  <td>
    <select name='muxer:setParam:FE2_SWF_ON_END_OPTION'>
      <option value=""></option>
      <option value="<%=SwfOnMovieEndNothing%>">SwfOnMovieEndNothing</option>
      <option value="<%=SwfOnMovieEndSTOP%>">SwfOnMovieEndSTOP</option>
      <option value="<%=SwfOnMovieEndLoop%>">SwfOnMovieEndLoop</option>
      <option value="<%=SwfOnMovieEndUnload%>">SwfOnMovieEndUnload</option>
      <option value="<%=SwfOnMovieEndLoadMovie%>">SwfOnMovieEndLoadMovie</option>
    </select>
  </td>
</tr>

<tr>
  <th>FE2_SWF_ON_END_URL</th>
  <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ON_END_URL'></td>
</tr>
</table>
<!-- END - SWF muxer parameters -->

<!-- WEBM muxer parameters -->
<table id='webmtable' class='hidden'>
  <tr>
    <th>(FE2_MUXER_WEBM defines no parameters)</th>
  </tr>
</table>
<!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM##### -->
<hr>
<p>
  <input type="submit" value="Start Encode">
  <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>

```

6.3.2 process_sample.aspx

```

<%@ Page Language="VB" Strict = "true"%>
<%@ Assembly Name = "Interop.flixengine_com"%>
<%@ Import Namespace="flixengine_com" %>
<%@ Import Namespace="System.Reflection" %>
<%@ Import Namespace="System.Runtime.InteropServices" %>
<%@ Import Namespace="System.IO" %>

<script runat="server">
'=====
'
' Copyright (c) On2 Technologies Inc. All Rights Reserved.
'

```

```

'-----
'
'   File:           $Workfile$
'                   $Revision$
'
'   Last Update: $DateUTC$
'
'-----

'process_sample.aspx
' Receive a form via post from flix2_sample.aspx, treating each name=value pair
' as a function/param pair.
' These map to the Flix Engine COM API and each function that has a valid
' param will be called.
' Once setup is complete, calls encode() to produce an output file located in out
  dir

'Global Variables
Dim flix As Flix
Dim codec, filter, muxer As IFlixPlgn
Dim prefix, outdir As String

'Prints the status codes of the Flix Engine object
Sub PrintEncoderStatus()
    Dim flixerr As FE2_errno
    Dim syserr As Integer

    flix.errno_(flixerr, syserr)
    Response.Write("<p>Encoder Status<br>" & vbCrLf)
    Response.Write("&nbsp;&nbsp;&nbsp;flix.getEncoderState: " & flix.getEncoderState() & "<br>" & vbCrLf)
    Response.Write("&nbsp;&nbsp;&nbsp;flix.errno_: flixerrno:" & flixerr _
        & " syserrno:" & syserr & "</p>" & vbCrLf)
End Sub

'Handles a COMException object while processing any of the functions
'which are included in the HTML table when executed.
'errNum is passed in as the error value of the object
'in which the exception originated.
Sub HandleComException(ByVal e As COMException, ByVal errNum As Integer)
    Response.Write("<td align=""center"">" & errNum & "</td>" & vbCrLf)
    Response.Write("</tr></table><br>" & vbCrLf)
    PrintStackTrace(e)
End Sub

'Handles an Exception passed to it and calls Finish() to exit.
Sub PrintStackTrace(ByVal e As Exception)
    Response.Write("<br>" & e.ToString() & "<br>" & vbCrLf)
    Finish()
End Sub

'Parses parameters which have been passed as form items from flix2_sample.asp
Sub ParseParams()
    Response.Write("<table border='1' cellpadding='5'>" _
        & "<caption>Flix Function Calls</caption>" & vbCrLf _
        & "<tr><th>Function Name</th><th>Return Value</th></tr>" _
        & vbCrLf)

    Dim x As Integer
    Dim name, value As String
    For x = 0 to (Request.Form.Count - 1)
        name = Request.Form.GetKey(x)
        If (Len( CStr( Request.Form.Item(name))) > 0) Then
            value = Request.Form.Item(name)
            Response.Write(name & " = " & value & "<br>" & vbCrLf)
            If (name = "setOutputFile") Then
                SimpleSet(name, outdir & value)
            End If
        End If
    Next x
End Sub

```

```

        ElseIf ((name = "vcodec:") Or (name = "acodec:")) Then
            InitCodec(value)
        ElseIf ((InStr(1, value, "afilter:") > 0) _
            Or (InStr(1, value, "vfilter:") > 0)) Then
            InitFilter(name)
        ElseIf (name = "muxer:") Then
            InitMuxer(value)
        Else
            Dim temp() As String = Split(name, ":", -1)
            If (temp(0) = "codec") Then
                CodecInterface(temp(1), temp(2), value)
            ElseIf (temp(0) = "filter") Then
                FilterInterface(temp(1), temp(2), value)
            ElseIf (temp(0) = "muxer") Then
                MuxerInterface(temp(1), temp(2), value)
            Else
                SimpleSet(name, value)
            End If
        End If
    End If
Next
End Sub

'Creates the Flix Engine Object and displays information about the object
Sub LoadEngine()
    '
    ' retrieve the main engine interface, IFlix
    '
    Try
        flix = New Flix()
    Catch e as COMException
        PrintStackTrace(e)
    End Try

    'Print Flix Version information
    Try
        Response.Write("Flix Engine COM library. Flix Engine v" _
            & flix.version() & vbCrLf)
        Response.Write(" COM v" & flix.com_version() & "<br>" & vbCrLf & vbCrLf)
        Response.Write(Replace(flix.copyright(), vbCrLf, "<br>" & vbCrLf) _
            & "<br>" & vbCrLf & vbCrLf)
    Catch e as COMException
        PrintStackTrace(e)
    End Try
End Sub

'Finishes the HTML page and ends execution
Sub Finish()
    'Release all global references
    codec = Nothing
    filter = Nothing
    muxer = Nothing

    If Not IsNothing(flix) Then
        PrintEncoderStatus()
        If (flix.isEncoderRunning = 1) Then
            Try
                flix.stopEncoding()
            Catch e as COMException
                Response.Write("flix.stopEncoding Error: " & vbCrLf _
                    & flix.flixerrno() & "<br>" & vbCrLf)
                Response.Write(e.ToString() & "<br>" & vbCrLf)
            End Try
            If (flix.isEncoderRunning = 1) Then
                Response.Write("Flix failed to stop encoding, but did not report
an error!<br>" & vbCrLf)
            End if
        End if
    End If
End Sub

```

```

    End If
    'Force the cleanup of IFlix.
    'Though this is not strictly necessary in this sample, as
    'it is about to exit, if the script is more involved it may be
    'necessary so the input file can be moved as destruction of
    'the underlying FLIX2HANDLE occurs within IFlix's destructor.
    'By explicitly defining a WeakReference and removing the strong
    'reference to the flix object we are guaranteeing that it will
    'not survive the garbage collection.
    Dim wkref As WeakReference = New WeakReference(flix)
    flix = Nothing
    System.GC.Collect(System.GC.GetGeneration(wkref))
    System.GC.WaitForPendingFinalizers()
End If

'Close the html page
Response.Write("</body>" & vbCrLf & "</html>" & vbCrLf)
Response.End
End Sub

'Executes the method funcName with arguments args on object targetObject
'Requires System.Reflection
Function ExecuteMethod(ByRef targetObject As Object, ByVal funcName As String, _
    ByVal args() As Object) As Object
    Dim objectType As Type = targetObject.GetType()
    Try
        ExecuteMethod = objectType.InvokeMember(funcName, BindingFlags.InvokeMethod, _
            Nothing, targetObject, args)
    Catch e As TargetInvocationException
        Throw e.InnerException
    End Try
End Function

'Reads property propName from object targetObject
'Requires System.Reflection
Function GetProp(ByRef targetObject As Object, ByVal propName As String) As Object
    Dim objectType As Type = targetObject.GetType()
    Try
        GetProp = objectType.InvokeMember(propName, BindingFlags.GetProperty, _
            Nothing, targetObject, Nothing)
    Catch e As TargetInvocationException
        Throw e.InnerException
    End Try
End Function

'Calls an IFlix function with a single parameter
Sub SimpleSet(ByVal func As String, ByVal arg As String)
    Response.Write("<tr><td>flix." & func & "( " & arg & " )</td>" & vbCrLf)
    Try
        ExecuteMethod(CType(flix, Object), func, New Object() {arg})
    Catch e As COMException
        HandleComException(e, flix.sc)
    End Try
    Response.Write("<td align=""center"">" & flix.sc & "</td></tr>" & vbCrLf)
End Sub

'Adds a codec, creating an IFlixPlgn object.
Sub InitCodec(ByVal name As String)
    'if name is a codec name, e.g. FE2_CODEEC_VP6, add an instance
    'we'll assume all IFlixPlgn::setParam's relate to this codec until we
    'hit the next codec name
    Response.Write("<tr><td>flix.addCodec( " & name & " )</td>" & vbCrLf)
    Try
        codec = flix.addCodec(Cstr(GetProp(CType(flix, Object), name)))
    Catch e As COMException

```

```

        HandleCOMException(e, flix.sc)
    End Try
    Response.Write("<td align=""center"">" & flix.sc & "</td></tr>" & vbCrLf)
End Sub

'Calls a codec-related IFlixPlgn function.
'Assumes that a codec has been added in InitCodec.
Sub CodecInterface(ByVal funcname As String, ByVal name As String, _
    ByVal value As String)
    Response.Write("<tr><td>codec." & funcname & "(" & name & ", " & _
        & value & ")"</td>" & vbCrLf)
    Try
        ExecuteMethod(CType(codec, Object), funcname, _
            New String() {CStr(GetProp(CType(flix, Object), name)), _
                value})
    Catch e as COMException
        HandleCOMException(e, codec.sc)
    End Try
    Response.Write("<td align=""center"">" & codec.sc & "</td></tr>" & vbCrLf)
End Sub

'Adds a filter, creating an IFlixPlgn object.
Sub InitFilter(ByVal name As String)
    'if name is a filter name, e.g. FE2_FILTER_CUT, add an instance
    'we'll assume all IFlixPlgn::setParam's relate to this filter until we
    'hit the next filter name
    Response.Write("<tr><td>flix.addFilter( " & name & " )</td>" & vbCrLf)
    Try
        filter = flix.addFilter(Cstr(GetProp(CType(flix, Object), name)))
    Catch e as COMException
        HandleCOMException(e, flix.sc)
    End Try
    Response.Write("<td align=""center"">" & flix.sc & "</td></tr>" & vbCrLf)
End Sub

'Calls a filter-related IFlixPlgn function.
'Assumes that a filter has been added in InitFilter.
Sub FilterInterface(ByVal funcname As String, ByVal name As String, _
    ByVal value As String)
    Response.Write("<tr><td>filter." & funcname & "(" & name & ", " & _
        & value & ")"</td>" & vbCrLf)
    Try
        ExecuteMethod(CType(filter, Object), funcname, _
            New String() {CStr(GetProp(CType(flix, Object), name)), _
                value})
    Catch e as COMException
        HandleCOMException(e, filter.sc)
    End Try
    Response.Write("<td align=""center"">" & filter.sc & "</td></tr>" & vbCrLf)
End Sub

'Adds a muxer, creating an IFlixPlgn object.
Sub InitMuxer(ByVal name As String)
    'if name is a muxer name, e.g. FE2_MUXER_FLV, add an instance
    'we'll assume all IFlixPlgn::setParam's relate to this muxer until we
    'hit the next muxer name
    Response.Write("<tr><td>flix.addMuxer( " & name & " )</td>" & vbCrLf)
    Try
        muxer = flix.addMuxer(Cstr(GetProp(CType(flix, Object), name)))
    Catch e as COMException
        HandleCOMException(e, flix.sc)
    End Try
    Response.Write("<td align=""center"">" & flix.sc & "</td></tr>" & vbCrLf)
End Sub

'Calls a muxer-related IFlixPlgn function.

```



```

'Assumes that a muxer has been added in InitMuxer.
Sub MuxerInterface(ByVal funcname As String, ByVal name As String, _
    ByVal value As String)
    Response.Write("<tr><td>muxer." & funcname & "(" & name & ", " & _
        & value & " )</td>" & vbCrLf)
    Try
        ExecuteMethod(CType(muxer, Object), funcname, _
            New String() {CStr(GetProp(CType(flix, Object), name)), _
                value})
    Catch e as COMException
        HandleCOMException(e, muxer.sc)
    End Try
    Response.Write("<td align=""center"">" & muxer.sc & "</td></tr>" & vbCrLf)
End Sub

'Starts the encode and prints encoding updates.
Sub Encode()
    '
    ' start the encode
    '
    Response.Write("<tr><td>flix.encode( )</td>" & vbCrLf)
    Try
        flix.encode()
    Catch e as COMException
        HandleCOMException(e, flix.sc)
    End Try
    Response.Write("<td align=""center"">" & flix.sc & "</td></tr>" & vbCrLf)

    '
    ' retrieve the encoding status interface, IEncodingStatus
    '
    Response.Write("<tr><td>flix.encodingStatus( )</td>" & vbCrLf)
    Dim encStatus As IEncodingStatus
    Try
        encStatus = flix.encodingStatus()
    Catch e as COMException
        HandleCOMException(e, flix.sc)
    End Try
    Response.Write("<td align=""center"">" & flix.sc & "</td></tr>" & vbCrLf)

    Response.Write("</table>" & vbCrLf)

    Response.Write("<p>Encoding...(video frames encoded, percent complete). " & _
        & "Total frames will reset when doing 2pass.<br>" & vbCrLf)

    'Start the progress updates
    Dim isRunning As Integer
    Try
        Do
            isRunning = flix.isEncoderRunning()
            Response.Write("(" & encStatus.totalFrames())
            Response.Write(", " & encStatus.percentComplete() & "%)<br>" & vbCrLf)
            Threading.Thread.Sleep(1000) '1 second intervals
        Loop Until ((isRunning = 0) or (Response.IsClientConnected() = False))
    Catch e as COMException
        PrintStackTrace(e)
    End Try
    Response.Write("<br>Done!</p>" & vbCrLf)
End Sub

</script>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/str
ict.dtd">
<html>
    <head>
        <title>Flix CGI Process Sample - ASP.NET</title>

```

```

        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    </head>
<body>

<%
' Increase script timeout to allow encodes of long files to complete,
' barring any further server side timeout settings
Server.ScriptTimeout = 43200 '= 12 hour timeout
Response.BufferOutput = False

Response.Write("<hr><p>process_sample.aspx version 1.1<br>" & vbCrLf)

prefix = "C:\Inetpub\"
outdir = prefix & "flixmedia\out\"

LoadEngine()

Try
    If Directory.Exists(outdir) Then
        Dim temp as StreamWriter = File.CreateText(outdir & "temp.dat")
        temp.Close
    End If
Catch e as Exception
    Response.Write("<p>*****<br>" & vbCrLf)
    Response.Write("<b>WARNING</b>: <i>' " & outdir & "'</i>" _
        & " MUST exist and be writeable by" _
        & " <i>flixengine_com.dll</i> as user " _
        & System.Security.Principal.WindowsIdentity.GetCurrent().Name
    _
        & ".<br>" & vbCrLf)

    Response.Write("Please make <i>' " & outdir & "'</i> accessible or modify" _
        & " the <tt>outdir</tt> value defined in '" _
        & Server.MapPath(Request.ServerVariables("SCRIPT_NAME")) _
        & "'.<br>" & vbCrLf)

    Response.Write("*****</p>"&vbLf)
End Try
'If possible, delete the temp.dat file we created to check writability
Try
    File.Delete(outdir & "temp.dat")
Catch e as Exception
    'Ignore failure - Full Control access is not required for script operation
End Try

ParseParams()
Encode()
Finish()
%>

</body>
</html>

```

6.4 Command Line

```

/*
//=====
//
// Copyright (c) On2 Technologies Inc. All Rights Reserved.
//
//-----
//
// File:          $Workfile$
//               $Revision$
//
// Last Update:  $DateUTC$

```

```
//
//-----
*/
#define _WIN32_WINNT 0x0400 //for COINIT_MULTITHREADED
#include <atlbase.h>
#include <atlcom.h>
#include <tchar.h>
#include <iostream>
#include <iomanip>
#include "flixengine_com.h"

using std::cout;
using std::cerr;
using std::endl;
using std::ios_base;

/*
    Retrieve description for HRESULT via FormatMessage
*/
LPTSTR hresult_desc(HRESULT hr)
{
    LPTSTR str;
    if(HRESULT_FACILITY(hr) == FACILITY_WINDOWS)
        hr = HRESULT_CODE(hr);

    if(!FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL, hr, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&str, 0, NULL)) {
        _stprintf(str= (LPTSTR)LocalAlloc(LPTR,512),
            _T("[no description available, HRESULT_CODE(hr):%d]\n"),hr);
    }

    return str;
}

/*
    Call 'func', outputting any available information on failure before exiting
*/
#define CHECK_HR(func) {\
    HRESULT hr = (func);\
    if(FAILED(hr)) {\
        LPTSTR desc = hresult_desc(hr);\
        /*print the failing function*/\
        cerr << endl << #func << " failed!" << endl\
            << "HRESULT(" << endl << hr << "," << endl << desc << ")" << endl;\
        LocalFree(desc);\
        \
        /*if we've instantiated the IFlix interface..*/\
        __if_exists(flix) {\
            FE2_errno flixerr = ErrNone;\
            on2s32 syserr = 0;\
            /*retrieve further error information if possible*/\
            hr= flix->errno_(&flixerr,&syserr);\
            cerr << "flix->errno_: hr:"\
                << std::setiosflags(ios_base::hex|ios_base::showbase)\
                << std::setw(8) << hr\
                << std::dec << " flixerrno:" << flixerr\
                << " syserrno:" << syserr << endl;\
        }\
        exit(EXIT_FAILURE);\
    }\
}

void print_encoder_status(IFlix* flix)
{
    cout << "\nEncoder Status\n";
}
```

```

FE2_EncState state;
CHECK_HR( flix->getEncoderState(&state) )
cout << " flix->getEncoderState: " << state << endl;

FE2_errno flixerr = ErrNone;
on2s32 syserr;
CHECK_HR( flix->errno_(&flixerr,&syserr) )
cout << " flix->errno_: flixerrno:" << flixerr
    << " syserrno:" << syserr << endl;
}

int _tmain(int argc, _TCHAR* argv[])
{
    CHECK_HR( CoInitializeEx(NULL,COINIT_MULTITHREADED) );

    /*
        create an instance of the Flix class (Flix Engine COM)
    */
    IUnknown* unk;
    CHECK_HR( CoCreateInstance(CLSID_Flix, 0, CLSCTX_INPROC_SERVER,
        IID_IUnknown, reinterpret_cast<void*>(&unk)) );

    /*
        retrieve the main engine interface, IFlix
    */
    IFlix* flix = NULL;
    CHECK_HR( unk->QueryInterface(IID_IFlix, reinterpret_cast<void*>(&flix)) );
    unk->Release();

    /*enable logging, 0=none(disable) 1=info 2=error(asserts) 3=debug 4=heavy
        CONOUT$ can be used as the log file name to send output to the console
    CHECK_HR( flix->setLogLevel(3) );
    CHECK_HR( flix->setLogPath(L"\\cli_encode_com.log") );*/

    /*
        print some library information
    */
    BSTR bstr;
    CHECK_HR( flix->version(&bstr) );
    cout << "Flix Engine COM library. Flix Engine v" << COLE2CT(bstr);
    ::SysFreeString(bstr);

    CHECK_HR( flix->com_version(&bstr) );
    cout << " COM v" << COLE2CT(bstr) << endl;
    ::SysFreeString(bstr);

    CHECK_HR( flix->copyright(&bstr) );
    cout << COLE2CT(bstr) << endl << endl;
    ::SysFreeString(bstr);

    if(argc<3) {
        cerr << "usage: cli_encode_com.exe <infile> <outfile>" << endl << endl;
        return EXIT_FAILURE;
    }

    /*
        set the source file
    */
    cout << "Input file : " << argv[1] << endl;
    CHECK_HR( flix->setInputFile(CComBSTR(argv[1])) );

    /*
        retrieve the video options interface, IVideoOptions
    */
    IVideoOptions* vidopts;
    CHECK_HR( flix->videoOptions(&vidopts) );

```

```

/*
    print input file information
*/
on2s32 duration; CHECK_HR( flix->getSourceDuration(&duration) );
on2s32 srcw;      CHECK_HR( vidopts->getSourceWidth(&srcw) );
on2s32 srch;      CHECK_HR( vidopts->getSourceHeight(&srch) );
cout << std::setw(25) << "Width:      " << srcw << endl
     << std::setw(25) << "Height:     " << srch << endl
     << std::setw(25) << "Duration:  " << duration << "ms" << endl;
/*release vidopts as we're through with it*/
vidopts->Release();

/*
    set the destination file
*/
cout << "Output File : " << argv[2] << endl;
CHECK_HR( flix->setOutputFile(CComBSTR(argv[2])) );

/*
    Options may be set and codecs/filters/muxers may be added prior to encode(
    )
*/

/*Add the scale filter
IFlixPlgn* filter;
CHECK_HR( flix->addFilter(CComBSTR(FE2_FILTER_SCALE), &filter) );

CHECK_HR( filter->setParam(CComBSTR(FE2_SCALE_WIDTH),240) );
CHECK_HR( filter->setParam(CComBSTR(FE2_SCALE_HEIGHT),160) );
filter->Release();*/

/*Add the vp6 codec. Though it is the default, you must add it in order
to modify its settings
IFlixPlgn* codec;
CHECK_HR( flix->addCodec(CComBSTR(FE2_CODEC_VP6), &codec) );

CHECK_HR( codec->setParam(CComBSTR(FE2_VP6_RC_MODE), VBR_1PASSControl) );
codec->Release();*/

/*Use the FLV muxer (default)
IFlixPlgn* muxer;
CHECK_HR( flix->addMuxer(CComBSTR(FE2_MUXER_FLV), &muxer) );

muxer->Release();*/

/*
    start the encode
*/
CHECK_HR( flix->encode() );

/*
    retrieve the encoding status interface, IEncodingStatus
*/
IEncodingStatus* encstatus;
CHECK_HR( flix->encodingStatus(&encstatus) );

on2bool b;
on2s32 pcnt;
cout << endl;
do {
    Sleep(500);
    CHECK_HR( flix->isEncoderRunning(&b) );
    CHECK_HR( encstatus->percentComplete(&pcnt) );
    cout << "\rEncoding..." << std::setw(3) << pcnt << "% ";
} while(b);

```

```

    cout << "Done!" << endl;
    print_encoder_status(flix);
    encstatus->Release();

    /*
    cleanup
    */
    flix->Release();
    return EXIT_SUCCESS;
}

```

6.5 C#

The C# example requires that the Microsoft .NET Framework (1.1|2.0) be installed. A pre-built executable is included with the install and can be found under *<InstallDir>\sample\c#\bin\Release*. Should you receive an error dialog containing the text "The application failed to initialize ...", it is likely that you do not have the Framework installed. This component is located on Windows Update under "Software, Optional".

6.6 Command Line

```

/*
//=====
//
// Copyright (c) On2 Technologies Inc. All Rights Reserved.
//
//-----
//
// File:          $Workfile$
//               $Revision$
//
// Last Update:  $DateUTC$
//
//-----
*/
using System;
using System.Runtime.InteropServices; //for COMException
using flixengine_com; //Flix Engine Interfaces

namespace cli_encode
{
    public class cli_encode
    {
        static object createInstance(string progid)
        {
            Type t = Type.GetTypeFromProgID(progid);
            object o = null;
            try {
                o = Activator.CreateInstance(t);
            } catch (Exception e) {
                Console.Error.WriteLine("Error creating object instance: "+e);
                Environment.Exit(1);
            }
            return o;
        }

        static void printStackTrace(IFlix flix, COMException e)
        {
            Console.WriteLine(e);
            if(flix!=null) {

```

```

        printEncoderStatus(flix);
    }
    Environment.Exit(e.ErrorCode);
}

[MTAThread]
static void Main(string[] args)
{
    /*
     * create an instance of the main engine interface, IFlix
     */
    IFlix flix = (IFlix)createInstance("On2.FlixEngine");

    /*enable logging, 0=none(disable) 1=info 2=error(asserts) 3=debug 4=h
eavy
ole
    CONOUT$ can be used as the log file name to send output to the cons
ole
    try {
        flix.setLogLevel(3);
        flix.setLogPath("\\cli_encode_com.log");
    } catch(COMException e) {
        printStackTrace(flix,e);
    }*/

    /*
     * print some library information
     */
    try {
        Console.WriteLine("Flix Engine COM library. Flix Engine v"+
            flix.version()+" COM v"+flix.com_version());
        Console.WriteLine(flix.copyright()+"\n");
    } catch(COMException e) {
        printStackTrace(flix,e);
    }

    if(args.Length<2) {
        Console.Error.WriteLine(
            "usage: cli_encode.exe <infile> <outfile>\n");
        Environment.Exit(1);
    }

    /*
     * set the source file
     */
    try {
        Console.WriteLine("Input file : "+args[0]);
        flix.setInputFile(args[0]);
    } catch(COMException e) {
        printStackTrace(flix,e);
    }

    /*
     * retrieve the video options interface, IVideoOptions
     */
    try {
        IVideoOptions vidopts = flix.videoOptions();

        /*
         * print input file information
         */
        Console.WriteLine("Width:      ".PadLeft(24)+
            vidopts.getSourceWidth());
        Console.WriteLine("Height:    ".PadLeft(24)+
            vidopts.getSourceHeight());
        Console.WriteLine("Duration: ".PadLeft(24)+
            flix.getSourceDuration()+"ms");
    } catch(COMException e) {

```

```

        printStackTrace(flix,e);
    }

    /*
    set the destination file
    */
    try {
        Console.WriteLine("Output File : "+args[1]);
        flix.setOutputFile(args[1]);
    } catch(COMException e) {
        printStackTrace(flix,e);
    }

    /*
    Options may be set and codecs/filters/muxers may be added prior to
encode()
    */

    /*Add the scale filter
    try {
        IFlixPlgn filter = flix.addFilter(flix.FE2_FILTER_SCALE);

        filter.setParam(flix.FE2_SCALE_WIDTH,240);
        filter.setParam(flix.FE2_SCALE_HEIGHT,160);
    } catch(COMException e) {
        printStackTrace(flix,e);
    }*/

    /*Add the vp6 codec. Though it is the default, you must add it in ord
er
    to modify its settings
    try {
        IFlixPlgn codec = flix.addCodec(flix.FE2_CODEC_VP6);

        codec.setParam(flix.FE2_VP6_RC_MODE,
            Convert.ToDouble(FE2_VideoBitrateControls.VBR_1PASSControl));

    } catch(COMException e) {
        printStackTrace(flix,e);
    }*/

    /*Use the FLV muxer (default)
    try {
        IFlixPlgn muxer = flix.addMuxer(flix.FE2_MUXER_FLV);
    } catch(COMException e) {
        printStackTrace(flix,e);
    }*/

    /*
    start the encode
    */
    try {
        flix.encode();

        /*
        retrieve the encoding status interface, IEncodingStatus
        */
        IEncodingStatus encstatus = flix.encodingStatus();

        Console.WriteLine();
        bool ier;
        do {
            System.Threading.Thread.Sleep(500);
            ier = flix.isEncoderRunning()==1;
            Console.Write("\rEncoding..." +
                encstatus.percentComplete().ToString().PadLeft(3)+"% ");
        } while(ier);
    }

```



```

    } catch(COMException e) {
        printStackTrace(flix,e);
    }

    Console.WriteLine("Done!");
    printEncoderStatus(flix);

    /*
    Force the cleanup of IFlix.
    Though this is not strictly necessary in this sample, as
    it is about to exit, if the script is more involved it may be
    necessary so the input file can be moved as destruction of
    the underlying FLIX2HANDLE occurs within IFlix's destructor.
    By explicitly defining a WeakReference and removing the strong
    reference to the flix object we are guaranteeing that it will
    not survive the garbage collection.
    */
    WeakReference wkref = new WeakReference(flix); flix = null;
    System.GC.Collect(System.GC.GetGeneration(wkref));
    System.GC.WaitForPendingFinalizers();
}

static void printEncoderStatus(IFlix flix)
{
    Console.WriteLine("\nEncoder Status");
    try {
        Console.WriteLine(" flix.getEncoderState: "+flix.getEncoderState(
));

        FE2_errno flixerr = FE2_errno.ErrNone;
        int syserr = 0;
        flix.errno_(out flixerr,out syserr);

        Console.WriteLine(" flix.errno_: flixerrno:"+flixerr+
                           " syserrno:" + syserr);
    } catch(COMException e) {
        printStackTrace(null,e);
    }
}
}
}

```

6.7 ColdFusion

The ColdFusion samples have been tested and verified to work with versions 7 and 8.

Additional References:

- [Adobe ColdFusion Product Page](#)

6.8 CGI

This example consists of 2 parts: [flix2_sample.cfm](#) and [process_sample.cfm](#). [flix2_sample.cfm](#) searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's COM interfaces.

This example requires that the account the ColdFusion service runs under have access to the input and output directories. By default this will be the SYSTEM user account.

To use this example navigate to [flix2_sample.cfm](#) in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to [process_sample.cfm](#) which runs the encode loop.

Default file locations used by the scripts:

Input : *C:/Inetpub/flixmedia/in* (indir in [flix2_sample.cfm](#))
Overlay : *C:/Inetpub/flixmedia/overlay* (overlaydir in [flix2_sample.cfm](#))
Output : *C:/Inetpub/flixmedia/out* (outdir in [process_sample.cfm](#))

6.8.1 flix2_sample.cfm

```
<cfscript>
//=====
//
// Copyright (c) On2 Technologies Inc. All Rights Reserved.
//
//-----
//
// File:          $Workfile$
//               $Revision$
//
// Last Update:  $DateUTC$
//
//-----

//NB: For the sample to function correctly these directories must exist
prefix      = "C:\Inetpub\";
indir       = prefix&"flixmedia\in\";           //src file directory
overlaydir  = prefix&"flixmedia\overlay\";      //overlay image file directory
</cfscript>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head>
<title>Flix CGI Sample - Cold Fusion</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<script type="text/javascript">
function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
        i++;
    }
}

function toggle_expand(_this)
{
    _this.parentNode.className= (_this.parentNode.className=='expanded') ?
        'collapsed' : 'expanded';
    document.getElementById('showall').checked=false;
}

function set_table_visible(ctable,visible)
{
    var table= document.getElementById(ctable);

    if(table) {
        if(visible && table.className == 'hidden') {
            table.className= '';
        }
    }
}
```

```

        } else if(!visible && table.className == '') {
            table.className= 'hidden';
        }
    }
}

function hide_tablelist(vistablename,tablelist)
{
    var table= document.getElementById(vistablename);

    if(table&&table.className=='hidden') {
        table.className= '';

        var i=0;
        while(tablelist[i]) {
            if(tablelist[i]!=vistablename) {
                var hiddentable= document.getElementById(tablelist[i]);

                set_table_visible(tablelist[i],false);
                /*clear down any values set in the hidden table to avoid posting
                unnecessary values*/
                clear_value(hiddentable.getElementsByTagName('input'));
                clear_value(hiddentable.getElementsByTagName('select'));
            }
            i++;
        }
    }
}

function set_acodec_visible(ctable)
{
    var acodecs= new Array('aactable','aacplustable','amrnhtable','lametable','pcm
        table','vorbistable');
    hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
    var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8tabl
        e');
    hide_tablelist(ctable,vcodecs);
}

function set_muxer_visible(mtable)
{
    var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','
        tg2table','tgptable','webmtable');
    hide_tablelist(mtable,muxers);
}

function clear_value(list)
{
    var i=0;
    while(list[i]) {
        if(list[i].type=='checkbox') { list[i++].checked=false; }
        else { list[i++].value=''; }
    }
}

function toggle_ftable(ftable,enabled)
{
    var table= document.getElementById(ftable);

    if(table) {
        table.className = enabled ? '' : 'disabled';
        if(!enabled) {
            clear_value(table.getElementsByTagName('input'));
        }
    }
}

```

```

        clear_value(table.getElementsByTagName('select'));
    }
}

function reset_tables()
{
    var table_list= document.getElementsByTagName('table');
    var i=0;
    while(table_list[i]) {
        if (table_list[i].id.length > 7 &&
            table_list[i].id.substring(0,7) == 'filter_') {
            table_list[i].className= 'disabled';
        } else if (table_list[i].id.indexOf('table') != -1) {
            table_list[i].className= 'hidden';
        }
        i++;
    }
}

</script>

<style type="text/css">
<!--
html {
    font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
    font-size: 100%;
    color: rgb(56,56,56);
    background-color: rgb(236,236,236);
    border-style: solid;
    border-color: rgb(236,236,236);
}

body {
    text-align: center;
    margin: 0 auto;
}

div.content {
    color: rgb(56,56,56);
    background-color: rgb(246,246,246);
    text-align: left;
    margin: 0 auto;
    width: 80%;
    min-width: 768px;
    max-width: 932px;
    border-width: 0 1px;
    border-color: rgb(144,144,144);
    border-style: solid;
}

div.content:after {
    content: "";
    color: inherit;
    background-color: rgb(250,250,250);
    border-top: 1px solid rgb(144,144,144);
    height: 20px;
    width: 100%;
    display: block;
}

fieldset table, fieldset {display: none;}
fieldset.expanded, fieldset.collapsed {display: block;}

/*first is fallback for IE*/
fieldset.expanded table {display: block;}
fieldset.expanded table {display: table;}

```

```

h1 {
    font-family: sans-serif;
    font-size: 150%;
    font-weight: normal;
    text-align: left;
    letter-spacing: -1px;
    color: rgb(74,74,74);
    background-color: inherit;
    margin: 0;
}

a {
    color: rgb(74,74,74);
    background-color: transparent;
}

label {
    font-size: 75%;
}

fieldset {
    font-size: 75%;
    line-height: 130%;
    padding: 0;
    margin: 20px;
    border: none;
}

fieldset.expanded {
    color: inherit;
    background-color: rgb(252,252,252);
    border-style: solid;
    border-width: 1px;
    border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
}

legend {
    padding: 0 5px;
    border-left: 10px solid rgb(217,217,217);
    cursor: pointer;
}

legend:hover {text-decoration: underline;}
fieldset.expanded legend {
    font-size: 150%;
    font-weight: bold;
    letter-spacing: -1px;
    background: transparent;
    margin-left: 12px;
    border-right: 10px solid rgb(217,217,217);
    display: block;
}

table {
    font-size: 100%;
    border-spacing: 0;
    /*border-collapse: collapse;*/
    width: 100%;
}

th:before { display: none; }
th, td {
    width: 50%;
    vertical-align: top;
    padding: 2px 3px;
    border-width: 1px 0;
    border-style: solid;
    border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
}

```

```

    }
    tr:first-child>* {border-top-color: rgb(252,252,252);}
    th {
        font-weight: normal;
        text-align: left;
        padding: 2px 2px 2px 5px;
    }
    input[type] {
        font-family: monospace;
        font-size: 100%;
        color: rgb(56,56,56);
        background-color: inherit;
    }
    [type="text"], [type="number"] {
        margin-right: 13px;
        width: 222px;
    }

    [type="checkbox"].filter {
        margin-left: 4px
    }
    [type="checkbox"] {
        margin-left: 13px
    }
    [type="button"], [type="submit"] {
        font-size: 1em;
        margin: 0 2px 0 13px;
    }

    fieldset {
        font-size: 75%;
        margin: 20px 10px;
    }
    fieldset input[type="text"], fieldset input[type="number"] {
        width: 95% !important;
        margin: 0;
        display: block;
    }
    fieldset.expanded table.hidden {
        display: none;
    }

    table.disabled {
        color: rgb(176,176,176);
    }
    -->
</style>
</head>

<body>
<div class='content'>
<noscript>
    <p>This page requires javascript be enabled.</p>
</noscript>

<hr>
<h1>Flix CGI Sample</h1>

<p><small>flix2_sample.cgi version 1.9</small></p>
<h4>Instructions</h4>
<ul>
<li>In this sample you must choose a source file and an output file.<br>
        If you leave all the other options blank then the sample will not call
        the corresponding Flix Engine function and the default will be used.<br>
        When done please press the "Start Encode" button at the bottom of the page.<br>

```

```

    <li>Mouse over a function name to see its default, if applicable.
    <li>Current source file directory: <cfoutput>#indir#</cfoutput>
</ul>

<p>
    <label><input type="checkbox" id='showall' onclick='showall(this.checked)'/>Show
        all</label>
</p>

<form action="process_sample.cfm" method="post">

<!-- ##SOURCE FILE##### -->

<hr>
<fieldset class='expanded' id="srcfile">
<legend onclick='toggle_expand(this)'/>Source File</legend>
<table>

<tr>
<td>
<cfif DirectoryExists(indir)>
<cfdirectory
    directory="#indir#"
    name="DIR"
    sort="directory ASC">
<cfoutput>
<select name="setInputFile"
    size=#IIf(DIR.recordcount LT 15,'#DIR.recordcount#','15')#>
</cfoutput>
    <cfoutput query="DIR">
        <option value="#indir##DIR.name#">#DIR.name#</option>
    </cfoutput>
</select>
<cfelse>
<cfoutput>WARNING couldn't open #indir#: directory does not
    exist/insufficient permissions.</cfoutput>
</cfif>

    </td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE##### -->
-->
<hr>
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)'/>Output File</legend>
<table>

<tr>
<td>
    <input type="text" name="setOutputFile" value="coldfusion-out.flv">
</td>
</tr>

</table>
</fieldset>

<!-- ##MAIN OPTIONS#####
## -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)'/>Main Options</legend>
<table>

```

```

<tr>
  <th><abbr title="Default: FALSE">setOverwriteExistingFiles</abbr></th>
  <td>
    <select name="setOverwriteExistingFiles">
      <option value=""></option>
      <option value="1">TRUE</option>
      <option value="0">FALSE</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="Default: TRUE">setExportAudio</abbr></th>
  <td>
    <select name="setExportAudio">
      <option value=""></option>
      <option value="1">TRUE</option>
      <option value="0">FALSE</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="Default: TRUE">setExportVideo</abbr></th>
  <td>
    <select name="setExportVideo">
      <option value=""></option>
      <option value="1">TRUE</option>
      <option value="0">FALSE</option>
    </select>
  </td>
</tr>

</table>
</fieldset>

<!-- ##CODECS#####
      ### -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this) '>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
  <td>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
        onfocus="set_vcodec_visible('vp8table') ">
      <abbr title="For use with WebM">FE2_CODEC_VP8</abbr>&nbsp;
    </label>
    <br>

    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
        onfocus="set_vcodec_visible('vp6table') ">
      <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr>&nbsp;
    </label>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
        onfocus="set_vcodec_visible('vp6atable') ">
      <abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr>&nbsp;
    </label>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_H263'
        onfocus="set_vcodec_visible('h263table') ">
      <abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr>&nbsp;

```



```

</label>
<br>

<label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
    onfocus="set_vcodec_visible('h263table')">
  <abbr title="For use with 3GP">FE2_CODEC_H263_BASELINE</abbr>&nbsp;
</label>
<label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_H264'
    onfocus="set_vcodec_visible('h264table')">
  <abbr title="For use with 3GP/3G2/MOV/MP4">FE2_CODEC_H264</abbr>&nbsp;
</label>

<!-- VP6 codec parameters -->
<table id='vp6table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_VP6_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_KFINTTYPE">
        <option value=""></option>
        <option value="0">MAX_KEYFRAMES</option>
        <option value="1">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE
      2_VP6_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP6_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_RC_MODE">
        <option value=""></option>
        <option value="3">VBR_2PASSControl</option>
        <option value="2">CBR_2PASSControl</option>
        <option value="1">VBR_1PASSControl</option>
        <option value="0">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_CXMODE">
        <option value=""></option>
        <option value="0">COMPRESSMODE_GOOD</option>
        <option value="1">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: VP6_E">FE2_VP6_PROFILE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_PROFILE">
        <option value=""></option>

```

```

        <option value="0">VP6_E</option>
        <option value="1">VP6_S</option>
    </select>
</td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
    <th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_CONCURRENCY'></td>
</tr>

<tr>
    <th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_UNDERSHOOT_PCT'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MIN
        _Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_MIN_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MAX
        _Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_SHA
        RPNES</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_SHARPNESS'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_NOI
        SE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_NOISE_REDUCTION'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEM
        PORAL_RESAMPLING</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEM
        PORAL_DOWN_WATERMARK</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'><
        /td>
</tr>

<tr>
    <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
    <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PREBUFFER'></td>
</tr>

```

```
|  |  |
| --- | --- |
| <abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td> | |
| <abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6_STREAM_MAX_BUFFER'></td> | |
| <abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6_2PASS_MIN_SECTION'></td> | |
| <abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6_2PASS_MAX_SECTION'></td> | |


</table>
<!-- END - VP6 codec parameters -->

<!-- VP6A codec parameters -->
<table id='vp6atable' class='hidden'>
| <abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_BITRATE'></td> | |
| <abbr title="Default: 68kbps (15% of default 448kbps)">FE2_VP6A_ALPHA_BITRATE</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_ALPHA_BITRATE'></td> | |
| <abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>  <select name="codec:setParam:FE2_VP6A_KFINTTYPE"> <option value=""></option> <option value="0">MAX_KEYFRAMES</option> <option value="1">FIXED_KEYFRAMES</option> </select> | |
| <abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP6A_KFFREQ</abbr></th>  <input type='text' name='codec:setParam:FE2_VP6A_KFFREQ'></td> | |
| <abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>  <select name="codec:setParam:FE2_VP6A_RC_MODE"> <option value=""></option> <option value="3">VBR_2PASSControl</option> <option value="2">CBR_2PASSControl</option> <option value="1">VBR_1PASSControl</option> <option value="0">CBR_1PASSControl</option> </select> | |

```

```

<tr>
  <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>
  <td>
    <select name="codec:setParam:FE2_VP6A_CXMODE">
      <option value=""></option>
      <option value="0">COMPRESSMODE_GOOD</option>
      <option value="1">COMPRESSMODE_BEST</option>
    </select>
  </td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
  <th><abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
</tr>

```

```

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_RESAMPLING</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_DOWN_WATERMARK</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
  <th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MIN_SECTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MAX_SECTION'></td>
</tr>
</table>
<!-- END - VP6A codec parameters -->

<!-- H263 codec parameters -->
<table id='h263table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H263_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_H263_KFINTTYPE">
        <option value=""></option>
        <option value="0">MAX_KEYFRAMES</option>
        <option value="1">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>
</table>

```

```

</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE
  2_H263_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_KFFREQ'></td>
</tr>

<tr>
<th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>
<td>
<select name="codec:setParam:FE2_H263_RC_MODE">
<option value=""></option>
<option value="3">VBR_2PASSControl</option>
<option value="2">CBR_2PASSControl</option>
<option value="1">VBR_1PASSControl</option>
<option value="0">CBR_1PASSControl</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MAX_Q'></td>
</tr>

<tr>
<th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MIN_Q'></td>
</tr>

</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
<tr>
<th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE
  2_H264_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_KFFREQ'></td>
</tr>

<tr>
<th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
<td>
<select name="codec:setParam:FE2_H264_RC_MODE">
<option value=""></option>
<option value="1">VBR_1PASSControl</option>
<option value="0">CBR_1PASSControl</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
<td>
<select name="codec:setParam:FE2_H264_PROFILE">
<option value=""></option>
<option value="0">BASE_H264PROFILE</option>
<option value="1">MAIN_H264PROFILE</option>
<option value="2">HIGH_H264PROFILE</option>
</select>

```

```

    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H264_B_FRAME_RATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: Dependent on profile selection, see API docs. Valid
      Range [0,5]">FE2_H264_SPEED</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H264_SPEED'></td>
  </tr>
</table>
<!-- END - H264 codec parameters -->

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_KFINTTYPE">
        <option value=""></option>
        <option value="0">MAX_KEYFRAMES</option>
        <option value="1">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE
      2_VP8_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_RC_MODE">
        <option value=""></option>
        <option value="3">VBR_2PASSControl</option>
        <option value="2">CBR_2PASSControl</option>
        <option value="1">VBR_1PASSControl</option>
        <option value="0">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_CXMODE">
        <option value=""></option>
        <option value="0">COMPRESSMODE_GOOD</option>
        <option value="1">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>

```

```

    <td><input type='text' name='codec:setParam:FE2_VP8_THREADS'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_PROFILE'></td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_LAG'></td>
</tr>

<tr>
  <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_UNDERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_OVERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
  <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

```



```

<tr>
  <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_ALTREF'></td>
</tr>

<tr>
  <th><abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_AR_MAX_FRAMES'></td>
</tr>

<tr>
  <th><abbr title="">FE2_VP8_AR_TYPE</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_AR_TYPE'></td>
</tr>

<tr>
  <th><abbr title="">FE2_VP8_AR_STRENGTH</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_AR_STRENGTH'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MB_STATIC_THRESHOLD'></td>
</tr>

<tr>
  <th><abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_TOKEN_PARTITIONS'></td>
</tr>

</table>
<!-- END - VP8 codec parameters -->

</td>
</tr> <!-- END - video codecs -->

<tr><th><b>Audio Codecs</b></th></tr>
<tr>
  <td>
    <label>
      <input type='radio' name='acodec:' value='FE2_CODEC_VORBIS'
        onfocus="set_acodec_visible('vorbistable') ">
      <abbr title="For use with WebM">FE2_CODEC_VORBIS</abbr>&nbsp;
    </label>
    <br>

    <label>
      <input type='radio' name='acodec:' value='FE2_CODEC_AAC'
        onfocus="set_acodec_visible('aactable') ">
      <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AAC</abbr>&nbsp;
    </label>
    <label>
      <input type='radio' name='acodec:' value='FE2_CODEC_AACPLUS'
        onfocus="set_acodec_visible('aacplustable') ">
      <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AACPLUS</abbr>&nbsp;

```

```

</label>
<label>
  <input type='radio' name='acodec:' value='FE2_CODEC_LAME'
    onfocus="set_acodec_visible('lametable') ">
  <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_LAME</abbr>&nbsp;
</label>
<label>
  <input type='radio' name='acodec:' value='FE2_CODEC_PCM'
    onfocus="set_acodec_visible('pcmtable') ">
  <abbr title="For use with FLV/SWF">FE2_CODEC_PCM</abbr>&nbsp;
</label>
<br>

<label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AMR_NB'
    onfocus="set_acodec_visible('amrnhtable') ">
  <abbr title="For use with 3GP">FE2_CODEC_AMR_NB</abbr>&nbsp;
</label>

<!-- AMR_NB codec parameters -->
<table id='amrnhtable' class='hidden'>
  <tr>
    <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_AMR_BITRATE'></td>
  </tr>
</table>
<!-- END - AMR_NB codec parameters -->

<!-- AAC codec parameters -->
<table id='aactable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_AAC_BITRATE'></td>
  </tr>
</table>
<!-- END - AAC codec parameters -->

<!-- AACPLUS codec parameters -->
<table id='aacplustable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_AACPLUS_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: disabled (aacPlus v1)">FE2_AACPLUS_PARAMETRIC_STEREO
      </abbr></th>
    <td>
      <select name="codec:setParam:FE2_AACPLUS_PARAMETRIC_STEREO">
        <option value=""></option>
        <option value="0">disable (aacPlus v1)</option>
        <option value="1">enable (aacPlus v2)</option>
      </select>
    </td>
  </tr>
</table>
<!-- END - AACPLUS codec parameters -->

<!-- LAME codec parameters -->
<table id='lametable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_LAME_BITRATE'></td>
  </tr>

```

```

<tr>
  <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_LAME_QUALITY'></td>
</tr>

<tr>
  <th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
  <td>
    <select name="codec:setParam:FE2_LAME_RC_MODE">
      <option value=""></option>
      <option value="0">LAME_CBR</option>
      <option value="1">LAME_ABR</option>
      <option value="2">LAME_VBR_rh</option>
      <option value="3">LAME_VBR_mtrh</option>
    </select>
  </td>
</tr>
</table>
<!-- END - LAME codec parameters -->

<!-- PCM codec parameters -->
<table id='pcmtable' class='hidden'>
  <tr>
    <th>(FE2_CODEEC_PCM defines no parameters)</th>
  </tr>
</table>
<!-- END - PCM codec parameters -->

<!-- VORBIS codec parameters -->
<table id='vorbistable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VORBIS_BITRATE'></td>
  </tr>
</table>
<!-- END - VORBIS codec parameters -->

</td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS#####
    ### -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this)''>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_CUT' value='avfilter:'
        onchange="toggle_ftable('filter_cut',this.checked)">
      FE2_FILTER_CUT
    </label>

    <table id='filter_cut' class='disabled'>
      <tr>
        <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CUT_START_SEC'></td>
      </tr>

```

```
|  |  |  |  | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>  <td><input type='text' name='filter:setParam:FE2_CUT_STOP_SEC'></td> </tr>  |  |  | | --- | --- | | <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>  <td> <select name="filter:setParam:FE2_CUT_USE_SEEK"> <option value=""></option> <option value="0">FALSE</option> <option value="1">TRUE</option> </select> </td> </tr>  </table>  </td> </tr> <!-- END - CUT filter parameters -->  <tr><th><b>Video Filters</b></th></tr>  <tr> <td> <label> <input type='checkbox' class='filter' name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:' onchange="toggle_ftable('filter_adaptive_deinterlace',this.checked)"> FE2_FILTER_ADAPTIVE_DEINTERLACE </label>  <table id='filter_adaptive_deinterlace' class='disabled'> <tr> <th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</ab br></th> <td> <select name="filter:setParam:FE2_ADAPTIVE_DEINTERLACE_MODE"> <option value=""></option> <option value="0">DEINTERLACE_NONE</option> <option value="1">DEINTERLACE_1_2_1_BLUR</option> <option value="2">DEINTERLACE_DROP_FIELD</option> <option value="3">DEINTERLACE_ADAPTIVE</option> </select> </td> </tr> </table>  </td> </tr> <!-- END - ADAPTIVE DEINTERLACE filter parameters -->  <tr> <td> <label> <input type='checkbox' class='filter' name='FE2_FILTER_BCHS' value='vfilter:' onchange="toggle_ftable('filter_bchs',this.checked)"> FE2_FILTER_BCHS </label>  <table id='filter_bchs' class='disabled'> <tr> <th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th> <td><input type='text' name='filter:setParam:FE2_BCHS_BRIGHTNESS'></td> | | | |

```

```

    </tr>
    <tr>
      <th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
      <td><input type='text' name='filter:setParam:FE2_BCHS_CONTRAST'></td>
    </tr>
    <tr>
      <th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
      <td><input type='text' name='filter:setParam:FE2_BCHS_HUE'></td>
    </tr>
    <tr>
      <th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
      <td><input type='text' name='filter:setParam:FE2_BCHS_SATURATION'></td>
    </tr>
  </table>

</td>
</tr>
<!-- END - BCHS filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_BLUR' value='vfilter:'
        onchange="toggle_ftable('filter_blur',this.checked)">
      FE2_FILTER_BLUR
    </label>

    <table id='filter_blur' class='disabled'>
      <tr>
        <th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
        <td>
          <select name="filter:setParam:FE2_BLUR_FILTER">
            <option value=""></option>
            <option value="0">BLUR_LOWPASS</option>
            <option value="1">BLUR_GAUSS</option>
          </select>
        </td>
      </tr>

      <tr>
        <th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
        <td>
          <select name="filter:setParam:FE2_BLUR_MASKSIZE">
            <option value=""></option>
            <option value="0">MASK_3x3</option>
            <option value="1">MASK_5x5</option>
          </select>
        </td>
      </tr>
    </table>

  </td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_CROP' value='vfilter:'
        onchange="toggle_ftable('filter_crop',this.checked)">
      FE2_FILTER_CROP
    </label>

    <table id='filter_crop' class='disabled'>
      <tr>

```

```

        <th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CROP_TOP'></td>
    </tr>

    <tr>
        <th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CROP_BOTTOM'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CROP_LEFT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CROP_RIGHT'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - CROP filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_DENOISE' value='vfilter:'
                onchange="toggle_ftable('filter_denoise',this.checked)">
            FE2_FILTER_DENOISE
        </label>

        <table id='filter_denoise' class='disabled'>
            <tr>
                <th><abbr title="Default: 0. Range: [0.0,1.0]">FE2_DENOISE_NOISE_LEVEL</abbr>
                </th>
                <td><input type='text' name='filter:setParam:FE2_DENOISE_NOISE_LEVEL'></td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_FRAMERATE' value='vfilter:'
                onchange="toggle_ftable('filter_framerate',this.checked)">
            FE2_FILTER_FRAMERATE
        </label>

        <table id='filter_framerate' class='disabled'>
            <tr>
                <th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAM
                    ERATE_DECIMATE</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_FRAMERATE_DECIMATE'></td>
            </tr>

            <tr>
                <th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FR
                    AMERATE_FPS</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_FRAMERATE_FPS'></td>
            </tr>
        </table>
    </td>
</tr>

```

```

</td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_MIRROR' value='vfilter:'
onchange="toggle_ftable('filter_mirror',this.checked)">
FE2_FILTER_MIRROR
</label>

<table id='filter_mirror' class='disabled'>
<tr>
<th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
<td>
<select name="filter:setParam:FE2_MIRROR_HORIZONTAL">
<option value=""></option>
<option value="0">FALSE</option>
<option value="1">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
<td>
<select name="filter:setParam:FE2_MIRROR_VERTICAL">
<option value=""></option>
<option value="0">FALSE</option>
<option value="1">TRUE</option>
</select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_OVERLAY' value='vfilter:'
onchange="toggle_ftable('filter_overlay',this.checked)">
FE2_FILTER_OVERLAY
</label>

<table id='filter_overlay' class='disabled'>
<tr>
<th>
<abbr title="Currently searching <cfoutput>#overlaydir#</cfoutput> for overl
ay images">FE2_OVERLAY_FILE</abbr>
</th>
<td>
<cfif DirectoryExists(overlaydir)>
<cfdirectory
directory="#overlaydir#"
name="DIR"
sort="directory ASC">
<cfoutput>
<select name="FE2_OVERLAY_FILE"
size=#IIf(DIR.recordcount LT 15,'#DIR.recordcount#','15')#>
<option value=""></option>

```

```

</cfoutput>
  <cfoutput query="DIR">
    <option value="#overlaydir##DIR.name#">#DIR.name#</option>
  </cfoutput>
</select>
<cfelse>
<cfoutput>WARNING couldn't open #overlaydir#: directory does not
    exist/insufficient permissions.</cfoutput>
</cfif>

  </td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
<td>
  <select name="filter:setParam:FE2_OVERLAY_MASK_XY">
    <option value=""></option>
    <option value="0">FALSE</option>
    <option value="1">TRUE</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_Y'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
<td>
  <select name="filter:setParam:FE2_OVERLAY_MASK_RGB">
    <option value=""></option>
    <option value="0">FALSE</option>
    <option value="1">TRUE</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_R'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_G'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_B'></td>
</tr>

<tr>
<th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
<td>
  <select name="filter:setParam:FE2_OVERLAY_POS">
    <option value=""></option>
    <option value="0">FE2_OVERLAY_POS_MODE_TOPLEFT</option>
    <option value="1">FE2_OVERLAY_POS_MODE_BOTLEFT</option>
    <option value="2">FE2_OVERLAY_POS_MODE_CENTER</option>
    <option value="3">FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
    <option value="4">FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
  </select>
</td>
</tr>

```



```

        <option value="5">FE2_OVERLAY_POS_MODE_XY</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_Y'></td>
</tr>

</table>

</td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_PNGEX' value='vfilter:'
onchange="toggle_ftable('filter_pngex',this.checked)">
FE2_FILTER_PNGEX
</label>

<table id='filter_pngex' class='disabled'>
<tr>
<th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'></td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_PREFIX'>
</td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'>
</td>
</tr>

<tr>
<th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_WIDTH'></td>
</tr>

<tr>
<th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_HEIGHT'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>

<td>
<select name="filter:setParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">
<option value=""></option>
<option value="0">FALSE</option>
<option value="1">TRUE</option>

```

```

        </select>
      </td>
    </tr>

    <tr>
      <th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
      <td>
        <select name="filter:setParam:FE2_PNGEX_ENABLE_ALPHA">
          <option value=""></option>
          <option value="0">FALSE</option>
          <option value="1">TRUE</option>
        </select>
      </td>
    </tr>

    <tr>
      <th><abbr title="interval in ms; Default: disabled">
        FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
      <td><input type='text' name='filter:setParam:FE2_PNGEX_EXPORT_INTERVAL'></td>
    </tr>

    <tr>
      <th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">
        FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
      <td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_EXPORT_TIME_STRIN
        G'></td>
    </tr>

    <tr>
      <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
      <td>
        <select name="filter:setParam:FE2_PNGEX_EXPORT_CUE_POINTS">
          <option value=""></option>
          <option value="3">All cue points (FE2_PNGEX_CP_ALL)</option>
          <option value="2">Only navigation cue points (FE2_PNGEX_CP_NAV)</option>
          <option value="1">Only event cue points (FE2_PNGEX_CP_EVENT)</option>
        </select>
      </td>
    </tr>

    <tr>
      <th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">
        FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>
      <td><input type='text' name='filter:setParam:FE2_PNGEX_COMPRESSION_LEVEL'></td>
    </tr>

    <tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>

    <tr>
      <th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>
      <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></td>
    </tr>

    <tr>
      <th><abbr title="start time in ms; Default: 0">
        FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
      <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_START_TIME
        '></td>
    </tr>

    <tr>
      <th><abbr title="stop time in ms; Default: <clip length>">
        FE2_PNGEX_AUTO_EXPORT_END_TIME</abbr></th>
      <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_END_TIME'>

```

```

        </td>
    </tr>

    <tr>
        <th><abbr title="Default: 0">FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_RANDOM_PER
            IOD'></td>
    </tr>

</table>

</td>
</tr>
<!-- END - PNGEX filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_ROTATE' value='vfilter:'
                onchange="toggle_ftable('filter_rotate',this.checked)">
            FE2_FILTER_ROTATE
        </label>

        <table id='filter_rotate' class='disabled'>
            <tr>
                <th><abbr title="Default: 0. valid: {0,90,180,270}">FE2_ROTATE_ANGLE</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_ROTATE_ANGLE'></td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - ROTATE filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_SCALE' value='vfilter:'
                onchange="toggle_ftable('filter_scale',this.checked)">
            FE2_FILTER_SCALE
        </label>

        <table id='filter_scale' class='disabled'>
            <tr>
                <th><abbr title="Default: input image width">FE2_SCALE_WIDTH</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_SCALE_WIDTH'></td>
            </tr>

            <tr>
                <th><abbr title="Default: input image height">FE2_SCALE_HEIGHT</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_SCALE_HEIGHT'></td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - SCALE filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_SHARPEN' value='vfilter:'
                onchange="toggle_ftable('filter_sharpen',this.checked)">

```

```

    FE2_FILTER_SHARPEN
  </label>

  <table id='filter_sharpen' class='disabled'>
    <tr>
      <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
    </tr>
  </table>

</td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_HIGHPASS' value='afilter:'
        onchange="toggle_ftable('filter_highpass',this.checked)">
      FE2_FILTER_HIGHPASS
    </label>

    <table id='filter_highpass' class='disabled'>
      <tr>
        <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_HIGHPASS_Q'></td>
      </tr>

      <tr>
        <th>FE2_HIGHPASS_CUTOFF</th>
        <td><input type='text' name='filter:setParam:FE2_HIGHPASS_CUTOFF'></td>
      </tr>
    </table>

  </td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_LOWPASS' value='afilter:'
        onchange="toggle_ftable('filter_lowpass',this.checked)">
      FE2_FILTER_LOWPASS
    </label>

    <table id='filter_lowpass' class='disabled'>
      <tr>
        <th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_LOWPASS_Q'></td>
      </tr>

      <tr>
        <th>FE2_LOWPASS_CUTOFF</th>
        <td><input type='text' name='filter:setParam:FE2_LOWPASS_CUTOFF'></td>
      </tr>
    </table>

  </td>
</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
  <td>

```

```

<label>
  <input type='checkbox' class='filter'
        name='FE2_FILTER_RESAMPLE' value='afilter:'
        onchange="toggle_ftable('filter_resample',this.checked)">
    FE2_FILTER_RESAMPLE
</label>

<table id='filter_resample' class='disabled'>
  <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_RATE</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_RESAMPLE_RATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_CHANNELS</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_RESAMPLE_CHANNELS'></td>
  </tr>
</table>

</td>
</tr>
<!-- END - RESAMPLE filter parameters -->

</table>
</fieldset>

<!-- ##MUXERS#####
      ### -->
<hr>
<fieldset class='collapsed' id="muxers">
<legend onclick='toggle_expand(this)'>Muxers</legend>
<table>

<tr>
<td>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3GP'
        onfocus="set_muxer_visible('tgptable')">
    FE2_MUXER_3GP&nbsp;
</label>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3G2'
        onfocus="set_muxer_visible('tg2table')">
    FE2_MUXER_3G2&nbsp;
</label>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MOV'
        onfocus="set_muxer_visible('movtable')">
    FE2_MUXER_MOV&nbsp;
</label>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MP4'
        onfocus="set_muxer_visible('mp4table')">
    FE2_MUXER_MP4&nbsp;
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FLV'
        onfocus="set_muxer_visible('flvtable',true)">
    FE2_MUXER_FLV&nbsp;
</label>
<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_SWF'
        onfocus="set_muxer_visible('swftable',true)">
    FE2_MUXER_SWF
</label>

```

```

<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
    onfocus="set_muxer_visible('fxmtable',true)">
  FE2_MUXER_FXM&nbsp;
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
    onfocus="set_muxer_visible('webmtable',true)">
  FE2_MUXER_WEBM&nbsp;
</label>
</td>
</tr>

<tr>
  <td>
    <!-- 3GP muxer parameters -->
    <table id='tgptable' class='hidden'>
      <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
          <select name="muxer:setParam:FE2_3GP_FASTSTART">
            <option value=""></option>
            <option value="0">FALSE</option>
            <option value="1">TRUE</option>
          </select>
        </td>
      </tr>
    </table>
    <!-- END - 3GP muxer parameters -->

    <!-- 3G2 muxer parameters -->
    <table id='tg2table' class='hidden'>
      <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
          <select name="muxer:setParam:FE2_3G2_FASTSTART">
            <option value=""></option>
            <option value="0">FALSE</option>
            <option value="1">TRUE</option>
          </select>
        </td>
      </tr>
    </table>
    <!-- END - 3G2 muxer parameters -->

    <!-- FLV muxer parameters -->
    <table id='flvtable' class='hidden'>
      <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'"
          >
          FE2_FLV_CUEPT_EVENT</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
      </tr>

      <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.
          0'"
          >
          FE2_FLV_CUEPT_NAV</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_NAV'></td>
      </tr>

      <tr>
        <th><abbr title="e.g. 'cuept_name& n0=v0& n1=v1...'"

```

```

        FE2_FLV_CUEPT_PARAM</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
    </tr>

    <tr>
        <th><abbr title="Select specific metadata entries to enable. Default for each
            item is provided.">
                FE2_FLV_METADATA_ENABLE</abbr></th>
        <td>
            <table id='flv_metadata_enable' class=''>
                <tr>
                    <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr
                        title="Default: Enabled">MD_DURATION</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                        DURATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' value='0'></t
                        d>
                    </tr>

                    <tr>
                        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr
                            title="Default: Enabled">MD_DATASIZE</abbr></label></th>
                        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                            DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' value='1'></t
                            d>
                        </tr>

                        <tr>
                            <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><ab
                                br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
                            <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                                AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' value='2'
                                ></td>
                            </tr>

                            <tr>
                                <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><ab
                                    br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
                                <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                                    VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' value='3'
                                    ></td>
                                </tr>

                                <tr>
                                    <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'
                                        ><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
                                    <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                                        AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' v
                                        alue='4'></td>
                                    </tr>

                                    <tr>
                                        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'
                                            ><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
                                        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                                            VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' v
                                            alue='5'></td>
                                        </tr>

                                        <tr>
                                            <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'
                                                ><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
                                            <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
                                                AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' val
                                                ue='6'></td>
                                            </tr>

                                            <tr>

```

```

        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'>
<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' val
ue='7'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr ti
tle="Default: Enabled">MD_WIDTH</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' value='8'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' value='9'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' value='10'>
</td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' value
='11'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'>
<abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' val
ue='12'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIM
ESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></
th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRA
METIMESTAMP' value='13'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOC
ATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAM
ELOCATION' value='14'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abb
r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_

```



```

KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' value='15'>
</td>
</tr>
</table>
</td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
FE2_FLV_METADATA_DISABLE</abbr></th>
<td>
<table id='flv_metadata_disable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abbr
r title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_DURATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' value='0'><
/td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abbr
r title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' value='1'><
/td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><a
bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' value='
2'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><a
bbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' value='
3'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE
'><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'
value='4'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE
'><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'
value='5'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID'
><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' v
alue='6'></td>

```

```

</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID'
><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' v
alue='7'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr t
itle="Default: Enabled">MD_WIDTH</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_WIDTH' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' value='8'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr
title="Default: Enabled">MD_HEIGHT</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' value='9'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><ab
br title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' value='10
'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'>
<abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' val
ue='11'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP'
><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' v
alue='12'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMEETI
MESTAMP'><abbr title="Default: Enabled">MD_LASTKEYFRAMEETIMESTAMP</abbr></label></
th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTKEYFRAMEETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYF
RAMETIMESTAMP' value='13'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Enabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='14'></td>
</tr>

```

```

        <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><abbr
        title="Default: Enabled">MD_KEYFRAMES</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_
        _KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES' value='15
        '></td>
        </tr>
    </table>
</td>
</tr>
</table>
<!-- END - FLV muxer parameters -->

<!-- FXM muxer parameters -->
<table id='fxmtable' class='hidden'>
    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0' "
        >
            FE2_FXM_CUEPT_EVENT</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_EVENT'></td>
    </tr>

    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.
        0' ">
            FE2_FXM_CUEPT_NAV</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_NAV'></td>
    </tr>

    <tr>
        <th><abbr title="e.g. 'cuept_name& n0=v0& n1=v1...' ">
            FE2_FXM_CUEPT_PARAM</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM'></td>
    </tr>

    <tr>
        <th><abbr title="Select specific metadata entries to enable. Default for each
        item is provided.">
            FE2_FXM_METADATA_ENABLE</abbr></th>
        <td>
            <table id='fxm_metadata_enable' class=''>
                <tr>
                    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr
                    title="Default: Enabled">MD_DURATION</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
                    DURATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION' value='0'></t
                    d>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr
                    title="Default: Enabled">MD_DATASIZE</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
                    DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' value='1'></t
                    d>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><ab
                    br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
                    AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' value='2'
                    ></td>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><ab

```

```

br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' value='3'
></td>
</tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE'
><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' v
alue='4'></td>
  </tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE'
><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' v
alue='5'></td>
  </tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'>
<abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' val
ue='6'></td>
  </tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'>
<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' val
ue='7'></td>
  </tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr ti
tle="Default: Enabled">MD_WIDTH</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' value='8'></td>
  </tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' value='9'></td>
  </tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' value='10'>
</td>
  </tr>

  <tr>
    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' value
='11'></td>
  </tr>

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'> <abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' val ue='12'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIM ESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRA METIMESTAMP' value='13'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOC ATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAM ELOCATION' value='14'></td> </tr>  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abb r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_ KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' value='15'> </td> </tr> </table> </td> </tr>  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | <th><abbr title="Select specific metadata entries to enable. Default for each item is provided."> FE2_FXM_METADATA_DISABLE</abbr></th>  <table id='fxm_metadata_disable' class=''> |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abb r title="Default: Enabled">MD_DURATION</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_ _DURATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' value='0'> </td> </tr>  |  |  |  |  | | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abb r title="Default: Enabled">MD_DATASIZE</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_ _DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' value='1'> </td> </tr>  |  |  | | --- | --- | | <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><a bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_ _AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' value=' 2'></td> </tr> | | | | | | | | | | | | | | | |

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' value='3'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE' value='4'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE' value='5'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' value='6'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' value='7'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabled">MD_WIDTH</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' value='8'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enabled">MD_HEIGHT</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' value='9'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: Enabled">MD_FRAMERATE</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' value='10'></td>   <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th> | | | | | | | | | | | | | | | | |

```

```

        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' val
ue='11'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'
><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' v
alue='12'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETI
MESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYF
RAMETIMESTAMP' value='13'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='14'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><ab
br title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' value='15
'></td>
    </tr>
</table>
</td>
</tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>
    <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_MOV_FASTSTART">
                <option value=""></option>
                <option value="0">FALSE</option>
                <option value="1">TRUE</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - MOV muxer parameters -->

<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
    <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_MP4_FASTSTART">
                <option value=""></option>
                <option value="0">FALSE</option>
            </select>
        </td>
    </tr>
</table>

```

```

        <option value="1">TRUE</option>
    </select>
</td>
</tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
    <tr>
        <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_WIDTH'></td>
    </tr>

    <tr>
        <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_HEIGHT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_FRAMERATE'></td>
    </tr>

    <tr>
        <th>FE2_SWF_LOOP_COUNT</th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_LOOP_COUNT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
    </tr>

    <tr>
        <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
        <td>
            <select name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>
                <option value=""></option>
                <option value="_self">_self</option>
                <option value="_blank">_blank</option>
                <option value="_parent">_parent</option>
                <option value="_top">_top</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</a
            bbr></th>
        <td>
            <select name='muxer:setParam:FE2_SWF_EMBEDDED_URL_TYPE'>
                <option value=""></option>
                <option value="0">EmbeddedUrlIsGetUrl</option>
                <option value="1">EmbeddedUrlIsLoadMovie</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="e.g. n0=v0& n1=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
    </tr>

    <tr><th><b>Preloader Settings:</b></th><td></td></tr>

    <tr>

```



```

<th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
<td>
  <select name='muxer:setParam:FE2_SWF_PRELOAD_TYPE'>
    <option value=""></option>
    <option value="-1">SwfPreloaderNone</option>
    <option value="0">SwfFixedPreloader</option>
    <option value="1">SwfAdaptivePreloader</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
<th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTION</abbr></th>
<td>
  <select name='muxer:setParam:FE2_SWF_ON_START_OPTION'>
    <option value=""></option>
    <option value="8">SwfOnMovieStartAutomatically</option>
    <option value="16">SwfOnMovieStartOnClick</option>
    <option value="32">SwfOnMovieStartWait</option>
    <option value="64">SwfOnMovieStartEmbedSTOP</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_START_BLANK_FRAME'></td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_START_WAIT_SEC'></td>
</tr>

<tr><th><b>End Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
<td>
  <select name='muxer:setParam:FE2_SWF_ON_END_OPTION'>
    <option value=""></option>
    <option value="-1">SwfOnMovieEndNothing</option>
    <option value="0">SwfOnMovieEndSTOP</option>
    <option value="1">SwfOnMovieEndLoop</option>
    <option value="2">SwfOnMovieEndUnload</option>
    <option value="4">SwfOnMovieEndLoadMovie</option>
  </select>
</td>
</tr>

<tr>

```

```

        <th>FE2_SWF_ON_END_URL</th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ON_END_URL'></td>
    </tr>
</table>
<!-- END - SWF muxer parameters -->

<!-- WEBM muxer parameters -->
<table id='webmtable' class='hidden'>
    <tr>
        <th>(FE2_MUXER_WEBM defines no parameters)</th>
    </tr>
</table>
<!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM##### -->
<hr>
<p>
    <input type="submit" value="Start Encode">
    <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>

```

6.8.2 process_sample.cfm

```

<cfscript>
//
//=====
//
// Copyright (c) On2 Technologies Inc. All Rights Reserved.
//
//-----
//
// File:          $Workfile$
//               $Revision$
//
// Last Update:  $DateUTC$
//
//-----
//

flix = CreateObject("COM", "On2.FlixEngine");
prefix = "C:\Inetpub\";
outdir = "#prefix#flixmedia\out\"; //destination file directory
exceptionOccurred = false;

function flush_output() {
    GetPageContext().getOut().flush();
}

function print_encoder_status() {
    WriteOutput("<p>Encoder Status<br>");

    res = flix.getEncoderState();
    WriteOutput("&nbsp;flix.getEncoderState: #res#<br>");

    flixerr= flix.flixerrno();
    syserr= flix.syserrno();
    WriteOutput("&nbsp;flix.errno_: flixerrno:#flixerr# syserrno:#syserr#</p>");
}

```

```

function handle_exception(e) {
    WriteOutput("#e.Message#<br>#e.RootCause#<br>");
    WriteOutput("Tag Context:<br>");
    for(i=1;i LTE ArrayLen(e.TagContext);i=i+1) {
        tc=e.TagContext[i];
        WriteOutput("###i# #tc.ID# #tc.Template#: line #tc.Line#<br>");
    }
    //ensure script terminates after logging the failing function via process_hr
    exceptionOccurred = true;
}

function process_hr(funcname,sc) {
    WriteOutput("<td align=\"center\">#sc#</td>");
    if(sc NEQ 0 OR exceptionOccurred) {
        try {
            flixelr= flix.flixeerrno();
            syserr= flix.syserrno();
            WriteOutput("<td>flix.errno_: flixelrno:#flixeerr# syserrno:#syserr#</td>");
        } catch(Any e) {}
        ReleaseComObject(flix);
        abort();
    }
    WriteOutput("</tr>");
}

function SimpleSet(funcname,funcparm) {
    WriteOutput("<tr><td>flix.#funcname#( #funcparm# )</td>");

    try {
        Evaluate("flix.#funcname#(funcparm)");
    } catch(Any e) {
        handle_exception(e);
    }
    process_hr("flix.#funcname#",flix.sc());
}

function init_codec(name) {
    //if name is a codec name, e.g., FE2_CODEC_VP6, add an instance
    //we'll assume all IFlixPlgn::setParam's relate to this codec until we
    //hit the next codec name
    WriteOutput("<tr><td>flix.addCodec( #name# )</td>");

    try {
        codec= flix.addCodec(flix[name]);
    } catch(Any e) {
        handle_exception(e);
    }
    process_hr('flix.addCodec',flix.sc());
}

function codec_interface(funcname,name,value) {
    WriteOutput("<tr><td>codec.#funcname#( #name#, #value# )</td>");

    try {
        Evaluate("codec.#funcname#(flix[name],value)");
    } catch(Any e) {
        handle_exception(e);
    }
    process_hr("codec.#funcname#",codec.sc());
}

function init_filter(name) {
    //if name is a filter name, e.g., FE2_FILTER_CUT, add an instance
    //we'll assume all IFlixPlgn::setParam's relate to this filter until we
    //hit the next filter name

```

```

WriteOutput("<tr><td>flix.addFilter( #name# )</td>");

try {
    filter= flix.addFilter(flix[name]);
} catch(Any e) {
    handle_exception(e);
}
process_hr('flix.addFilter',flix.sc());
}

function filter_interface(funcname,name,value) {
    WriteOutput("<tr><td>filter.#funcname#( #name#, #value# )</td>");

    try {
        Evaluate("filter.#funcname#(flix[name],value)");
    } catch(Any e) {
        handle_exception(e);
    }
    process_hr("filter.#funcname#",filter.sc());
}

function init_muxer(name) {
    //if name is a muxer name, e.g., FE2_MUXER_FLV, add an instance
    //we'll assume all IFlixPlgn::setParam's relate to this muxer until we
    //hit the next muxer name
    WriteOutput("<tr><td>flix.addMuxer( #name# )</td>");

    try {
        muxer= flix.addMuxer(flix[name]);
    } catch(Any e) {
        handle_exception(e);
    }
    process_hr('flix.addMuxer',flix.sc());
}

function muxer_interface(funcname,name,value) {
    WriteOutput("<tr><td>muxer.#funcname#( #name#, #value# )</td>");

    try {
        Evaluate("muxer.#funcname#(flix[name],value)");
    } catch(Any e) {
        handle_exception(e);
    }
    process_hr("muxer.#funcname#",muxer.sc());
}

function encode() {
    /*
     * Start the Encode
     */
    WriteOutput("<tr><td>flix.encode()</td>");
    try {
        flix.encode();
    } catch(Any e) {
        handle_exception(e);
    }
    process_hr("flix.encode",flix.sc());

    /*
     * Retrieve the encoding status interface
     */
    WriteOutput("<tr><td>flix.encodingStatus()</td>");
    try {
        encstatus= flix.encodingStatus();
    } catch(Any e) {
        handle_exception(e);
    }
}

```

```

process_hr("flix.encodingStatus",flix.sc());

WriteOutput("</table>");

try {
    WriteOutput("<p>Encoding...(video frames encoded, percent complete).");
    WriteOutput(" Total frames will reset when doing 2 pass.<br>");
    thisThread = CreateObject("java", "java.lang.Thread");
    do {
        ier = flix.isEncoderRunning();
        tf = encstatus.totalFrames();
        p = encstatus.percentComplete();
        WriteOutput("#tf#, #p#%<br>");

        flush_output();
        thisThread.sleep(1000);
    } while(ier);
} catch(Any e) {
    handle_exception(e);
}

WriteOutput("<br>Done!</p>");
print_encoder_status();
}
</cfscript>

<cffunction name="abort" access="public" output="false" returntype="void">
    <cfabort/>
</cffunction>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Flix CGI Process Sample - ColdFusion</title>
</head>
<body>

<cfoutput>
    <hr>
    <p>process_sample.cfm version 1.0<br>
    Flix Engine COM library. Flix Engine v#flix.version()#
    COM v#flix.com_version()#<br>
    #REReplace(flix.copyright(),"\n","<br>","all")#</p>
</cfoutput>

<cfif NOT DirectoryExists(outdir)>
    <cfset i="<i>#outdir#</i>">
    <cfset tt="<tt>outdir</tt>">
    <cfoutput>
        <p>*****<br>
        <b>WARNING</b>: #i# MUST exist and be writeable by <i>flixengine_com.dll</i>
        <br>Please make #i# accessible or modify the #tt#
        value defined in #CGI.SCRIPT_NAME#.<br>
        *****</p>
    </cfoutput>
</cfif>

<table border='1' cellpadding='5'>
<caption>Flix Function Calls</caption>
<tr><th>Function Call</th><th>Return Value</th></tr>

<cfscript>
flush_output();
/*
    * Split the post data into name=value pairs.

```

```

* We are avoiding the Form variable as we are using the parameter names
* and values as references into the Flix Engine COM API. If only for
* presentation purposes, case will be significant.
*/
parm = ListToArray(URLDecode(GetHttpRequestData().content), "&");
for(i=1; i LTE ArrayLen(parm); i=i+1) {
    p = ListToArray(parm[i], "=");
    if(ArrayLen(p) EQ 2) {
        //WriteOutput("#p[1]# -> "&URLDecode(p[2])&"<br>");
        if(p[1] EQ 'setOutputFile') {
            SimpleSet(p[1],outdir&p[2]);
        } else if(ReFind("(?:a|v)codec:",p[1])) {
            init_codec(p[2]);
        } else if(ReFind("(?:a|v)filter:",p[1])) {
            init_filter(p[1]);
        } else if(ReFind("^muxer:",p[1])) {
            init_muxer(p[2]);
        } else {
            temp = ListToArray(p[1], ":");
            if(temp[1] EQ "codec") {
                codec_interface(temp[2],temp[3],p[2]);
            } else if(temp[1] EQ "filter") {
                filter_interface(temp[2],temp[3],p[2]);
            } else if(temp[1] EQ "muxer") {
                muxer_interface(temp[2],temp[3],p[2]);
            } else {
                SimpleSet(p[1],p[2]);
            }
        }
    }
}
}
}

encode();
ReleaseComObject(flix);
</cfscript>

</body>
</html>

```

6.9 Java

The Java wrapper for the [Flix Engine COM](#) interface is implemented using [com4j](#). This consists of 3 main parts:

- [com4j.dll](#) - provides native code to interface directly with the COM library (i.e., *flixengine_com.dll*)
- [com4j.jar](#) - interfaces used by the generated annotated interface
- [flixengine_com.jar](#) - [Flix Engine COM](#) wrapper interfaces generated by [com4j](#)

The jar files must be in the application's classpath. *com4j.dll* may be placed in the same directory as *com4j.jar* or in another location by setting the `java.library.path` system property. All of these files can be found within the Java sample install directory which by default is: `%PROGRAMFILES%\On2 Flix Engine (Demo)\samples\java`.

In addition to the above files the Java sample directory contains an archive named *flixengine_com-src.zip*. This archive holds the source generated by [com4j](#) as well as its associated Javadoc documentation.

Attention:

The com4j implementation binds directly to the vtable so, similar to the .NET languages and the associated assembly file, *flixengine_com.jar* will need to be updated should the [COM version change](#).

6.10 Command Line

```

/*
//=====
//
// Copyright (c) On2 Technologies Inc. All Rights Reserved.
//
//-----
//
// File:          $Workfile$
//               $Revision$
//
// Last Update:  $DateUTC$
//
//-----
*/
import com.on2.flixengine_com.*;

public class cli_encode {
    static IFlix flix;

    private static void init() {
        try {
            /*
             * Retrieve the main engine interface, IFlix
             */
            flix = ClassFactory.createFlix();
        } catch(RuntimeException e) {
            e.printStackTrace();
            System.exit(1);
        }
    }

    private static void printStackTrace(RuntimeException e) {
        e.printStackTrace();
        printEncoderStatus();
        System.exit(1);
    }

    public static void main(String[] args) {
        init();

        /*enable logging, 0=none(disable) 1=info 2=error(asserts) 3=debug 4=heavy

        CONOUT$ can be used as the log file name to send output to the console
        try {
            flix.setLogLevel(3);
            flix.setLogPath("C:\\cli_encode_java.log");
        } catch(RuntimeException e) {
            printStackTrace(e);
        }*/

        /*
         * print some library information
         */
        try {
            System.out.println("Flix Engine COM library. Flix Engine v" +
                               flix.version() + " COM v" + flix.com_version());
            System.out.println(flix.copyright() + "\n");
        } catch(RuntimeException e) {
    
```

```

        printStackTrace(e);
    }

    if (args.length != 2) {
        System.out.println("usage: java -classpath "+
            "flixengine_com.jar;com4j.jar;."+
            " cli_encode <infile> <outfile>\n");
        System.exit(1);
    }

    /*
     * set the source file
     */
    try {
        System.out.println("Input File : " + args[0]);
        flx.setInputFile(args[0]);
    } catch(RuntimeException e) {
        printStackTrace(e);
    }

    /*
     * retrieve the video options interface, IVideoOptions
     */
    try {
        IVideoOptions vidOpts = flx.videoOptions();

        /*
         * print input file information
         */
        int width      = vidOpts.getSourceWidth();
        int height     = vidOpts.getSourceHeight();
        int duration    = flx.getSourceDuration();
        System.out.println("
                                Width:      " + width + "\n" +
                                "                                Height:     " + height + "\n" +
                                "                                Duration:   " + duration + "ms");
    } catch(RuntimeException e) {
        printStackTrace(e);
    }

    /*
     * set the destination file
     */
    try {
        flx.setOutputFile(args[1]);
        System.out.println("Output File : " + args[1]);
    } catch(RuntimeException e) {
        printStackTrace(e);
    }

    /*
     * Options may be set and codecs/filters/muxers may be added prior to encode()
     */

    /*Add the scale filter
    try {
        IFlixPlgn filter= flx.addFilter(flix.fE2_FILTER_SCALE());
        filter.setParam(flix.fE2_SCALE_WIDTH(), 320);
        filter.setParam(flix.fE2_SCALE_HEIGHT(), 240);
    } catch(RuntimeException e) {
        printStackTrace(e);
    }*/

    /*Add the vp6 codec. Though it is the default, you must add it in order
    to modify its settings
    try {
        IFlixPlgn codec= flx.addCodec(flix.fE2_CODEEC_VP6());

```



```

        codec.setParam(flix.fE2_VP6_RC_MODE(),
            FE2_VideoBitrateControls.VBR_1PASSControl.ordinal());
    } catch (RuntimeException e) {
        printStackTrace(e);
    } */

    /*Use the FLV muxer (default)
    try {
        IFlixPlgn muxer= flix.addMuxer(flix.fE2_MUXER_FLV());
    } catch (RuntimeException e) {
        printStackTrace(e);
    } */

    /*
    * start the encode
    */
    try {
        flix.encode();

        /*
        * retrieve the encoding status interface, IEncodingStatus
        */
        IEncodingStatus status = flix.encodingStatus();

        System.out.println();
        boolean ier;
        do {
            try {Thread.currentThread().sleep(500);}
            catch (InterruptedException e) {}
            ier = flix.isEncoderRunning() == 1;
            System.out.print("\rEncoding... " + status.percentComplete() + "% ");

            } while(ier);
        System.out.println("Done!");
        printEncoderStatus();
    } catch (RuntimeException e) {
        System.out.println(e);
        printEncoderStatus();
    }

    /*
    * Force the cleanup of IFlix.
    * Though this is not strictly necessary in this sample, as
    * it is about to exit, if the script is more involved it may be
    * necessary so the input file can be moved as destruction of
    * the underlying FLIX2HANDLE occurs within IFlix's destructor.
    */
    flix.dispose(); flix = null;
    /*
    * Additionally, free any resources/threads started by com4j for use
    * with this object
    */
    com4j.COM4J.cleanUp();
}

private static void printEncoderStatus() {
    System.out.println("\nEncoder Status");
    FE2_EncState res = flix.getEncoderState();
    System.out.println(" flix.getEncoderState: " + res );

    System.out.println(" flix.errno_: flixerrno: " + flix.flixerrno() +
        " syserrno: " + flix.syserrno());
}
}

```

6.11 Perl

The Perl examples require the `Win32::OLE` module to be installed. A typical install of [ActivePerl](#) includes this module, while [Cygwin](#) does not. You can verify that you have this module installed by typing the following at the command prompt, assuming perl is in your PATH:

```
$ perl -MWin32::OLE -e'__END__'
```

Nothing will be output should the command succeed.
On failure a message containing the text:

```
Can't locate Win32/OLE.pm in @INC (@INC contains: ... ).
BEGIN failed--compilation aborted.
```

will be output. In this case you will need to install `Win32::OLE` before running the examples either by downloading the source from [CPAN](#) or using the CPAN module.

When you have `Win32::OLE` installed you can start by running the [command line](#) sample, then try the [CGI](#) sample, if you have a webserver installed.

6.12 Command Line

```
#!/usr/bin/perl -w
##=====
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##-----
##
## File:          $Workfile$
##               $Revision$
##
## Last Update:  $DateUTC$
##
##-----
##
use Win32::OLE;
## extract constants from the type lib
use Win32::OLE::Const 'On2.FlixEngine Type Library';
use Win32::OLE::Variant;

sub print_encoder_status()
{
    print "\nEncoder Status\n";

    $res= $flix->getEncoderState();
    print " flix->getEncoderState: $res\n";

    my $flixerr= Variant(VT_I4|VT_BYREF,0);
    my $syserr= Variant(VT_I4|VT_BYREF,0);
    $flix->errno_($flixerr,$syserr);
    printf(" flix->errno_: flixerrno:%d syserrno:%d\n",$flixerr,$syserr);
}

sub checkhr($)
{
    if(Win32::OLE->LastError) {
        warn "$_[0] failed. hr= ".Win32::OLE->LastError;
        print_encoder_status();
        exit 1;
    }
}
```

```

}

##
## retrieve the main engine interface, IFlix
##
$flix= Win32::OLE->new("On2.FlixEngine") or
    die("Error loading Flix Engine COM: $!, ".Win32::OLE->LastError);

##enable logging, 0=none(disable) 1=info 2=error(asserts) 3=debug 4=heavy
##CONOUT$ can be used as the log file name to send output to the console
#$flix->setLogLevel(3); checkhr('flix->setLogLevel');
#$flix->setLogPath("\\cli_encode.pl.log"); checkhr('flix->setLogPath');

##
## print some library information
##
print "Flix Engine COM library. Flix Engine v".$flix->version().
    " COM v".$flix->com_version()."\n";
print $flix->copyright()."\n\n";

if(@ARGV<2) {
    die("usage: cli_encode.pl <infile> <outfile>\n");
}

##
## set the source file
##
print "Input File : $ARGV[0]\n";
$flix->setInputFile($ARGV[0]); checkhr('flix->setInputFile($ARGV[0])');

##
## retrieve the video options interface, IVideoOptions
##
$vidopts= $flix->videoOptions(); checkhr('flix->videoOptions()');

##
## print input file information
##
$srcduration= $flix->getSourceDuration(); checkhr('flix->getSourceDuration()');
$srcw= $vidopts->getSourceWidth(); checkhr('vidopts->getSourceWidth()');
$srch= $vidopts->getSourceHeight(); checkhr('vidopts->getSourceHeight()');
print <<EOT;
        Width:      $srcw
        Height:     $srch
        Duration:   ${srcduration}ms
EOT

##
## set the destination file
##
print "Output File : $ARGV[1]\n";
$flix->setOutputFile($ARGV[1]); checkhr('flix->setOutputFile($ARGV[1])');

##
## Options may be set and codecs/filters/muxers may be added prior to encode()
##

##Add the scale filter
$filter= $flix->addFilter($flix->FE2_FILTER_SCALE);
# checkhr('flix->addFilter(FE2_FILTER_SCALE)');
#
$filter->setParam($flix->FE2_SCALE_WIDTH, 240);
# checkhr('flix->setParam(FE2_SCALE_WIDTH,240)');
$filter->setParam($flix->FE2_SCALE_HEIGHT,160);
# checkhr('flix->setParam(FE2_SCALE_HEIGHT,160)');

##Add the vp6 codec. Though it is the default, you must add it in order

```

```
##to modify its settings
#$codec= $flix->addCodec($flix->FE2_CODEC_VP6);
# checkhr('flix->addCodec(FE2_CODEC_VP6)');
#
#$codec->setParam($flix->FE2_VP6_RC_MODE,VBR_1PASSControl);
# checkhr('codec->setParam(FE2_VP6_RC_MODE,VBR_1PASSControl)');

##Use the FLV muxer (default)
#$muxer= $flix->addMuxer($flix->FE2_MUXER_FLV);
# checkhr('flix->addMuxer(FE2_MUXER_FLV)');

##
## start the encode
##
$flix->encode(); checkhr('flix->encode()');

##
## retrieve the encoding status interface, IEncodingStatus
##
$encstatus= $flix->encodingStatus(); checkhr('flix->encodingStatus()');

print "\n";
$| = 1; #auto flush
my $ier;
do {
    sleep(1);
    $ier = $flix->isEncoderRunning();
    print "\rEncoding...".$encstatus->percentComplete()."% ";
} while($ier);

print "Done!\n";
print_encoder_status();

##cleanup
Win32::OLE->Uninitialize();
```

6.13 CGI

This example consists of 2 parts: [flix2_sample.cgi](#) and [process_sample.cgi](#). [flix2_sample.cgi](#) searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's COM interfaces.

To use this example navigate to [flix2_sample.cgi](#) in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to [process_sample.cgi](#) which runs the encode loop.

Default file locations used by the scripts:

Input : *C:/inetpub/flixmedia/in* (\$indir in [flix2_sample.cgi](#))
Overlay : *C:/inetpub/flixmedia/overlay* (\$overlaydir in [flix2_sample.cgi](#))
Output : *C:/inetpub/flixmedia/out* (\$outdir in [process_sample.cgi](#))

6.13.1 flix2_sample.cgi

```
#!/perl -w
##=====
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
```

```
##
##-----
##
## File:          $Workfile$
##              $Revision$
##
## Last Update: $DateUTC$
##
##-----
##

#flix2_sample.cgi
#expose API functions available in the COM library
#submits a form to process_sample.cgi which makes calls to flixengine_com.dll
#through the Win32::OLE module

use CGI qw(:standard -no_xhtml);

$| = 1; #auto flush

my $prefix      = "C:\\Inetpub\\";
my $indir       = $prefix."flixmedia\\in\\";      #src file directory
my $overlaydir  = $prefix."flixmedia\\overlay\\";  #overlay image file directory

print header;
print <<HTML_END;

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head>
  <title>Flix CGI Sample - Perl</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

  <script type="text/javascript">
function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
        i++;
    }
}

function toggle_expand(_this)
{
    _this.parentNode.className= (_this.parentNode.className=='expanded') ?
        'collapsed' : 'expanded';
    document.getElementById('showall').checked=false;
}

function set_table_visible(ctable,visible)
{
    var table= document.getElementById(ctable);

    if(table) {
        if(visible && table.className == 'hidden') {
            table.className= '';
        } else if(!visible && table.className == '') {
            table.className= 'hidden';
        }
    }
}

function hide_tablelist(vistablename,tablelist)
{
    var table= document.getElementById(vistablename);
```

```

if(table&&table.className=='hidden') {
    table.className= '';

    var i=0;
    while(tablelist[i]) {
        if(tablelist[i]!=vistablename) {
            var hiddentable= document.getElementById(tablelist[i]);

            set_table_visible(tablelist[i],false);
            /*clear down any values set in the hidden table to avoid posting
            unnecessary values*/
            clear_value(hiddentable.getElementsByTagName('input'));
            clear_value(hiddentable.getElementsByTagName('select'));
        }
        i++;
    }
}

function set_acodec_visible(ctable)
{
    var acodecs= new Array('aactable','aacplustable','amrnhtable','lametable','pcm
        table','vorbistable');
    hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
    var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8tabl
        e');
    hide_tablelist(ctable,vcodecs);
}

function set_muxer_visible(mtable)
{
    var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','
        tg2table','tgptable','webmtable');
    hide_tablelist(mtable,muxers);
}

function clear_value(list)
{
    var i=0;
    while(list[i]) {
        if(list[i].type=='checkbox') { list[i++].checked=false; }
        else { list[i++].value=''; }
    }
}

function toggle_ftable(ftable,enabled)
{
    var table= document.getElementById(ftable);

    if(table) {
        table.className = enabled ? '' : 'disabled';
        if(!enabled) {
            clear_value(table.getElementsByTagName('input'));
            clear_value(table.getElementsByTagName('select'));
        }
    }
}

function reset_tables()
{
    var table_list= document.getElementsByTagName('table');
    var i=0;

```

```

while(table_list[i]) {
  if (table_list[i].id.length > 7 &&
      table_list[i].id.substring(0,7) == 'filter_') {
    table_list[i].className= 'disabled';
  } else if (table_list[i].id.indexOf('table') != -1) {
    table_list[i].className= 'hidden';
  }
  i++;
}
}

</script>

<style type="text/css">
<!--
html {
  font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
  font-size: 100%;
  color: rgb(56,56,56);
  background-color: rgb(236,236,236);
  border-style: solid;
  border-color: rgb(236,236,236);
}

body {
  text-align: center;
  margin: 0 auto;
}

div.content {
  color: rgb(56,56,56);
  background-color: rgb(246,246,246);
  text-align: left;
  margin: 0 auto;
  width: 80%;
  min-width: 768px;
  max-width: 932px;
  border-width: 0 1px;
  border-color: rgb(144,144,144);
  border-style: solid;
}

div.content:after {
  content: "";
  color: inherit;
  background-color: rgb(250,250,250);
  border-top: 1px solid rgb(144,144,144);
  height: 20px;
  width: 100%;
  display: block;
}

fieldset table, fieldset {display: none;}
fieldset.expanded, fieldset.collapsed {display: block;}

/*first is fallback for IE*/
fieldset.expanded table {display: block;}
fieldset.expanded table {display: table;}

h1 {
  font-family: sans-serif;
  font-size: 150%;
  font-weight: normal;
  text-align: left;
  letter-spacing: -1px;
  color: rgb(74,74,74);
  background-color: inherit;

```

```

        margin: 0;
    }

    a {
        color: rgb(74,74,74);
        background-color: transparent;
    }

    label {
        font-size: 75%;
    }

    fieldset {
        font-size: 75%;
        line-height: 130%;
        padding: 0;
        margin: 20px;
        border: none;
    }

    fieldset.expanded {
        color: inherit;
        background-color: rgb(252,252,252);
        border-style: solid;
        border-width: 1px;
        border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
    }

    legend {
        padding: 0 5px;
        border-left: 10px solid rgb(217,217,217);
        cursor: pointer;
    }

    legend:hover {text-decoration: underline;}
    fieldset.expanded legend {
        font-size: 150%;
        font-weight: bold;
        letter-spacing: -1px;
        background: transparent;
        margin-left: 12px;
        border-right: 10px solid rgb(217,217,217);
        display: block;
    }

    table {
        font-size: 100%;
        border-spacing: 0;
        /*border-collapse: collapse;*/
        width: 100%;
    }

    th:before { display: none; }
    th, td {
        width: 50%;
        vertical-align: top;
        padding: 2px 3px;
        border-width: 1px 0;
        border-style: solid;
        border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
    }
    tr:first-child>* {border-top-color: rgb(252,252,252);}
    th {
        font-weight: normal;
        text-align: left;
        padding: 2px 2px 2px 5px;
    }

    input[type] {
        font-family: monospace;

```



```

        font-size: 100%;
        color: rgb(56,56,56);
        background-color: inherit;
    }
    [type="text"], [type="number"] {
        margin-right: 13px;
        width: 222px;
    }

    [type="checkbox"].filter {
        margin-left: 4px
    }
    [type="checkbox"] {
        margin-left: 13px
    }
    [type="button"], [type="submit"] {
        font-size: 1em;
        margin: 0 2px 0 13px;
    }

    fieldset {
        font-size: 75%;
        margin: 20px 10px;
    }
    fieldset input[type="text"], fieldset input[type="number"] {
        width: 95% !important;
        margin: 0;
        display: block;
    }
    fieldset.expanded table.hidden {
        display: none;
    }

    table.disabled {
        color: rgb(176,176,176);
    }
    -->
</style>
</head>

<body>
<div class='content'>
<noscript>
<p>This page requires javascript be enabled.</p>
</noscript>

<hr>
<h1>Flix CGI Sample</h1>

<p><small>flix2_sample.cgi version 1.9</small></p>
<h4>Instructions</h4>
<ul>
<li>In this sample you must choose a source file and an output file.<br>
    If you leave all the other options blank then the sample will not call
    the corresponding Flix Engine function and the default will be used.<br>
    When done please press the "Start Encode" button at the bottom of the page.<br>
<li>Mouse over a function name to see its default, if applicable.
<li>Current source file directory: $indir
</li>
</ul>

<p>
<label><input type="checkbox" id='showall' onclick='showall(this.checked)''>Show
    all</label>
</p>

```

```

<form action="process_sample.cgi" method="post">

<!-- ##SOURCE FILE##### -->

<hr>
<fieldset class='expanded' id="srcfile">
<legend onclick='toggle_expand(this)''>Source File</legend>
<table>

<tr>
<td>
HTML_END

if(opendir DIR,$indir) {
    #search the input directory for potential files to encode
    #populating a list box in the process
    @a = sort grep { -f $indir.$_ } readdir DIR;
    foreach (@a) {
        $a{$indir.$_} = $_;
        $_ = $indir.$_;
    }
    print scrolling_list(-name=>'setInputFile',
                        -values=>\@a,
                        -labels=>\%a,
                        -default=>$a[0],
                        -size=>scalar(@a)<15?scalar(@a):15);

    closedir DIR;
} else {
    print "WARNING couldn't open $indir: $!",br;
}
print <<HTML_END;

</td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE##### -->
-->
<hr>
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)''>Output File</legend>
<table>

<tr>
<td>
<input type="text" name="setOutputFile" value="cgi-pl-out.flv">
</td>
</tr>

</table>
</fieldset>

<!-- ##MAIN OPTIONS#####
### -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)''>Main Options</legend>
<table>

<tr>
<th><abbr title="Default: FALSE">setOverwriteExistingFiles</abbr></th>
<td>
<select name="setOverwriteExistingFiles">
<option value=""></option>
<option value="on2true">TRUE</option>

```

```

    <option value="on2false">FALSE</option>
  </select>
</td>
</tr>

<tr>
  <th><abbr title="Default: TRUE">setExportAudio</abbr></th>
  <td>
    <select name="setExportAudio">
      <option value=""></option>
      <option value="on2true">TRUE</option>
      <option value="on2false">FALSE</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="Default: TRUE">setExportVideo</abbr></th>
  <td>
    <select name="setExportVideo">
      <option value=""></option>
      <option value="on2true">TRUE</option>
      <option value="on2false">FALSE</option>
    </select>
  </td>
</tr>

</table>
</fieldset>

<!-- ##CODECS#####
    ## -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this)''>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
  <td>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
        onfocus="set_vcodec_visible('vp8table')">
      <abbr title="For use with WebM">FE2_CODEC_VP8</abbr>&nbsp;
    </label>
    <br>

    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
        onfocus="set_vcodec_visible('vp6table')">
      <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr>&nbsp;
    </label>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
        onfocus="set_vcodec_visible('vp6atable')">
      <abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr>&nbsp;
    </label>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_H263'
        onfocus="set_vcodec_visible('h263table')">
      <abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr>&nbsp;
    </label>
    <br>

    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
        onfocus="set_vcodec_visible('h263table')">

```

```

    <abbr title="For use with 3GP">FE2_CODEEC_H263_BASELINE</abbr>&nbsp;
</label>
<label>
  <input type='radio' name='vcodec:' value='FE2_CODEEC_H264'
    onfocus="set_vcodec_visible('h264table') ">
  <abbr title="For use with 3GP/3G2/MOV/MP4">FE2_CODEEC_H264</abbr>&nbsp;
</label>

<!-- VP6 codec parameters -->
<table id='vp6table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_VP6_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_KFINTTYPE">
        <option value=""></option>
        <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
        <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_VP6_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP6_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_RC_MODE">
        <option value=""></option>
        <option value="VBR_2PASSControl">VBR_2PASSControl</option>
        <option value="CBR_2PASSControl">CBR_2PASSControl</option>
        <option value="VBR_1PASSControl">VBR_1PASSControl</option>
        <option value="CBR_1PASSControl">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_CXMODE">
        <option value=""></option>
        <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
        <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: VP6_E">FE2_VP6_PROFILE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6_PROFILE">
        <option value=""></option>
        <option value="VP6_E">VP6_E</option>
        <option value="VP6_S">VP6_S</option>
      </select>
    </td>
  </tr>

```

```

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_CONCURRENCY'></td>
</tr>

<tr>
  <th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_UNDERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_TEMPORAL_RESAMPLING</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">
    FE2_VP6_TEMPORAL_DOWN_WATERMARK</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
  <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td>
</tr>

```

```

<tr>
  <th><abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_2PASS_MIN_SECTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP6_2PASS_MAX_SECTION'></td>
</tr>
</table>
<!-- END - VP6 codec parameters -->

<!-- VP6A codec parameters -->
<table id='vp6atable' class='hidden'>
  <tr>
    <th><abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 68kbps (15% of default 448kbps)">
      FE2_VP6A_ALPHA_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6A_KFINTTYPE">
        <option value=""></option>
        <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
        <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_VP6A_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6A_RC_MODE">
        <option value=""></option>
        <option value="VBR_2PASSControl">VBR_2PASSControl</option>
        <option value="CBR_2PASSControl">CBR_2PASSControl</option>
        <option value="VBR_1PASSControl">VBR_1PASSControl</option>
        <option value="CBR_1PASSControl">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP6A_CXMODE">
        <option value=""></option>

```

```

        <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
        <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
    </select>
</td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
    <th><abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_MIN_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_ALPHA_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MIN_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_SHARPNESS'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_NOISE_REDUCTION'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6A_TEMPORAL_RESAMPLING</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
    </tr>

```

```

</tr>

<tr>
<th><abbr title="Default: bits per pixel dependent (see API docs)">
  FE2_VP6A_TEMPORAL_DOWN_WATERMARK</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'>
  </td>
</tr>

<tr>
<th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PREBUFFER'></td>
</tr>

<tr>
<th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr>
  </th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
<th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
<th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MIN_SECTION'></td>
</tr>

<tr>
<th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MAX_SECTION'></td>
</tr>
</table>
<!-- END - VP6A codec parameters -->

<!-- H263 codec parameters -->
<table id='h263table' class='hidden'>
<tr>
<th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>
<td>
  <select name="codec:setParam:FE2_H263_KFINTTYPE">
    <option value=""></option>
    <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
    <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
  FE2_H263_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_KFFREQ'></td>

```



```

</tr>

<tr>
<th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>
<td>
<select name="codec:setParam:FE2_H263_RC_MODE">
<option value=""></option>
<option value="VBR_2PASSControl">VBR_2PASSControl</option>
<option value="CBR_2PASSControl">CBR_2PASSControl</option>
<option value="VBR_1PASSControl">VBR_1PASSControl</option>
<option value="CBR_1PASSControl">CBR_1PASSControl</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MAX_Q'></td>
</tr>

<tr>
<th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MIN_Q'></td>
</tr>

</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
<tr>
<th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
FE2_H264_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_KFFREQ'></td>
</tr>

<tr>
<th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
<td>
<select name="codec:setParam:FE2_H264_RC_MODE">
<option value=""></option>
<option value="VBR_1PASSControl">VBR_1PASSControl</option>
<option value="CBR_1PASSControl">CBR_1PASSControl</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
<td>
<select name="codec:setParam:FE2_H264_PROFILE">
<option value=""></option>
<option value="BASE_H264PROFILE">BASE_H264PROFILE</option>
<option value="MAIN_H264PROFILE">MAIN_H264PROFILE</option>
<option value="HIGH_H264PROFILE">HIGH_H264PROFILE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_B_FRAME_RATE'></td>

```

```

</tr>

<tr>
<th><abbr title="Default: Dependent on profile selection, see API docs. Valid
Range [0,5]">FE2_H264_SPEED</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H264_SPEED'></td>
</tr>

</table>
<!-- END - H264 codec parameters -->

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
<tr>
<th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP8_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
<td>
<select name="codec:setParam:FE2_VP8_KFINTTYPE">
<option value=""></option>
<option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
<option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
FE2_VP8_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP8_KFFREQ'></td>
</tr>

<tr>
<th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
<td>
<select name="codec:setParam:FE2_VP8_RC_MODE">
<option value=""></option>
<option value="VBR_2PASSControl">VBR_2PASSControl</option>
<option value="CBR_2PASSControl">CBR_2PASSControl</option>
<option value="VBR_1PASSControl">VBR_1PASSControl</option>
<option value="CBR_1PASSControl">CBR_1PASSControl</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
<td>
<select name="codec:setParam:FE2_VP8_CXMODE">
<option value=""></option>
<option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
<option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP8_THREADS'></td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP8_PROFILE'></td>

```

```

</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_LAG'></td>
</tr>

<tr>
  <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_UNDERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_OVERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
  <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>

```

```

    <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

<tr>
    <th><abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_ALTREF'></td>
</tr>

<tr>
    <th><abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_AR_MAX_FRAMES'></td>
</tr>

<tr>
    <th><abbr title="">FE2_VP8_AR_TYPE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_AR_TYPE'></td>
</tr>

<tr>
    <th><abbr title="">FE2_VP8_AR_STRENGTH</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_AR_STRENGTH'></td>
</tr>

<tr>
    <th><abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_MB_STATIC_THRESHOLD'></td>
</tr>

<tr>
    <th><abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_TOKEN_PARTITIONS'></td>
</tr>
</table>
<!-- END - VP8 codec parameters -->

</td>
</tr> <!-- END - video codecs -->

<tr><th><b>Audio Codecs</b></th></tr>
<tr>
<td>
<label>
    <input type='radio' name='acodec:' value='FE2_CODEC_VORBIS'
        onfocus="set_acodec_visible('vorbistable') ">
    <abbr title="For use with WebM">FE2_CODEC_VORBIS</abbr>&nbsp;
</label>
<br>

<label>
    <input type='radio' name='acodec:' value='FE2_CODEC_AAC'
        onfocus="set_acodec_visible('aactable') ">
    <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AAC</abbr>&nbsp;
</label>
<label>
    <input type='radio' name='acodec:' value='FE2_CODEC_AACPLUS'
        onfocus="set_acodec_visible('aacplustable') ">
    <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AACPLUS</abbr>&nbsp;
</label>
<label>
    <input type='radio' name='acodec:' value='FE2_CODEC_LAME'
        onfocus="set_acodec_visible('lametable') ">
    <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_LAME</abbr>&nbsp;
</label>

```

```

<label>
  <input type='radio' name='acodec:' value='FE2_CODEC_PCM'
        onfocus="set_acodec_visible('pcmtable') ">
  <abbr title="For use with FLV/SWF">FE2_CODEC_PCM</abbr>&nbsp;
</label>
<br>

<label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AMR_NB'
        onfocus="set_acodec_visible('amrnhtable') ">
  <abbr title="For use with 3GP">FE2_CODEC_AMR_NB</abbr>&nbsp;
</label>

<!-- AMR_NB codec parameters -->
<table id='amrnhtable' class='hidden'>
  <tr>
    <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_AMR_BITRATE'></td>
  </tr>

</table>
<!-- END - AMR_NB codec parameters -->

<!-- AAC codec parameters -->
<table id='aactable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_AAC_BITRATE'></td>
  </tr>

</table>
<!-- END - AAC codec parameters -->

<!-- AACPLUS codec parameters -->
<table id='aacplustable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_AACPLUS_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: disabled (aacPlus v1)">
      FE2_AACPLUS_PARAMETRIC_STEREO</abbr></th>
    <td>
      <select name="codec:setParam:FE2_AACPLUS_PARAMETRIC_STEREO">
        <option value=""></option>
        <option value="0">disable (aacPlus v1)</option>
        <option value="1">enable (aacPlus v2)</option>
      </select>
    </td>
  </tr>

</table>
<!-- END - AACPLUS codec parameters -->

<!-- LAME codec parameters -->
<table id='lametable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_LAME_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_LAME_QUALITY'></td>
  </tr>

  <tr>

```

```

<th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
<td>
  <select name="codec:setParam:FE2_LAME_RC_MODE">
    <option value=""></option>
    <option value="LAME_CBR">LAME_CBR</option>
    <option value="LAME_ABR">LAME_ABR</option>
    <option value="LAME_VBR_rh">LAME_VBR_rh</option>
    <option value="LAME_VBR_mtrh">LAME_VBR_mtrh</option>
  </select>
</td>
</tr>
</table>
<!-- END - LAME codec parameters -->

<!-- PCM codec parameters -->
<table id='pcmtable' class='hidden'>
  <tr>
    <th>(FE2_CODEC_PCM defines no parameters)</th>
  </tr>
</table>
<!-- END - PCM codec parameters -->

<!-- VORBIS codec parameters -->
<table id='vorbistable' class='hidden'>
  <tr>
    <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VORBIS_BITRATE'></td>
  </tr>
</table>
<!-- END - VORBIS codec parameters -->

</td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS#####
    ### -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this) '>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_CUT' value='avfilter:'
        onchange="toggle_ftable('filter_cut',this.checked) ">
      FE2_FILTER_CUT
    </label>

    <table id='filter_cut' class='disabled'>
      <tr>
        <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CUT_START_SEC'></td>
      </tr>

      <tr>
        <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CUT_STOP_SEC'></td>
      </tr>
    </table>
  </td>
</tr>

```

```

        <tr>
        <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>
        <td>
        <select name="filter:setParam:FE2_CUT_USE_SEEK">
        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
        </select>
        </td>
        </tr>

    </table>

</td>
</tr>
<!-- END - CUT filter parameters -->

<tr><th><b>Video Filters</b></th></tr>

<tr>
<td>
    <label>
    <input type='checkbox' class='filter'
        name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:'
        onchange="toggle_ftable('filter_adaptive_deinterlace',this.checked)">
    FE2_FILTER_ADAPTIVE_DEINTERLACE
    </label>

    <table id='filter_adaptive_deinterlace' class='disabled'>
    <tr>
    <th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</ab
        br></th>
    <td>
    <select name="filter:setParam:FE2_ADAPTIVE_DEINTERLACE_MODE">
    <option value=""></option>
    <option value="DEINTERLACE_NONE">DEINTERLACE_NONE</option>
    <option value="DEINTERLACE_1_2_1_BLUR">DEINTERLACE_1_2_1_BLUR</option>
    <option value="DEINTERLACE_DROP_FIELD">DEINTERLACE_DROP_FIELD</option>
    <option value="DEINTERLACE_ADAPTIVE">DEINTERLACE_ADAPTIVE</option>
    </select>
    </td>
    </tr>
    </table>

    </td>
</tr>
<!-- END - ADAPTIVE DEINTERLACE filter parameters -->

<tr>
<td>
    <label>
    <input type='checkbox' class='filter'
        name='FE2_FILTER_BCHS' value='vfilter:'
        onchange="toggle_ftable('filter_bchs',this.checked)">
    FE2_FILTER_BCHS
    </label>

    <table id='filter_bchs' class='disabled'>
    <tr>
    <th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_BCHS_BRIGHTNESS'></td>
    </tr>
    <tr>
    <th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_BCHS_CONTRAST'></td>
    </tr>
    </table>

```

```

        <th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_BCHS_HUE'></td>
    </tr>
    <tr>
        <th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_BCHS_SATURATION'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - BCHS filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_BLUR' value='vfilter:'
                onchange="toggle_ftable('filter_blur',this.checked)">
            FE2_FILTER_BLUR
        </label>

        <table id='filter_blur' class='disabled'>
            <tr>
                <th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
                <td>
                    <select name="filter:setParam:FE2_BLUR_FILTER">
                        <option value=""></option>
                        <option value="BLUR_LOWPASS">BLUR_LOWPASS</option>
                        <option value="BLUR_GAUSS">BLUR_GAUSS</option>
                    </select>
                </td>
            </tr>

            <tr>
                <th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
                <td>
                    <select name="filter:setParam:FE2_BLUR_MASKSIZE">
                        <option value=""></option>
                        <option value="MASK_3x3">MASK_3x3</option>
                        <option value="MASK_5x5">MASK_5x5</option>
                    </select>
                </td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_CROP' value='vfilter:'
                onchange="toggle_ftable('filter_crop',this.checked)">
            FE2_FILTER_CROP
        </label>

        <table id='filter_crop' class='disabled'>
            <tr>
                <th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_CROP_TOP'></td>
            </tr>

            <tr>
                <th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>

```



```

        <td><input type='text' name='filter:setParam:FE2_CROP_BOTTOM'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CROP_LEFT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_CROP_RIGHT'></td>
    </tr>
</table>

</td>
</tr>
<!-- END - CROP filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_DENOISE' value='vfilter:'
            onchange="toggle_ftable('filter_denoise',this.checked)">
        FE2_FILTER_DENOISE
    </label>

    <table id='filter_denoise' class='disabled'>
        <tr>
            <th><abbr title="Default: 0. Range: [0.0,1.0]">FE2_DENOISE_NOISE_LEVEL</abbr>
            </th>
            <td><input type='text' name='filter:setParam:FE2_DENOISE_NOISE_LEVEL'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_FRAMERATE' value='vfilter:'
            onchange="toggle_ftable('filter_framerate',this.checked)">
        FE2_FILTER_FRAMERATE
    </label>

    <table id='filter_framerate' class='disabled'>
        <tr>
            <th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAM
                ERATE_DECIMATE</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_FRAMERATE_DECIMATE'></td>
        </tr>

        <tr>
            <th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FR
                AMERATE_FPS</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_FRAMERATE_FPS'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>

```

```

<td>
<label>
  <input type='checkbox' class='filter'
    name='FE2_FILTER_MIRROR' value='vfilter:'
    onchange="toggle_ftable('filter_mirror',this.checked)">
  FE2_FILTER_MIRROR
</label>

<table id='filter_mirror' class='disabled'>
<tr>
<th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
<td>
  <select name="filter:setParam:FE2_MIRROR_HORIZONTAL">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
<td>
  <select name="filter:setParam:FE2_MIRROR_VERTICAL">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
  </select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
<td>
<label>
  <input type='checkbox' class='filter'
    name='FE2_FILTER_OVERLAY' value='vfilter:'
    onchange="toggle_ftable('filter_overlay',this.checked)">
  FE2_FILTER_OVERLAY
</label>

<table id='filter_overlay' class='disabled'>
<tr>
<th>
  <abbr title="Currently searching $overlaydir for overlay images">FE2_OVERLAY
    _FILE</abbr>
</th>
<td>
HTML_END

if(opendir DIR,$overlaydir) {
  #populate the list box with files from the overlay directory
  @a = sort grep { -f $overlaydir.$_ } readdir DIR;
  foreach (@a) {
    $a{$overlaydir.$_} = $_;
    $_ = $overlaydir.$_;
  }
  unshift(@a,"");
  print scrolling_list(-name=>'filter:setParamAsStr:FE2_OVERLAY_FILE',
    -values=>\@a,
    -labels=>\%a,
    -size=>scalar(@a)),br;
  print "default: None, must be set to the absolute path of the overlay".

```

```

        " input file, e.g. '/path/to/my/overlay.png'";
        closedir DIR;
    } else {
        print "WARNING couldn't open $overlaydir: $!",br;
    }

print <<HTML_END;

    </td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
<td>
    <select name="filter:setParam:FE2_OVERLAY_MASK_XY">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
<td><input type='text' name=' filter:setParam:FE2_OVERLAY_MASK_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
<td><input type='text' name=' filter:setParam:FE2_OVERLAY_MASK_Y'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
<td>
    <select name="filter:setParam:FE2_OVERLAY_MASK_RGB">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
<td><input type='text' name=' filter:setParam:FE2_OVERLAY_MASK_R'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
<td><input type='text' name=' filter:setParam:FE2_OVERLAY_MASK_G'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
<td><input type='text' name=' filter:setParam:FE2_OVERLAY_MASK_B'></td>
</tr>

<tr>
<th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
<td>
    <select name="filter:setParam:FE2_OVERLAY_POS">
    <option value=""></option>
    <option value="FE2_OVERLAY_POS_MODE_TOPLEFT">FE2_OVERLAY_POS_MODE_TOPLEFT</option>
    <option value="FE2_OVERLAY_POS_MODE_BOTLEFT">FE2_OVERLAY_POS_MODE_BOTLEFT</option>
    <option value="FE2_OVERLAY_POS_MODE_CENTER">FE2_OVERLAY_POS_MODE_CENTER</option>
    <option value="FE2_OVERLAY_POS_MODE_TOPRIGHT">

```

```

        FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
        <option value="FE2_OVERLAY_POS_MODE_BOTRIGHT">
        FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
        <option value="FE2_OVERLAY_POS_MODE_XY">FE2_OVERLAY_POS_MODE_XY</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_Y'></td>
</tr>

</table>

</td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
        name='FE2_FILTER_PNGEX' value='vfilter:'
        onchange="toggle_ftable('filter_pngex',this.checked)">
        FE2_FILTER_PNGEX
</label>

<table id='filter_pngex' class='disabled'>
<tr>
<th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'></td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_PREFIX'></td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'></td>
</tr>

<tr>
<th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_WIDTH'></td>
</tr>

<tr>
<th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_HEIGHT'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>
<td>
<select name="filter:setParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">

```

```

        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
<td>
    <select name="filter:setParam:FE2_PNGEX_ENABLE_ALPHA">
        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="interval in ms; Default: disabled">
    FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_EXPORT_INTERVAL'></td>

</tr>

<tr>
<th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">
    FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_EXPORT_TIME_STRIN
    G'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
<td>
    <select name="filter:setParam:FE2_PNGEX_EXPORT_CUE_POINTS">
        <option value=""></option>
        <option value="FE2_PNGEX_CP_ALL">All cue points (FE2_PNGEX_CP_ALL)</option>
        <option value="FE2_PNGEX_CP_NAV">Only navigation cue points (
            FE2_PNGEX_CP_NAV)</option>
        <option value="FE2_PNGEX_CP_EVENT">Only event cue points (
            FE2_PNGEX_CP_EVENT)</option>
    </select>
</td>
</tr>

<tr>
<th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">
    FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_COMPRESSION_LEVEL'></t
    d>
</tr>

<tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>

<tr>
<th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></t
    d>
</tr>

<tr>
<th><abbr title="start time in ms; Default: 0">
    FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_START_TIME
    '></td>
</tr>

```



```

<td>
<label>
  <input type='checkbox' class='filter'
    name='FE2_FILTER_SHARPEN' value='vfilter:'
    onchange="toggle_ftable('filter_sharpen',this.checked)">
  FE2_FILTER_SHARPEN
</label>

<table id='filter_sharpen' class='disabled'>
  <tr>
    <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
  </tr>
</table>

</td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
<td>
<label>
  <input type='checkbox' class='filter'
    name='FE2_FILTER_HIGHPASS' value='afilter:'
    onchange="toggle_ftable('filter_highpass',this.checked)">
  FE2_FILTER_HIGHPASS
</label>

<table id='filter_highpass' class='disabled'>
  <tr>
    <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_HIGHPASS_Q'></td>
  </tr>

  <tr>
    <th>FE2_HIGHPASS_CUTOFF</th>
    <td><input type='text' name='filter:setParam:FE2_HIGHPASS_CUTOFF'></td>
  </tr>
</table>

</td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
<td>
<label>
  <input type='checkbox' class='filter'
    name='FE2_FILTER_LOWPASS' value='afilter:'
    onchange="toggle_ftable('filter_lowpass',this.checked)">
  FE2_FILTER_LOWPASS
</label>

<table id='filter_lowpass' class='disabled'>
  <tr>
    <th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_LOWPASS_Q'></td>
  </tr>

  <tr>
    <th>FE2_LOWPASS_CUTOFF</th>
    <td><input type='text' name='filter:setParam:FE2_LOWPASS_CUTOFF'></td>
  </tr>
</table>

</td>

```

```

</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_RESAMPLE' value='afilter:'
        onchange="toggle_ftable('filter_resample',this.checked)">
        FE2_FILTER_RESAMPLE
    </label>

    <table id='filter_resample' class='disabled'>
      <tr>
        <th><abbr title="Default: 0">FE2_RESAMPLE_RATE</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_RESAMPLE_RATE'></td>
      </tr>

      <tr>
        <th><abbr title="Default: 0">FE2_RESAMPLE_CHANNELS</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_RESAMPLE_CHANNELS'></td>
      </tr>
    </table>

  </td>
</tr>
<!-- END - RESAMPLE filter parameters -->

</table>
</fieldset>

<!-- ##MUXERS#####
    ### -->
<hr>
<fieldset class='collapsed' id="muxers">
<legend onclick='toggle_expand(this)'>Muxers</legend>
<table>

<tr>
  <td>
    <label>
      <input type='radio' name='muxer:' value='FE2_MUXER_3GP'
        onfocus="set_muxer_visible('tgptable')">
        FE2_MUXER_3GP&nbsp;
    </label>
    <label>
      <input type='radio' name='muxer:' value='FE2_MUXER_3G2'
        onfocus="set_muxer_visible('tg2table')">
        FE2_MUXER_3G2&nbsp;
    </label>
    <label>
      <input type='radio' name='muxer:' value='FE2_MUXER_MOV'
        onfocus="set_muxer_visible('movtable')">
        FE2_MUXER_MOV&nbsp;
    </label>
    <label>
      <input type='radio' name='muxer:' value='FE2_MUXER_MP4'
        onfocus="set_muxer_visible('mp4table')">
        FE2_MUXER_MP4&nbsp;
    </label>
    <br>

    <label>
      <input type='radio' name='muxer:' value='FE2_MUXER_FLV'
        onfocus="set_muxer_visible('flvtable',true)">
        FE2_MUXER_FLV&nbsp;
    </label>

```



```

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_SWF'
        onfocus="set_muxer_visible('swftable',true)">
    FE2_MUXER_SWF
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
        onfocus="set_muxer_visible('fxmtable',true)">
    FE2_MUXER_FXM&nbsp;
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
        onfocus="set_muxer_visible('webmtable',true)">
    FE2_MUXER_WEBM&nbsp;
</label>
</td>
</tr>

<tr>
  <td>
    <!-- 3GP muxer parameters -->
    <table id='tgptable' class='hidden'>
      <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
          <select name="muxer:setParam:FE2_3GP_FASTSTART">
            <option value=""></option>
            <option value="on2false">FALSE</option>
            <option value="on2true">TRUE</option>
          </select>
        </td>
      </tr>
    </table>
    <!-- END - 3GP muxer parameters -->

    <!-- 3G2 muxer parameters -->
    <table id='tg2table' class='hidden'>
      <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
          <select name="muxer:setParam:FE2_3G2_FASTSTART">
            <option value=""></option>
            <option value="on2false">FALSE</option>
            <option value="on2true">TRUE</option>
          </select>
        </td>
      </tr>
    </table>
    <!-- END - 3G2 muxer parameters -->

    <!-- FLV muxer parameters -->
    <table id='flvtable' class='hidden'>
      <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'"
          >
          FE2_FLV_CUEPT_EVENT</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
      </tr>

      <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'"
          >
          FE2_FLV_CUEPT_NAV</abbr></th>

```

```

<td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_NAV'></td>
</tr>

<tr>
<th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'">
FE2_FLV_CUEPT_PARAM</abbr></th>
<td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
FE2_FLV_METADATA_ENABLE</abbr></th>
<td>
<table id='flv_metadata_enable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr
title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
DURATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' value='MD_DUR
ATION'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr
title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' value='MD_DAT
ASIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><ab
br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' value='MD
_AUDIO_SIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><ab
br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' value='MD
_VIDEO_SIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'>
<abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' v
alue='MD_AUDIO_DATARATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'>
<abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' v
alue='MD_VIDEO_DATARATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'>
<abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_

```

```

AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' value='MD_AUDIO_CODECID'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'>
<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' value='MD_VIDEO_CODECID'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr title="Default: Enabled">MD_WIDTH</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' value='MD_WIDTH'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr title="Default: Enabled">MD_HEIGHT</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' value='MD_HEIGHT'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abbr title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' value='MD_FRAMERATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' value='MD_CANSEEKTOEND'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' value='MD_LASTTIMESTAMP'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAME_TIMESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAME_TIMESTAMP</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAME_TIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAME_TIMESTAMP' value='MD_LASTKEYFRAME_TIMESTAMP'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAME_LOCATION'><abbr title="Default: Disabled">MD_LASTKEYFRAME_LOCATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAME_LOCATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAME_LOCATION' value='MD_LASTKEYFRAME_LOCATION'></td>
</tr>

```

```

ELOCATION' value='MD_LASTKEYFRAMELOCATION'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abbr
r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' value='MD_K
EYFRAMES'></td>
</tr>
</table>
</td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
FE2_FLV_METADATA_DISABLE</abbr></th>
<td>
<table id='flv_metadata_disable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abbr
r title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_DURATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' value='MD_D
URATION'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abbr
r title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' value='MD_D
ATASIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><a
bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' value='
MD_AUDIO_SIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><a
bbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' value='
MD_VIDEO_SIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE
'><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'
value='MD_AUDIO_DATARATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE
'><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'
value='MD_VIDEO_DATARATE'></td>
</tr>

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' ><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' v alue='MD_AUDIO_CODECID'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' ><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' v alue='MD_VIDEO_CODECID'></td> </tr>  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr t itle="Default: Enabled">MD_WIDTH</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _WIDTH' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' value='MD_WIDTH'> </td> </tr>  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enabled">MD_HEIGHT</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' value='MD_HEIGH T'></td> </tr>  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><ab br title="Default: Enabled">MD_FRAMERATE</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' value='MD _FRAMERATE'></td> </tr>  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'> <abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' val ue='MD_CANSEEKTOEND'></td> </tr>  |  |  |  |  | | --- | --- | --- | --- | | <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' ><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' v alue='MD_LASTTIMESTAMP'></td> </tr>  |  |  | | --- | --- | | <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETI MESTAMP'><abbr title="Default: Enabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></ th>  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD _LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYF RAMETIMESTAMP' value='MD_LASTKEYFRAMETIMESTAMP'></td> </tr> | | | | | | | | | | | | | | | |

```

```

        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Enabled">MD_LASTKEYFRAMELOCATION</abbr></label></th
>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='MD_LASTKEYFRAMELOCATION'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><ab
br title="Default: Enabled">MD_KEYFRAMES</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES' value='MD
_KEYFRAMES'></td>
    </tr>
</table>
</td>
</tr>
</table>
<!-- END - FLV muxer parameters -->

<!-- FXM muxer parameters -->
<table id='fxmtable' class='hidden'>
    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0' "
>
            FE2_FXM_CUEPT_EVENT</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_EVENT'></td>
    </tr>

    <tr>
        <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.
0' "
            FE2_FXM_CUEPT_NAV</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_NAV'></td>
    </tr>

    <tr>
        <th><abbr title="e.g. 'cuept_name& n0=v0& n1=v1...' "
            FE2_FXM_CUEPT_PARAM</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM'></td>
    </tr>

    <tr>
        <th><abbr title="Select specific metadata entries to enable. Default for each
            item is provided.">
            FE2_FXM_METADATA_ENABLE</abbr></th>
        <td>
            <table id='fxm_metadata_enable' class=''>
                <tr>
                    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr
                        title="Default: Enabled">MD_DURATION</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
                        DURATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION' value='MD_DUR
                        ATION'></td>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr
                        title="Default: Enabled">MD_DATASIZE</abbr></label></th>
                    <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
                        DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' value='MD_DAT
                        ASIZE'></td>
                </tr>

                <tr>
                    <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><ab

```

```

br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' value='MD
_AUDIO_SIZE'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><ab
br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' value='MD
_VIDEO_SIZE'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE'
><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' v
alue='MD_AUDIO_DATARATE'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE'
><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' v
alue='MD_VIDEO_DATARATE'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'>
<abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' val
ue='MD_AUDIO_CODECID'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'>
<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' val
ue='MD_VIDEO_CODECID'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr ti
tle="Default: Enabled">MD_WIDTH</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' value='MD_WIDTH'></
td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' value='MD_HEIGHT'
></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' value='MD_F

```

```

RAMERATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' value='MD_CANSEEKTOEND'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' value='MD_LASTTIMESTAMP'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAME_TIMESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAME_TIMESTAMP</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAME_TIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAME_TIMESTAMP' value='MD_LASTKEYFRAME_TIMESTAMP'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAME_LOCATION'><abbr title="Default: Disabled">MD_LASTKEYFRAME_LOCATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAME_LOCATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAME_LOCATION' value='MD_LASTKEYFRAME_LOCATION'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' value='MD_KEYFRAMES'></td>
</tr>
</table>
</td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">FE2_FXM_METADATA_DISABLE</abbr></th>
<td>
<table id='fxm_metadata_disable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abbr title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' value='MD_DURATION'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' value='MD_D

```



```

        ATASIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><a
        bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' value='
        MD_AUDIO_SIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><a
        bbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' value='
        MD_VIDEO_SIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE
        '><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'
        value='MD_AUDIO_DATARATE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE
        '><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'
        value='MD_VIDEO_DATARATE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'
        '><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' v
        alue='MD_AUDIO_CODECID'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'
        '><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' v
        alue='MD_VIDEO_CODECID'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr t
        itle="Default: Enabled">MD_WIDTH</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _WIDTH' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' value='MD_WIDTH'>
    </td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr
        title="Default: Enabled">MD_HEIGHT</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
        _HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' value='MD_HEIGH
        T'></td>
    </tr>

```

```

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' value='MD_FRAMERATE'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' value='MD_CANSEEKTOEND'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' value='MD_LASTTIMESTAMP'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP' value='MD_LASTKEYFRAMETIMESTAMP'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION' value='MD_LASTKEYFRAMELOCATION'></td>
        </tr>

        <tr>
            <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
            <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' value='MD_KEYFRAMES'></td>
        </tr>
    </table>
</td>
</tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>
    <tr>
        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_MOV_FASTSTART">
                <option value=""></option>
                <option value="on2false">FALSE</option>
                <option value="on2true">TRUE</option>
            </select>
        </td>
    </tr>
</table>

```

```

</table>
<!-- END - MOV muxer parameters -->

<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
  <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
      <select name="muxer:setParam:FE2_MP4_FASTSTART">
        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
      </select>
    </td>
  </tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
  <tr>
    <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_WIDTH'></td>
  </tr>

  <tr>
    <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_HEIGHT'></td>
  </tr>

  <tr>
    <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_FRAMERATE'></td>
  </tr>

  <tr>
    <th>FE2_SWF_LOOP_COUNT</th>
    <td><input type='text' name='muxer:setParam:FE2_SWF_LOOP_COUNT'></td>
  </tr>

  <tr>
    <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
  </tr>

  <tr>
    <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
    <td>
      <select name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>
        <option value=""></option>
        <option value="_self">_self</option>
        <option value="_blank">_blank</option>
        <option value="_parent">_parent</option>
        <option value="_top">_top</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</a
      bbr></th>
    <td>
      <select name='muxer:setParam:FE2_SWF_EMBEDDED_URL_TYPE'>
        <option value=""></option>
        <option value="EmbeddedUrlIsGetUrl">EmbeddedUrlIsGetUrl</option>
        <option value="EmbeddedUrlIsLoadMovie">EmbeddedUrlIsLoadMovie</option>
      </select>
    </td>
  </tr>

```

```

</td>
</tr>

<tr>
<th><abbr title="e.g. n0=v0& n1=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
<td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
</tr>

<tr><th><b>Preloader Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
<td>
<select name='muxer:setParam:FE2_SWF_PRELOAD_TYPE'>
<option value=""></option>
<option value="SwfPreloaderNone">SwfPreloaderNone</option>
<option value="SwfFixedPreloader">SwfFixedPreloader</option>
<option value="SwfAdaptivePreloader">SwfAdaptivePreloader</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
<th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTION</abbr></th>
<td>
<select name='muxer:setParam:FE2_SWF_ON_START_OPTION'>
<option value=""></option>
<option value="SwfOnMovieStartAutomatically">SwfOnMovieStartAutomatically</option>
<option value="SwfOnMovieStartOnClick">SwfOnMovieStartOnClick</option>
<option value="SwfOnMovieStartWait">SwfOnMovieStartWait</option>
<option value="SwfOnMovieStartEmbedSTOP">SwfOnMovieStartEmbedSTOP</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_START_BLANK_FRAME'></td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_START_WAIT_SEC'></td>
</tr>

<tr><th><b>End Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
<td>

```

```

        <select name='muxer:setParam:FE2_SWF_ON_END_OPTION'>
        <option value=""></option>
        <option value="SwfOnMovieEndNothing">SwfOnMovieEndNothing</option>
        <option value="SwfOnMovieEndSTOP">SwfOnMovieEndSTOP</option>
        <option value="SwfOnMovieEndLoop">SwfOnMovieEndLoop</option>
        <option value="SwfOnMovieEndUnload">SwfOnMovieEndUnload</option>
        <option value="SwfOnMovieEndLoadMovie">SwfOnMovieEndLoadMovie</option>
        </select>
    </td>
</tr>

    <tr>
        <th>FE2_SWF_ON_END_URL</th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ON_END_URL'></td>
    </tr>
</table>
<!-- END - SWF muxer parameters -->

<!-- WEBM muxer parameters -->
<table id='webmtable' class='hidden'>
    <tr>
        <th>(FE2_MUXER_WEBM defines no parameters)</th>
    </tr>
</table>
<!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM##### -->
<hr>
<p>
    <input type="submit" value="Start Encode">
    <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>
HTML_END

```

6.13.2 process_sample.cgi

```

#!/perl -w
##=====
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##-----
##
## File:          $Workfile$
##               $Revision$
##
## Last Update:  $DateUTC$
##
##-----
##

#process_sample.cgi
# Receive a form via post from flix2_sample.cgi, treating each name=value pair
# as a function/param pair.
# These map to the Flix Engine COM API and each function that has a valid
# param will be called.
# Once setup is complete, calls encode() to produce an output file located in out
#   dir

```

```

use CGI qw(-no_xhtml :standard);
use Win32::OLE;
## extract constants from the type lib
use Win32::OLE::Const 'On2.FlixEngine Type Library';
use Win32::OLE::Variant;

$| = 1; #auto flush

my $prefix = "C:\\Inetpub\\";
my $outdir = $prefix."flxmedia\\out\\";

sub print_encoder_status()
{
    print "<p>Encoder Status<br>";

    my $res= $flx->getEncoderState();
    print "&nbsp;$flx->getEncoderState: $res<br>";

    my $flicerr= Variant(VT_I4|VT_BYREF,0);
    my $syserr= Variant(VT_I4|VT_BYREF,0);
    $flx->errno_($flicerr,$syserr);
    printf("&nbsp;$flx->errno_: flicerrno:%d syserrno:%d</p>",$flicerr,$syserr);
}

sub process_hr($)
{
    my $funcname= $_[0];
    my $hr= Win32::OLE->LastError;

    print "<td align=\"center\">$hr</td>";

    if($hr) {
        my $ehr;
        my $flicerr= Variant(VT_I4|VT_BYREF,0);
        my $syserr= Variant(VT_I4|VT_BYREF,0);

        $flx->errno_($flicerr,$syserr);
        $ehr= Win32::OLE->LastError;

        printf("<td>hr:%.8x flix->errno_(flicerrno:%d, syserrno:%d)</td>",$ehr,$flicerr,$syserr);
        die "Error calling $funcname, hr= $hr."
            . "ehr= $ehr flix->_errno( $flicerr, $syserr )\n";
    }
    print "</tr>";
}

sub setfunc($$)
{
    my ($funcname,$funcparam) = @_;
    print "<tr><td>flix->$funcname( $funcparam )</td>";

    if($funcname eq 'setInputFile' || $funcname eq 'setOutputFile') {
        $flx->$funcname($funcparam);
    } else {
        $flx->$funcname(eval($funcparam));
    }
    process_hr("flix->$funcname");
}

sub init_codec($)
{
    my $name= $_[0];

    ##if name is a codec name, e.g., FE2_CODEC_VP6, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this codec until we
    ##hit the next codec name

```

```

    print "<tr><td>flrx->addCodec( $name )</td>";

    $codec= $flrx->addCodec($flrx->$name);
    process_hr(' flrx->addCodec');
}

sub codec_interface($$$)
{
    my ($funcname,$name,$value) = @_;
    print "<tr><td>codec->$funcname( $name, $value )</td>";

    $codec->$funcname($flrx->$name, ($funcname=~ /AsStr/) ? $value:eval($value));
    process_hr("codec->$funcname");
}

sub init_filter($)
{
    my $name= $_[0];

    ##if name is a filter name, e.g., FE2_FILTER_CUT, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this filter until we
    ##hit the next filter name
    print "<tr><td>flrx->addFilter( $name )</td>";

    $filter= $flrx->addFilter($flrx->$name);
    process_hr(' flrx->addFilter');
}

sub filter_interface($$$)
{
    my ($funcname,$name,$value) = @_;
    print "<tr><td>filter->$funcname( $name, $value )</td>";

    $filter->$funcname($flrx->$name, ($funcname=~ /AsStr/) ? $value:eval($value));
    process_hr("filter->$funcname");
}

sub init_muxer($)
{
    my $name= $_[0];

    ##if name is a muxer name, e.g., FE2_MUXER_FLV, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this muxer until we
    ##hit the next muxer name
    print "<tr><td>flrx->addMuxer( $name )</td>";

    $muxer= $flrx->addMuxer($flrx->$name);
    process_hr(' flrx->addMuxer');
}

sub muxer_interface($$$)
{
    my ($funcname,$name,$value) = @_;
    print "<tr><td>muxer->$funcname( $name, $value )</td>";

    $muxer->$funcname($flrx->$name, ($funcname=~ /AsStr/) ? $value:eval($value));
    process_hr("muxer->$funcname");
}

sub encode()
{
    ##
    ## start the encode
    ##
    print "<tr><td>flrx->encode()</td>";
    $flrx->encode(); process_hr(' flrx->encode');
}

```

```

##
## retrieve the encoding status interface, IEncodingStatus
##
print "<tr><td>flix->encodingStatus()</td>";
$encstatus= $flix->encodingStatus(); process_hr('flix->encodingStatus');

print "</table>\n";

print "<p>Encoding...(video frames encoded, percent complete). ",
      "Total frames will reset when doing 2pass.",CGI::br;
my $ier;
do {
    $ier= $flix->isEncoderRunning();
    my $tf= $encstatus->totalFrames();
    my $p= $encstatus->percentComplete();
    print "($tf, $p%)<br>";
    sleep 1;
} while($ier);
print "<br>Done!</p>";
print_encoder_status();
}

print header;
print start_html("Flix CGI Process Sample - Perl");

##
## retrieve the main engine interface, IFlix
##
$flix= Win32::OLE->new("On2.FlixEngine") or
    print("Error loading Flix Engine COM: $!, ".Win32::OLE->LastError);

print "<hr>",
      "<p>process_sample.cgi version 1.6<br>",
      "Flix Engine COM library. Flix Engine v".$flix->version(),
      " COM v".$flix->com_version()."<br>";
($_= $flix->copyright()) =~ s/\n/<br>/g;
print "$_</p>";

##verify outdir's presence and accessibility
##NOTE: cygwin's perl build (5.8.7) seems to be misreporting the directory's
##NOTE: writability when running as guest, generating a false positive here.
if(!(-d $outdir && -w $outdir)) {
    my ($i,$tt)=("<i>' $outdir'</i>","<tt>$outdir</tt>");
    print "<p>*****<br>\n".
          "<b>WARNING</b>: $i MUST exist and be writeable by".
          " <i>flixengine_com.dll</i>.\n".
          "<br>Please make $i accessible or modify the $tt".
          " value defined in '$0'.<br>\n";

    my $g=`find /N 'my \'$outdir\' '$0'`;
    if($g) {
        $g=~s/\n/<br>/g;
        print "The definition of $tt can be found here:<br>\n$g<br>\n";
    }
    print "*****</p>\n";
}

print <<EOT;
<table border='1' cellpadding='5'>
<caption>Flix Function Calls</caption>
<tr><th>Function Name</th><th>Return Value</th></tr>
EOT

foreach my $p (param()) {
    if(param($p) ne "") {
        #print "p= $p -> ".param($p)."<br>";
        if($p eq 'setOutputFile') {

```



```

        setfunc($p,$outdir.param($p));
    } elseif($p=~ /^(?:a|v)codec:/) {
        init_codec(param($p));
    } elseif(param($p)=~/ (?:a|v)filter:/) {
        init_filter($p);
    } elseif($p=~ /^muxer:/) {
        init_muxer(param($p));
    } else {
        my @temp= split(/:/,$p);
        if($temp[0] eq "codec") {
            codec_interface($temp[1],$temp[2],param($p));
        } elseif($temp[0] eq "filter") {
            filter_interface($temp[1],$temp[2],param($p));
        } elseif($temp[0] eq "muxer") {
            muxer_interface($temp[1],$temp[2],param($p));
        } else {
            setfunc($p,param($p));
        }
    }
}
}
}

encode();
print end_html;

```

6.14 Command Line

```

#!/php
<?php
#####
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
## -----
##
## File:          $Workfile$
##               $Revision$
##
## Last Update:  $DateUTC$
##
## -----
##

$cli_encode= 1;
$version    = explode('.',phpversion());

## report all errors
error_reporting(E_ALL);
## disable html w/in error text
ini_set("html_errors","0");

##
## Manually load the Flix Engine type library to allow PHP to register
## constants contained within it. PHP4 seems to ignore com.autoregister_typelib.
## In addition PHP 5.2.0 will cause an exception on exit with autoregister.
## 5.0(1).x seem to show the opposite behavior.
##
if($version[0]<5 || ($version[0]==5 && $version[1]>1)) {
    ini_set("com.autoregister_typelib","0");
    com_load_typelib("On2.FlixEngine") or
        die("Error loading Flix Engine type library");
} else {
    ini_set("com.autoregister_typelib","1");
}

```

```

## allow encodes of long files to complete
ini_set("max_execution_time","-1");
ini_set("max_input_time","-1");

function print_encoder_status()
{
    global $flix, $version;

    print "\nEncoder Status\n";

    $res= $flix->getEncoderState();
    print " flix->getEncoderState: $res\n";

    if($version[0] == 4) {
        $flixerr= $flix->flixerrno();
        $syserr= $flix->syserrno();
    } else {
        ## PHP5 can automatically detect byref parameters
        $flixerr = $syserr = 0;
        $flix->errno_($flixerr,$syserr);
    }
    printf(" flix->errno_: flixerrno:%d syserrno:%d\n",$flixerr,$syserr);
}

function configure_encode()
{
    global $argv, $flix, $version;

    ##
    ## print some library information
    ##
    print "Flix Engine COM library. Flix Engine v".$flix->version().
        " COM v".$flix->com_version()."\n";
    print $flix->copyright()."\n\n";

    if (count($argv) < 3) {
        die("usage: cli_encode.php <infile> <outfile>\n");
    }

    ##
    ## set the source file
    ##
    print "Input File : $argv[1]\n";
    $flix->setInputFile($argv[1]);

    ##
    ## retrieve the video options interface, IVideoOptions
    ##
    $vidopts= $flix->videoOptions();

    ##
    ## print input file information
    ##
    $srcduration= $flix->getSourceDuration();
    $srcw= $vidopts->getSourceWidth();
    $srch= $vidopts->getSourceHeight();
echo <<<EOT
        Width:      $srcw
        Height:     $srch
        Duration:   ${srcduration}ms

EOT;
    if($version[0] == 4) {
        ##release vidopts as we're through with it
        $vidopts->Release();
    }
}

```

```

##
## set the destination file
##
echo "Output File : $argv[2]\n";
$flix->setOutputFile($argv[2]);

##
## Options may be set and codecs/filters/muxers may be added prior to encode(
)
##

##Add the scale filter
$filter= $flix->addFilter($flix->FE2_FILTER_SCALE);
#
$filter->setParam($flix->FE2_SCALE_WIDTH,240);
$filter->setParam($flix->FE2_SCALE_HEIGHT,160);
#if($version[0] == 4) {
#    $filter->Release();
#}

##Add the vp6 codec. Though it is the default, you must add it in order
##to modify its settings
$codec= $flix->addCodec($flix->FE2_CODEC_VP6);
#
$codec->setParam($flix->FE2_VP6_RC_MODE,VBR_1PASSControl);
#if($version[0] == 4) {
#    $codec->Release();
#}

##Use the FLV muxer (default)
$muxer= $flix->addMuxer($flix->FE2_MUXER_FLV);
#
#if($version[0] == 4) {
#    $muxer->Release();
#}
} ## function configure_encode()

function encode()
{
    global $flix;

    ##
    ## start the encode
    ##
    $flix->encode();

    ##
    ## retrieve the encoding status interface, IEncodingStatus
    ##
    $encstatus= $flix->encodingStatus();

    print "\n";
    do {
        sleep(1); flush();
        $ier= $flix->isEncoderRunning();
        print "\rEncoding...".$encstatus->percentComplete()."%  ";
    } while($ier);

    echo "Done!\n";
    print_encoder_status();
} ## function encode()

##
## retrieve the main engine interface, IFlix
##
$flix= new COM("On2.FlixEngine") or die("Error loading Flix Engine COM!");

```

```
##enable logging, 0=none(disable) 1=info 2=error(asserts) 3=debug 4=heavy
##CONOUT$ can be used as the log file name to send output to the console
#$flix->setLogLevel(3);
#$flix->setLogPath("\\cli_encode.php.log");

if($version[0] == 4) {
    print "*****\n";
    print "It appears you're running under PHP4 (".phpversion().").\n".
        "This version of PHP does not provide a way to trap".
        " errors returned from the COM library aside from the ways provided".
        " by PHP itself (track_errors, set_error_handler, ...).\n".
        " Should a function fail only an E_WARNING will be issued.\n\n".
        "For this reason it is STRONGLY recommended that you consider".
        " using PHP5 as it provides try/catch blocks and throws".
        " a com_exception should a function fail.\n";
    print "*****\n\n";

    include 'cli_encode.php4';
} else {
    include 'cli_encode.php5';
}
## vim:expandtab
?>
```

6.15 CGI

This example consists of 2 parts: [flix2_sample.php](#) and [process_sample.php](#). [flix2_sample.php](#) searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's COM interface. To take advantage of PHP5's support for try/catch blocks and com_exceptions some of the processing is done in version dependent files: [process_sample.php4](#) / [process_sample.php5](#)

To use this example navigate to [flix2_sample.php](#) in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to [process_sample.php](#) which runs the encode loop.

Default file locations used by the scripts:

Input : [C:/inetpub/flixmedia/in](#) (\$indir in [flix2_sample.php](#))
Overlay : [C:/inetpub/flixmedia/overlay](#) (\$overlaydir in [flix2_sample.php](#))
Output : [C:/inetpub/flixmedia/out](#) (\$outdir in [process_sample.php](#))

6.15.1 flix2_sample.php

```
#!/php
<?php
##=====
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##-----
##
## File:          $Workfile$
##               $Revision$
##
## Last Update:  $DateUTC$
##
##-----
##
## report all errors
```

```

error_reporting(E_ALL);

$prefix      = "C:\\Inetpub\\";
$indir       = $prefix."flixmedia\\in\\";
$overlaydir  = $prefix."flixmedia\\overlay\\";

echo <<<HTML_END

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head>
  <title>Flix CGI Sample - PHP</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

  <script type="text/javascript">
function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
        i++;
    }
}

function toggle_expand(_this)
{
    _this.parentNode.className= (_this.parentNode.className=='expanded') ?
        'collapsed' : 'expanded';
    document.getElementById('showall').checked=false;
}

function set_table_visible(ctable,visible)
{
    var table= document.getElementById(ctable);

    if(table) {
        if(visible && table.className == 'hidden') {
            table.className= '';
        } else if(!visible && table.className == '') {
            table.className= 'hidden';
        }
    }
}

function hide_tablelist(vistablename,tablelist)
{
    var table= document.getElementById(vistablename);

    if(table&&table.className=='hidden') {
        table.className= '';
    }

    var i=0;
    while(tablelist[i]) {
        if(tablelist[i]!=vistablename) {
            var hiddentable= document.getElementById(tablelist[i]);

            set_table_visible(tablelist[i],false);
            /*clear down any values set in the hidden table to avoid posting
            unnecessary values*/
            clear_value(hiddentable.getElementsByTagName('input'));
            clear_value(hiddentable.getElementsByTagName('select'));
        }
        i++;
    }
}
}

```

```

function set_acodec_visible(ctable)
{
    var acodecs= new Array('aactable','aacplustable','amrnbtable','lametable','pcm
        table','vorbistable');
    hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
    var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8tabl
        e');
    hide_tablelist(ctable,vcodecs);
}

function set_muxer_visible(mtable)
{
    var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','
        tg2table','tgptable','webmtable');
    hide_tablelist(mtable,muxers);
}

function clear_value(list)
{
    var i=0;
    while(list[i]) {
        if(list[i].type=='checkbox') { list[i++].checked=false; }
        else { list[i++].value=''; }
    }
}

function toggle_ftable(ftable,enabled)
{
    var table= document.getElementById(ftable);

    if(table) {
        table.className = enabled ? '' : 'disabled';
        if(!enabled) {
            clear_value(table.getElementsByTagName('input'));
            clear_value(table.getElementsByTagName('select'));
        }
    }
}

function reset_tables()
{
    var table_list= document.getElementsByTagName('table');
    var i=0;
    while(table_list[i]) {
        if (table_list[i].id.length > 7 &&
            table_list[i].id.substring(0,7) == 'filter_') {
            table_list[i].className= 'disabled';
        } else if (table_list[i].id.indexOf('table') != -1) {
            table_list[i].className= 'hidden';
        }
        i++;
    }
}

</script>

<style type="text/css">
<!--
html {
    font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
    font-size: 100%;
    color: rgb(56,56,56);

```

```
        background-color: rgb(236,236,236);
        border-style: solid;
        border-color: rgb(236,236,236);
    }

    body {
        text-align: center;
        margin: 0 auto;
    }

    div.content {
        color: rgb(56,56,56);
        background-color: rgb(246,246,246);
        text-align: left;
        margin: 0 auto;
        width: 80%;
        min-width: 768px;
        max-width: 932px;
        border-width: 0 1px;
        border-color: rgb(144,144,144);
        border-style: solid;
    }

    div.content:after {
        content: "";
        color: inherit;
        background-color: rgb(250,250,250);
        border-top: 1px solid rgb(144,144,144);
        height: 20px;
        width: 100%;
        display: block;
    }

    fieldset table, fieldset {display: none;}
    fieldset.expanded, fieldset.collapsed {display: block;}

    /*first is fallback for IE*/
    fieldset.expanded table {display: block;}
    fieldset.expanded table {display: table;}

    h1 {
        font-family: sans-serif;
        font-size: 150%;
        font-weight: normal;
        text-align: left;
        letter-spacing: -1px;
        color: rgb(74,74,74);
        background-color: inherit;
        margin: 0;
    }

    a {
        color: rgb(74,74,74);
        background-color: transparent;
    }

    label {
        font-size: 75%;
    }

    fieldset {
        font-size: 75%;
        line-height: 130%;
        padding: 0;
        margin: 20px;
        border: none;
    }
```

```

fieldset.expanded {
    color: inherit;
    background-color: rgb(252,252,252);
    border-style: solid;
    border-width: 1px;
    border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
}

legend {
    padding: 0 5px;
    border-left: 10px solid rgb(217,217,217);
    cursor: pointer;
}
legend:hover {text-decoration: underline;}
fieldset.expanded legend {
    font-size: 150%;
    font-weight: bold;
    letter-spacing: -1px;
    background: transparent;
    margin-left: 12px;
    border-right: 10px solid rgb(217,217,217);
    display: block;
}
table {
    font-size: 100%;
    border-spacing: 0;
    /*border-collapse: collapse;*/
    width: 100%;
}

th:before { display: none; }
th, td {
    width: 50%;
    vertical-align: top;
    padding: 2px 3px;
    border-width: 1px 0;
    border-style: solid;
    border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
}
tr:first-child>* {border-top-color: rgb(252,252,252);}
th {
    font-weight: normal;
    text-align: left;
    padding: 2px 2px 2px 5px;
}
input[type] {
    font-family: monospace;
    font-size: 100%;
    color: rgb(56,56,56);
    background-color: inherit;
}
[type="text"], [type="number"] {
    margin-right: 13px;
    width: 222px;
}

[type="checkbox"].filter {
    margin-left: 4px
}
[type="checkbox"] {
    margin-left: 13px
}
[type="button"], [type="submit"] {
    font-size: 1em;
    margin: 0 2px 0 13px;
}

```



```

        fieldset {
            font-size: 75%;
            margin: 20px 10px;
        }
        fieldset input[type="text"], fieldset input[type="number"] {
            width: 95% !important;
            margin: 0;
            display: block;
        }
        fieldset.expanded table.hidden {
            display: none;
        }

        table.disabled {
            color: rgb(176,176,176);
        }
        -->
    </style>
</head>

<body>
    <div class='content'>
        <noscript>
            <p>This page requires javascript be enabled.</p>
        </noscript>

        <hr>
        <h1>Flix CGI Sample</h1>

HTML_END;
$version= explode('.',phpversion());
if($version[0] == 4) {
    print "<p>*****<br>";
    print "It appears you're running under PHP4 (".phpversion().").<br>".
        "This version of PHP does not provide a way to trap".
        " errors returned from the COM library aside from the ways provided".
        " by PHP itself (track_errors, set_error_handler, ...).".
        " Should a function fail only an E_WARNING will be issued.<br><br>".
        "For this reason it is STRONGLY recommended that you consider".
        " using PHP5 as it provides try/catch blocks and throws".
        " a com_exception should a function fail.<br><br>";
    print "*****</p>";
}
echo <<<HTML_END

<p><small>flix2_sample.cgi version 1.9</small></p>
<h4>Instructions</h4>
<ul>
    <li>In this sample you must choose a source file and an output file.<br>
    If you leave all the other options blank then the sample will not call
    the corresponding Flix Engine function and the default will be used.<br>
    When done please press the "Start Encode" button at the bottom of the page.<br>
    <li>Mouse over a function name to see its default, if applicable.
    <li>Current source file directory: $indir
</ul>

<p>
    <label><input type="checkbox" id='showall' onclick='showall(this.checked)'/>Show
    all</label>
</p>

<form action="process_sample.php" method="post">

<!-- ##SOURCE FILE***** -->

```

```

<hr>
<fieldset class='expanded' id="srcfile">
<legend onclick='toggle_expand(this)''>Source File</legend>
<table>

<tr>
<td>
HTML_END;

if(is_dir($indir)) {
    $first= TRUE;
    #let glob sort the directory contents for us
    $files= glob($indir.'*');
    echo "<select name='setInputFile' size='".(count($files)-1<15?count($files)-1
        :15)."'>\n";
    foreach($files as $f) {
        if(!is_dir($f)) {
            echo "<option ";
            if($first) { echo "selected "; $first=FALSE; };
            echo "value=\"\$f\">".str_replace($indir,'',$f)."</option>\n";
        }
    }
    echo "</select>";
} else {
    echo "WARNING couldn't open $indir: is_dir() returned FALSE<br>\n";
}

echo <<<HTML_END

</td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE#####
-->
<hr>
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)''>Output File</legend>
<table>

<tr>
<td>
<input type="text" name="setOutputFile" value="cgi-php-out.flv">
</td>
</tr>

</table>
</fieldset>

<!-- ##MAIN OPTIONS#####
### -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)''>Main Options</legend>
<table>

<tr>
<th><abbr title="Default: FALSE">setOverwriteExistingFiles</abbr></th>
<td>
<select name="setOverwriteExistingFiles">
<option value=""></option>
<option value="on2true">TRUE</option>
<option value="on2false">FALSE</option>
</select>
</td>

```

```

</tr>

<tr>
  <th><abbr title="Default: TRUE">setExportAudio</abbr></th>
  <td>
    <select name="setExportAudio">
      <option value=""></option>
      <option value="on2true">TRUE</option>
      <option value="on2false">FALSE</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="Default: TRUE">setExportVideo</abbr></th>
  <td>
    <select name="setExportVideo">
      <option value=""></option>
      <option value="on2true">TRUE</option>
      <option value="on2false">FALSE</option>
    </select>
  </td>
</tr>

</table>
</fieldset>

<!-- ##CODECS#####
      ### -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this) '>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
  <td>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
        onfocus="set_vcodec_visible('vp8table') ">
      <abbr title="For use with WebM">FE2_CODEC_VP8</abbr>&nbsp;
    </label>
    <br>

    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
        onfocus="set_vcodec_visible('vp6table') ">
      <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr>&nbsp;
    </label>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
        onfocus="set_vcodec_visible('vp6atable') ">
      <abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr>&nbsp;
    </label>
    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_H263'
        onfocus="set_vcodec_visible('h263table') ">
      <abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr>&nbsp;
    </label>
    <br>

    <label>
      <input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
        onfocus="set_vcodec_visible('h263table') ">
      <abbr title="For use with 3GP">FE2_CODEC_H263_BASELINE</abbr>&nbsp;
    </label>
    <label>

```

```



```

```

    <th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_CONCURRENCY'></td>
</tr>

<tr>
    <th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_UNDERSHOOT_PCT'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_MIN_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_MAX_Q'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_SHARPNESS'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_NOISE_REDUCTION'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6_TEMPORAL_RESAMPLING</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
        FE2_VP6_TEMPORAL_DOWN_WATERMARK</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
    <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
    <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_PREBUFFER'></td>
</tr>

<tr>
    <th><abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
    <th><abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>

```

```

        <td><input type='text' name='codec:setParam:FE2_VP6_STREAM_MAX_BUFFER'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6_2PASS_MIN_SECTION'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6_2PASS_MAX_SECTION'></td>
    </tr>
</table>
<!-- END - VP6 codec parameters -->

<!-- VP6A codec parameters -->
<table id='vp6atable' class='hidden'>
    <tr>
        <th><abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_BITRATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 68kbps (15% of default 448kbps)">
            FE2_VP6A_ALPHA_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_BITRATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>
        <td>
            <select name="codec:setParam:FE2_VP6A_KFINTTYPE">
                <option value=""></option>
                <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
                <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
            FE2_VP6A_KFFREQ</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VP6A_KFFREQ'></td>
    </tr>

    <tr>
        <th><abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>
        <td>
            <select name="codec:setParam:FE2_VP6A_RC_MODE">
                <option value=""></option>
                <option value="VBR_2PASSControl">VBR_2PASSControl</option>
                <option value="CBR_2PASSControl">CBR_2PASSControl</option>
                <option value="VBR_1PASSControl">VBR_1PASSControl</option>
                <option value="CBR_1PASSControl">CBR_1PASSControl</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>
        <td>
            <select name="codec:setParam:FE2_VP6A_CXMODE">
                <option value=""></option>
                <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
                <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
            </select>
        </td>
    </tr>

```

```

    </td>
  </tr>

  <tr><th><b>Advanced Settings:</b></th></tr>

  <tr>
    <th><abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_MIN_Q'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_ALPHA_MIN_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MIN_Q'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_MAX_Q'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_MAX_Q</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_MAX_Q'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_SHARPNESS'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_NOISE_REDUCTION'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
  </tr>

  <tr>
    <th><abbr title="Default: bits per pixel dependent (see API docs)">
      FE2_VP6A_TEMPORAL_RESAMPLING</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
  </tr>

  <tr>

```

```

<th><abbr title="Default: bits per pixel dependent (see API docs)">
  FE2_VP6A_TEMPORAL_DOWN_WATERMARK</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'>
</td>
</tr>

<tr>
<th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_PREBUFFER'></td>
</tr>

<tr>
<th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr>
</th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
<th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
<th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MIN_SECTION'></td>
</tr>

<tr>
<th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>
<td><input type='text' name='codec:setParam:FE2_VP6A_2PASS_MAX_SECTION'></td>
</tr>
</table>
<!-- END - VP6A codec parameters -->

<!-- H263 codec parameters -->
<table id='h263table' class='hidden'>
<tr>
<th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_BITRATE'></td>
</tr>

<tr>
<th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>
<td>
<select name="codec:setParam:FE2_H263_KFINTTYPE">
<option value=""></option>
<option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
<option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
  FE2_H263_KFFREQ</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_KFFREQ'></td>
</tr>

<tr>

```



```

<th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>
<td>
  <select name="codec:setParam:FE2_H263_RC_MODE">
    <option value=""></option>
    <option value="VBR_2PASSControl">VBR_2PASSControl</option>
    <option value="CBR_2PASSControl">CBR_2PASSControl</option>
    <option value="VBR_1PASSControl">VBR_1PASSControl</option>
    <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MAX_Q'></td>
</tr>

<tr>
<th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
<td><input type='text' name='codec:setParam:FE2_H263_MIN_Q'></td>
</tr>

</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H264_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_H264_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H264_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_H264_RC_MODE">
        <option value=""></option>
        <option value="VBR_1PASSControl">VBR_1PASSControl</option>
        <option value="CBR_1PASSControl">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_H264_PROFILE">
        <option value=""></option>
        <option value="BASE_H264PROFILE">BASE_H264PROFILE</option>
        <option value="MAIN_H264PROFILE">MAIN_H264PROFILE</option>
        <option value="HIGH_H264PROFILE">HIGH_H264PROFILE</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H264_B_FRAME_RATE'></td>
  </tr>

  <tr>

```

```

    <th><abbr title="Default: Dependent on profile selection, see API docs. Valid
      Range [0,5]">FE2_H264_SPEED</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_H264_SPEED'></td>
  </tr>

</table>
<!-- END - H264 codec parameters -->

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
  <tr>
    <th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_BITRATE'></td>
  </tr>

  <tr>
    <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_KFINTTYPE">
        <option value=""></option>
        <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
        <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">
      FE2_VP8_KFFREQ</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_KFFREQ'></td>
  </tr>

  <tr>
    <th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_RC_MODE">
        <option value=""></option>
        <option value="VBR_2PASSControl">VBR_2PASSControl</option>
        <option value="CBR_2PASSControl">CBR_2PASSControl</option>
        <option value="VBR_1PASSControl">VBR_1PASSControl</option>
        <option value="CBR_1PASSControl">CBR_1PASSControl</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
    <td>
      <select name="codec:setParam:FE2_VP8_CXMODE">
        <option value=""></option>
        <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
        <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
      </select>
    </td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_THREADS'></td>
  </tr>

  <tr>
    <th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
    <td><input type='text' name='codec:setParam:FE2_VP8_PROFILE'></td>
  </tr>

  <tr><th><b>Advanced Settings:</b></th></tr>

```

```

<tr>
  <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_LAG'></td>
</tr>

<tr>
  <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_UNDERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_OVERSHOOT_PCT'></td>
</tr>

<tr>
  <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MIN_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
  <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
  <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>
  <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='codec:setParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

```

```
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>  <input type='text' name='codec:setParam:FE2_VP8_ALTREF'></td>   <abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th>  <input type='text' name='codec:setParam:FE2_VP8_AR_MAX_FRAMES'></td>   <abbr title="">FE2_VP8_AR_TYPE</abbr></th>  <input type='text' name='codec:setParam:FE2_VP8_AR_TYPE'></td>   <abbr title="">FE2_VP8_AR_STRENGTH</abbr></th>  <input type='text' name='codec:setParam:FE2_VP8_AR_STRENGTH'></td>   <abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th>  <input type='text' name='codec:setParam:FE2_VP8_MB_STATIC_THRESHOLD'></td>   <abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th>  <input type='text' name='codec:setParam:FE2_VP8_TOKEN_PARTITIONS'></td> | | | | | | | | | | | |

```

```

    <abbr title="For use with FLV/SWF">FE2_CODEC_PCM</abbr>&nbsp;
</label>
<br>

<label>
    <input type='radio' name='acodec:' value='FE2_CODEC_AMR_NB'
        onfocus="set_acodec_visible('amrnhtable')">
    <abbr title="For use with 3GP">FE2_CODEC_AMR_NB</abbr>&nbsp;
</label>

<!-- AMR_NB codec parameters -->
<table id='amrnhtable' class='hidden'>
    <tr>
        <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_AMR_BITRATE'></td>
    </tr>
</table>
<!-- END - AMR_NB codec parameters -->

<!-- AAC codec parameters -->
<table id='aactable' class='hidden'>
    <tr>
        <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_AAC_BITRATE'></td>
    </tr>
</table>
<!-- END - AAC codec parameters -->

<!-- AACPLUS codec parameters -->
<table id='aacplustable' class='hidden'>
    <tr>
        <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_AACPLUS_BITRATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: disabled (aacPlus v1)">
            FE2_AACPLUS_PARAMETRIC_STEREO</abbr></th>
        <td>
            <select name="codec:setParam:FE2_AACPLUS_PARAMETRIC_STEREO">
                <option value=""></option>
                <option value="0">disable (aacPlus v1)</option>
                <option value="1">enable (aacPlus v2)</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - AACPLUS codec parameters -->

<!-- LAME codec parameters -->
<table id='lametable' class='hidden'>
    <tr>
        <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_LAME_BITRATE'></td>
    </tr>

    <tr>
        <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_LAME_QUALITY'></td>
    </tr>

    <tr>
        <th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
        <td>
            <select name="codec:setParam:FE2_LAME_RC_MODE">

```

```

        <option value=""></option>
        <option value="LAME_CBR">LAME_CBR</option>
        <option value="LAME_ABR">LAME_ABR</option>
        <option value="LAME_VBR_rh">LAME_VBR_rh</option>
        <option value="LAME_VBR_mtrh">LAME_VBR_mtrh</option>
    </select>
</td>
</tr>
</table>
<!-- END - LAME codec parameters -->

<!-- PCM codec parameters -->
<table id='pcmtable' class='hidden'>
    <tr>
        <th>(FE2_CODEEC_PCM defines no parameters)</th>
    </tr>
</table>
<!-- END - PCM codec parameters -->

<!-- VORBIS codec parameters -->
<table id='vorbistable' class='hidden'>
    <tr>
        <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
        <td><input type='text' name='codec:setParam:FE2_VORBIS_BITRATE'></td>
    </tr>
</table>
<!-- END - VORBIS codec parameters -->

</td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS#####
    ### -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this)''>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
<td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_CUT' value='avfilter:'
                onchange="toggle_ftable('filter_cut',this.checked)">
            FE2_FILTER_CUT
        </label>

        <table id='filter_cut' class='disabled'>
            <tr>
                <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_CUT_START_SEC'></td>
            </tr>

            <tr>
                <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_CUT_STOP_SEC'></td>
            </tr>

            <tr>
                <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>
                <td>

```

```

        <select name="filter:setParam:FE2_CUT_USE_SEEK">
        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
        </select>
    </td>
</tr>

</table>

</td>
</tr>
<!-- END - CUT filter parameters -->

<tr><th><b>Video Filters</b></th></tr>

<tr>
<td>
<label>
<input type='checkbox' class='filter'
        name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:'
        onchange="toggle_fhtable('filter_adaptive_deinterlace',this.checked) ">
        FE2_FILTER_ADAPTIVE_DEINTERLACE
</label>

<table id='filter_adaptive_deinterlace' class='disabled'>
<tr>
<th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</abbr></th>
<td>
<select name="filter:setParam:FE2_ADAPTIVE_DEINTERLACE_MODE">
<option value=""></option>
<option value="DEINTERLACE_NONE">DEINTERLACE_NONE</option>
<option value="DEINTERLACE_1_2_1_BLUR">DEINTERLACE_1_2_1_BLUR</option>
<option value="DEINTERLACE_DROP_FIELD">DEINTERLACE_DROP_FIELD</option>
<option value="DEINTERLACE_ADAPTIVE">DEINTERLACE_ADAPTIVE</option>
</select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - ADAPTIVE DEINTERLACE filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
        name='FE2_FILTER_BCHS' value='vfilter:'
        onchange="toggle_fhtable('filter_bchs',this.checked) ">
        FE2_FILTER_BCHS
</label>

<table id='filter_bchs' class='disabled'>
<tr>
<th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th>
<td><input type='text' name='filter:setParam:FE2_BCHS_BRIGHTNESS'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
<td><input type='text' name='filter:setParam:FE2_BCHS_CONTRAST'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
<td><input type='text' name='filter:setParam:FE2_BCHS_HUE'></td>
</tr>

```

```

        <tr>
        <th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_BCHS_SATURATION'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - BCHS filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
        name='FE2_FILTER_BLUR' value='vfilter:'
        onchange="toggle_ftable('filter_blur',this.checked)">
    FE2_FILTER_BLUR
</label>

<table id='filter_blur' class='disabled'>
<tr>
<th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
<td>
<select name="filter:setParam:FE2_BLUR_FILTER">
<option value=""></option>
<option value="BLUR_LOWPASS">BLUR_LOWPASS</option>
<option value="BLUR_GAUSS">BLUR_GAUSS</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
<td>
<select name="filter:setParam:FE2_BLUR_MASKSIZE">
<option value=""></option>
<option value="MASK_3x3">MASK_3x3</option>
<option value="MASK_5x5">MASK_5x5</option>
</select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
        name='FE2_FILTER_CROP' value='vfilter:'
        onchange="toggle_ftable('filter_crop',this.checked)">
    FE2_FILTER_CROP
</label>

<table id='filter_crop' class='disabled'>
<tr>
<th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_TOP'></td>
</tr>

<tr>
<th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>
<td><input type='text' name='filter:setParam:FE2_CROP_BOTTOM'></td>
</tr>

```



```

        <tr>
            <th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_CROP_LEFT'></td>
        </tr>

        <tr>
            <th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>
            <td><input type='text' name='filter:setParam:FE2_CROP_RIGHT'></td>
        </tr>
    </table>

</td>
</tr>
<!-- END - CROP filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_DENOISE' value='vfilter:'
                onchange="toggle_ftable('filter_denoise',this.checked)">
            FE2_FILTER_DENOISE
        </label>

        <table id='filter_denoise' class='disabled'>
            <tr>
                <th><abbr title="Default: 0. Range: [0.0,1.0]">FE2_DENOISE_NOISE_LEVEL</abbr>
                </th>
                <td><input type='text' name='filter:setParam:FE2_DENOISE_NOISE_LEVEL'></td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'
                name='FE2_FILTER_FRAMERATE' value='vfilter:'
                onchange="toggle_ftable('filter_framerate',this.checked)">
            FE2_FILTER_FRAMERATE
        </label>

        <table id='filter_framerate' class='disabled'>
            <tr>
                <th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAM
                    ERATE_DECIMATE</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_FRAMERATE_DECIMATE'></td>
            </tr>

            <tr>
                <th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FR
                    AMERATE_FPS</abbr></th>
                <td><input type='text' name='filter:setParam:FE2_FRAMERATE_FPS'></td>
            </tr>
        </table>

    </td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>
    <td>
        <label>
            <input type='checkbox' class='filter'

```

```

        name='FE2_FILTER_MIRROR' value='vfilter:'
        onchange="toggle_ftable('filter_mirror',this.checked)">
FE2_FILTER_MIRROR
</label>

<table id='filter_mirror' class='disabled'>
<tr>
<th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
<td>
<select name="filter:setParam:FE2_MIRROR_HORIZONTAL">
<option value=""></option>
<option value="on2false">FALSE</option>
<option value="on2true">TRUE</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
<td>
<select name="filter:setParam:FE2_MIRROR_VERTICAL">
<option value=""></option>
<option value="on2false">FALSE</option>
<option value="on2true">TRUE</option>
</select>
</td>
</tr>
</table>

</td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
        name='FE2_FILTER_OVERLAY' value='vfilter:'
        onchange="toggle_ftable('filter_overlay',this.checked)">
FE2_FILTER_OVERLAY
</label>

<table id='filter_overlay' class='disabled'>
<tr>
<th>
<abbr title="Currently searching $overlaydir for overlay images">FE2_OVERLAY
_FILE</abbr>
</th>
<td>
HTML_END;

if(is_dir($overlaydir)) {
    echo "<select name=\"filter:setParamAsStr:FE2_OVERLAY_FILE\">\n";
    echo "<option value=\"\"></option>\n";
    foreach(glob($overlaydir.'*') as $f) {
        if ((ereg("\.bmp",$f) || ereg("\.png",$f)) && !is_dir($f)) {
            echo "<option value=\"$f\">".
                str_replace($overlaydir,'',$f)."</option>\n";
        }
    }
    echo "</select>\n";
    echo "<br>default: None, must be set to the absolute path of the overlay";
    echo " input file, e.g. '/path/to/my/overlay.png'";
} else {
    echo "WARNING couldn't open $overlaydir: is_dir() returned FALSE<br>\n";
}

```

```
echo <<<HTML_END
```

```

    </td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
<td>
  <select name="filter:setParam:FE2_OVERLAY_MASK_XY">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_Y'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
<td>
  <select name="filter:setParam:FE2_OVERLAY_MASK_RGB">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
  </select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_R'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_G'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_MASK_B'></td>
</tr>

<tr>
<th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
<td>
  <select name="filter:setParam:FE2_OVERLAY_POS">
    <option value=""></option>
    <option value="FE2_OVERLAY_POS_MODE_TOPLEFT">FE2_OVERLAY_POS_MODE_TOPLEFT</option>
    <option value="FE2_OVERLAY_POS_MODE_BOTLEFT">FE2_OVERLAY_POS_MODE_BOTLEFT</option>
    <option value="FE2_OVERLAY_POS_MODE_CENTER">FE2_OVERLAY_POS_MODE_CENTER</option>
    <option value="FE2_OVERLAY_POS_MODE_TOPRIGHT">FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
    <option value="FE2_OVERLAY_POS_MODE_BOTRIGHT">FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
    <option value="FE2_OVERLAY_POS_MODE_XY">FE2_OVERLAY_POS_MODE_XY</option>
  </select>
</td>

```

```

</tr>

<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_X'></td>
</tr>
<tr>
<th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
<td><input type='text' name='filter:setParam:FE2_OVERLAY_POS_Y'></td>
</tr>

</table>

</td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
<td>
<label>
<input type='checkbox' class='filter'
name='FE2_FILTER_PNGEX' value='vfilter:'
onchange="toggle_ftable('filter_pngex',this.checked)">
FE2_FILTER_PNGEX
</label>

<table id='filter_pngex' class='disabled'>
<tr>
<th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'></td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_PREFIX'>
</td>
</tr>

<tr>
<th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
<td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'>
</td>
</tr>

<tr>
<th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_WIDTH'></td>
</tr>

<tr>
<th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
<td><input type='text' name='filter:setParam:FE2_PNGEX_HEIGHT'></td>
</tr>

<tr>
<th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>

<td>
<select name="filter:setParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">
<option value=""></option>
<option value="on2false">FALSE</option>
<option value="on2true">TRUE</option>
</select>
</td>
</tr>

```

```

<tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
  <td>
    <select name="filter:setParam:FE2_PNGEX_ENABLE_ALPHA">
      <option value=""></option>
      <option value="on2false">FALSE</option>
      <option value="on2true">TRUE</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="interval in ms; Default: disabled">
    FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
  <td><input type='text' name='filter:setParam:FE2_PNGEX_EXPORT_INTERVAL'></td>
</tr>

<tr>
  <th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">
    FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
  <td><input type='text' name='filter:setParamAsStr:FE2_PNGEX_EXPORT_TIME_STRIN
    G'></td>
</tr>

<tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
  <td>
    <select name="filter:setParam:FE2_PNGEX_EXPORT_CUE_POINTS">
      <option value=""></option>
      <option value="FE2_PNGEX_CP_ALL">All cue points (FE2_PNGEX_CP_ALL)</option>
      <option value="FE2_PNGEX_CP_NAV">Only navigation cue points (
        FE2_PNGEX_CP_NAV)</option>
      <option value="FE2_PNGEX_CP_EVENT">Only event cue points (
        FE2_PNGEX_CP_EVENT)</option>
    </select>
  </td>
</tr>

<tr>
  <th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">
    FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>
  <td><input type='text' name='filter:setParam:FE2_PNGEX_COMPRESSION_LEVEL'></td>
</tr>

<tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>

<tr>
  <th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>
  <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></td>
</tr>

<tr>
  <th><abbr title="start time in ms; Default: 0">
    FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
  <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_START_TIME
    '></td>
</tr>

<tr>
  <th><abbr title="stop time in ms; Default: <clip length>">
    FE2_PNGEX_AUTO_EXPORT_END_TIME</abbr></th>
  <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_END_TIME'>
    </td>
</tr>

```

```

    </tr>

    <tr>
    <th><abbr title="Default: 0">FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_PNGEX_AUTO_EXPORT_RANDOM_PER
        IOD'></td>
    </tr>

</table>

</td>
</tr>
<!-- END - PNGEX filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_ROTATE' value='vfilter:'
            onchange="toggle_ftable('filter_rotate',this.checked)">
        FE2_FILTER_ROTATE
    </label>

    <table id='filter_rotate' class='disabled'>
    <tr>
    <th><abbr title="Default: 0. valid: {0,90,180,270}">FE2_ROTATE_ANGLE</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_ROTATE_ANGLE'></td>
    </tr>
    </table>

</td>
</tr>
<!-- END - ROTATE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_SCALE' value='vfilter:'
            onchange="toggle_ftable('filter_scale',this.checked)">
        FE2_FILTER_SCALE
    </label>

    <table id='filter_scale' class='disabled'>
    <tr>
    <th><abbr title="Default: input image width">FE2_SCALE_WIDTH</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_SCALE_WIDTH'></td>
    </tr>

    <tr>
    <th><abbr title="Default: input image height">FE2_SCALE_HEIGHT</abbr></th>
    <td><input type='text' name='filter:setParam:FE2_SCALE_HEIGHT'></td>
    </tr>
    </table>

</td>
</tr>
<!-- END - SCALE filter parameters -->

<tr>
<td>
    <label>
        <input type='checkbox' class='filter'
            name='FE2_FILTER_SHARPEN' value='vfilter:'
            onchange="toggle_ftable('filter_sharpen',this.checked)">
        FE2_FILTER_SHARPEN

```

```

</label>

<table id='filter_sharpen' class='disabled'>
  <tr>
    <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
  </tr>
</table>

</td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_HIGHPASS' value='afilter:'
        onchange="toggle_ftable('filter_highpass',this.checked)">
      FE2_FILTER_HIGHPASS
    </label>

    <table id='filter_highpass' class='disabled'>
      <tr>
        <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_HIGHPASS_Q'></td>
      </tr>

      <tr>
        <th>FE2_HIGHPASS_CUTOFF</th>
        <td><input type='text' name='filter:setParam:FE2_HIGHPASS_CUTOFF'></td>
      </tr>
    </table>

  </td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
  <td>
    <label>
      <input type='checkbox' class='filter'
        name='FE2_FILTER_LOWPASS' value='afilter:'
        onchange="toggle_ftable('filter_lowpass',this.checked)">
      FE2_FILTER_LOWPASS
    </label>

    <table id='filter_lowpass' class='disabled'>
      <tr>
        <th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>
        <td><input type='text' name='filter:setParam:FE2_LOWPASS_Q'></td>
      </tr>

      <tr>
        <th>FE2_LOWPASS_CUTOFF</th>
        <td><input type='text' name='filter:setParam:FE2_LOWPASS_CUTOFF'></td>
      </tr>
    </table>

  </td>
</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
  <td>
    <label>

```

```

☐

```



```

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
    onfocus="set_muxer_visible('fxmtable',true)">
  FE2_MUXER_FXM 
</label>
<br>

<label>
  <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
    onfocus="set_muxer_visible('webmtable',true)">
  FE2_MUXER_WEBM 
</label>
</td>
</tr>

<tr>
<td>
<!-- 3GP muxer parameters -->
<table id='tgptable' class='hidden'>
  <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
      <select name="muxer:setParam:FE2_3GP_FASTSTART">
        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
      </select>
    </td>
  </tr>
</table>
<!-- END - 3GP muxer parameters -->

<!-- 3G2 muxer parameters -->
<table id='tg2table' class='hidden'>
  <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
      <select name="muxer:setParam:FE2_3G2_FASTSTART">
        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
      </select>
    </td>
  </tr>
</table>
<!-- END - 3G2 muxer parameters -->

<!-- FLV muxer parameters -->
<table id='flvtable' class='hidden'>
  <tr>
    <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0' ">
      >
      FE2_FLV_CUEPT_EVENT</abbr></th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
  </tr>

  <tr>
    <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0' ">
      >
      FE2_FLV_CUEPT_NAV</abbr></th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_NAV'></td>
  </tr>

  <tr>
    <th><abbr title="e.g. 'cuept_name&#0=v0&#0=v1...' ">
      >
      FE2_FLV_CUEPT_PARAM</abbr></th>

```

```

<td><input type='text' name='muxer:setParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
FE2_FLV_METADATA_ENABLE</abbr></th>
<td>
<table id='flv_metadata_enable' class=''>
<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr
title="Default: Enabled">MD_DURATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
DURATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' value='MD_DUR
ATION'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr
title="Default: Enabled">MD_DATASIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' value='MD_DAT
ASIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><ab
br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' value='MD_
AUDIO_SIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><ab
br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' value='MD_
VIDEO_SIZE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'>
<abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' v
alue='MD_AUDIO_DATARATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'>
<abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' v
alue='MD_VIDEO_DATARATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'>
<abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' val
ue='MD_AUDIO_CODECID'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'>

```

```

<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' val
ue='MD_VIDEO_CODECID'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr ti
tle="Default: Enabled">MD_WIDTH</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' value='MD_WIDTH'></
td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' value='MD_HEIGHT'
></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' value='MD_F
RAMERATE'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' value
='MD_CANSEEKTOEND'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'>
<abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' val
ue='MD_LASTTIMESTAMP'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIM
ESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></
th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRA
METIMESTAMP' value='MD_LASTKEYFRAMETIMESTAMP'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOC
ATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
  <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAM
ELOCATION' value='MD_LASTKEYFRAMELOCATION'></td>
</tr>

<tr>
  <th><label for='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abb
r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>

```

```

        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_
        KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' value='MD_K
        EYFRAMES'></td>
    </tr>
</table>
</td>
</tr>

<tr>
<th><abbr title="Select specific metadata entries to enable. Default for each
    item is provided.">
        FE2_FLV_METADATA_DISABLE</abbr></th>
<td>
<table id='flv_metadata_disable' class=''>
    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abb
        r title="Default: Enabled">MD_DURATION</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
        _DURATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' value='MD_D
        URATION'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abb
        r title="Default: Enabled">MD_DATASIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
        _DATASIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' value='MD_D
        ATASIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><a
        bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
        _AUDIO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' value='
        MD_AUDIO_SIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><a
        bbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
        _VIDEO_SIZE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' value='
        MD_VIDEO_SIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE
        '><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
        _AUDIO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'
        value='MD_AUDIO_DATARATE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE
        '><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
        _VIDEO_DATARATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'
        value='MD_VIDEO_DATARATE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID'
        ><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
        _AUDIO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' v

```

```

alue='MD_AUDIO_CODECID'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID'
><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_VIDEO_CODECID' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' v
alue='MD_VIDEO_CODECID'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr t
itle="Default: Enabled">MD_WIDTH</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_WIDTH' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' value='MD_WIDTH'>
</td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr
title="Default: Enabled">MD_HEIGHT</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_HEIGHT' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' value='MD_HEIGH
T'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><ab
br title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_FRAMERATE' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' value='MD
_FRAMERATE'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'>
<abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_CANSEEKTOEND' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' val
ue='MD_CANSEEKTOEND'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP'
><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTTIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' v
alue='MD_LASTTIMESTAMP'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETI
MESTAMP'><abbr title="Default: Enabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></
th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYF
RAMETIMESTAMP' value='MD_LASTKEYFRAMETIMESTAMP'></td>
</tr>

<tr>
<th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Enabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
<td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='MD_LASTKEYFRAMELOCATION'></td>

```

```

    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><abbr
      title="Default: Enabled">MD_KEYFRAMES</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_
      _KEYFRAMES' name='muxer:setParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES' value='MD
      _KEYFRAMES'></td>
    </tr>
  </table>
</td>
</tr>
</table>
<!-- END - FLV muxer parameters -->

<!-- FXM muxer parameters -->
<table id='fxmtable' class='hidden'>
  <tr>
    <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'"
    >
      FE2_FXM_CUEPT_EVENT</abbr></th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_EVENT'></td>
  </tr>

  <tr>
    <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.
    0'">
      FE2_FXM_CUEPT_NAV</abbr></th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_NAV'></td>
  </tr>

  <tr>
    <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'">
      FE2_FXM_CUEPT_PARAM</abbr></th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_FXM_CUEPT_PARAM'></td>
  </tr>

  <tr>
    <th><abbr title="Select specific metadata entries to enable. Default for each
    item is provided.">
      FE2_FXM_METADATA_ENABLE</abbr></th>
    <td>
      <table id='fxm_metadata_enable' class=''>
        <tr>
          <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr
          title="Default: Enabled">MD_DURATION</abbr></label></th>
          <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
          DURATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DURATION' value='MD_DUR
          ATION'></td>
        </tr>

        <tr>
          <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr
          title="Default: Enabled">MD_DATASIZE</abbr></label></th>
          <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
          DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' value='MD_DAT
          ASIZE'></td>
        </tr>

        <tr>
          <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><ab
          br title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>
          <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
          AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' value='MD
          _AUDIO_SIZE'></td>
        </tr>
      </table>
    </td>
  </tr>
</table>

```

```

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><abbr
br title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' value='MD
_VIDEO_SIZE'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE'
><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' v
alue='MD_AUDIO_DATARATE'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE'
><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' v
alue='MD_VIDEO_DATARATE'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'>
<abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' val
ue='MD_AUDIO_CODECID'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'>
<abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' val
ue='MD_VIDEO_CODECID'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr ti
tle="Default: Enabled">MD_WIDTH</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
WIDTH' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' value='MD_WIDTH'></
td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr t
itle="Default: Enabled">MD_HEIGHT</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' value='MD_HEIGHT'
></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abb
r title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' value='MD_F
RAMERATE'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><
abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>

```

```

        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' value
='MD_CANSEEKTOEND'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'>
<abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' val
ue='MD_LASTTIMESTAMP'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIM
ESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRA
METIMESTAMP' value='MD_LASTKEYFRAMETIMESTAMP'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOC
ATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAM
ELOCATION' value='MD_LASTKEYFRAMELOCATION'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abb
r title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_
KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' value='MD_K
EYFRAMES'></td>
    </tr>
</table>
</td>
</tr>

<tr>
    <th><abbr title="Select specific metadata entries to enable. Default for each
item is provided.">
        FE2_FXM_METADATA_DISABLE</abbr></th>
    <td>
        <table id='fxm_metadata_disable' class=''>
            <tr>
                <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abb
r title="Default: Enabled">MD_DURATION</abbr></label></th>
                <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_
_DURATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' value='MD_D
URATION'></td>
            </tr>

            <tr>
                <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abb
r title="Default: Enabled">MD_DATASIZE</abbr></label></th>
                <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_
_DATASIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' value='MD_D
ATASIZE'></td>
            </tr>

            <tr>
                <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><a
bbr title="Default: Enabled">MD_AUDIO_SIZE</abbr></label></th>

```



```

        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_AUDIO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' value='
MD_AUDIO_SIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><a
abbr title="Default: Enabled">MD_VIDEO_SIZE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_VIDEO_SIZE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' value='
MD_VIDEO_SIZE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE
'><abbr title="Default: Enabled">MD_AUDIO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_AUDIO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'
value='MD_AUDIO_DATARATE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE
'><abbr title="Default: Enabled">MD_VIDEO_DATARATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_VIDEO_DATARATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'
value='MD_VIDEO_DATARATE'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'
><abbr title="Default: Enabled">MD_AUDIO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_AUDIO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' v
alue='MD_AUDIO_CODECID'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'
><abbr title="Default: Enabled">MD_VIDEO_CODECID</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_VIDEO_CODECID' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' v
alue='MD_VIDEO_CODECID'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr t
itle="Default: Enabled">MD_WIDTH</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_WIDTH' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' value='MD_WIDTH'>
</td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr
title="Default: Enabled">MD_HEIGHT</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_HEIGHT' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' value='MD_HEIGH
T'></td>
    </tr>

    <tr>
        <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><ab
br title="Default: Enabled">MD_FRAMERATE</abbr></label></th>
        <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_FRAMERATE' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' value='MD
_FRAMERATE'></td>
    </tr>

```

```

    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'>
<abbr title="Default: Enabled">MD_CANSEEKTOEND</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_CANSEEKTOEND' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' val
ue='MD_CANSEEKTOEND'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'
><abbr title="Default: Enabled">MD_LASTTIMESTAMP</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTTIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' v
alue='MD_LASTTIMESTAMP'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETI
MESTAMP'><abbr title="Default: Disabled">MD_LASTKEYFRAMETIMESTAMP</abbr></label><
/th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMETIMESTAMP' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYF
RAMETIMESTAMP' value='MD_LASTKEYFRAMETIMESTAMP'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELO
CATION'><abbr title="Default: Disabled">MD_LASTKEYFRAMELOCATION</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_LASTKEYFRAMELOCATION' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFR
AMELOCATION' value='MD_LASTKEYFRAMELOCATION'></td>
    </tr>

    <tr>
      <th><label for='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><ab
br title="Default: Disabled">MD_KEYFRAMES</abbr></label></th>
      <td><input type='checkbox' id='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD
_KEYFRAMES' name='muxer:setParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' value='MD
_KEYFRAMES'></td>
    </tr>
  </table>
</td>
</tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>
  <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
      <select name="muxer:setParam:FE2_MOV_FASTSTART">
        <option value=""></option>
        <option value="on2false">FALSE</option>
        <option value="on2true">TRUE</option>
      </select>
    </td>
  </tr>
</table>
<!-- END - MOV muxer parameters -->

<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
  <tr>

```

```

        <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
        <td>
            <select name="muxer:setParam:FE2_MP4_FASTSTART">
                <option value=""></option>
                <option value="on2false">FALSE</option>
                <option value="on2true">TRUE</option>
            </select>
        </td>
    </tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
    <tr>
        <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_WIDTH'></td>
    </tr>

    <tr>
        <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_HEIGHT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_FRAMERATE'></td>
    </tr>

    <tr>
        <th>FE2_SWF_LOOP_COUNT</th>
        <td><input type='text' name='muxer:setParam:FE2_SWF_LOOP_COUNT'></td>
    </tr>

    <tr>
        <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
    </tr>

    <tr>
        <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
        <td>
            <select name='muxer:setParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>
                <option value=""></option>
                <option value="_self">_self</option>
                <option value="_blank">_blank</option>
                <option value="_parent">_parent</option>
                <option value="_top">_top</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</a
            bbr></th>
        <td>
            <select name='muxer:setParam:FE2_SWF_EMBEDDED_URL_TYPE'>
                <option value=""></option>
                <option value="EmbeddedUrlIsGetUrl">EmbeddedUrlIsGetUrl</option>
                <option value="EmbeddedUrlIsLoadMovie">EmbeddedUrlIsLoadMovie</option>
            </select>
        </td>
    </tr>

    <tr>
        <th><abbr title="e.g. n0=v0& n1=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
        <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
    </tr>

```

```

</tr>

<tr><th><b>Preloader Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
<td>
<select name='muxer:setParam:FE2_SWF_PRELOAD_TYPE'>
<option value=""></option>
<option value="SwfPreloaderNone">SwfPreloaderNone</option>
<option value="SwfFixedPreloader">SwfFixedPreloader</option>
<option value="SwfAdaptivePreloader">SwfAdaptivePreloader</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
<th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTION</abbr></th>
<td>
<select name='muxer:setParam:FE2_SWF_ON_START_OPTION'>
<option value=""></option>
<option value="SwfOnMovieStartAutomatically">SwfOnMovieStartAutomatically</option>
<option value="SwfOnMovieStartOnClick">SwfOnMovieStartOnClick</option>
<option value="SwfOnMovieStartWait">SwfOnMovieStartWait</option>
<option value="SwfOnMovieStartEmbedSTOP">SwfOnMovieStartEmbedSTOP</option>
</select>
</td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_START_BLANK_FRAME'></td>
</tr>

<tr>
<th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
<td><input type='text' name='muxer:setParam:FE2_SWF_START_WAIT_SEC'></td>
</tr>

<tr><th><b>End Settings:</b></th><td></td></tr>

<tr>
<th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
<td>
<select name='muxer:setParam:FE2_SWF_ON_END_OPTION'>
<option value=""></option>
<option value="SwfOnMovieEndNothing">SwfOnMovieEndNothing</option>
<option value="SwfOnMovieEndSTOP">SwfOnMovieEndSTOP</option>
<option value="SwfOnMovieEndLoop">SwfOnMovieEndLoop</option>
<option value="SwfOnMovieEndUnload">SwfOnMovieEndUnload</option>

```

```

        <option value="SwfOnMovieEndLoadMovie">SwfOnMovieEndLoadMovie</option>
    </select>
</td>
</tr>

<tr>
    <th>FE2_SWF_ON_END_URL</th>
    <td><input type='text' name='muxer:setParamAsStr:FE2_SWF_ON_END_URL'></td>
</tr>
</table>
<!-- END - SWF muxer parameters -->

<!-- WEBM muxer parameters -->
<table id='webmtable' class='hidden'>
    <tr>
        <th>(FE2_MUXER_WEBM defines no parameters)</th>
    </tr>
</table>
<!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM##### -->
<hr>
<p>
    <input type="submit" value="Start Encode">
    <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>
HTML_END;

```

6.15.2 process_sample.php

```

#!/php
<?php
##=====
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##-----
##
## File:          $Workfile$
##               $Revision$
##
## Last Update:  $DateUTC$
##
##-----
##

#process_sample.php
# Receive a form via post from flix2_sample.php, treating each name=value pair
# as a function/param pair.
# These map to the Flix Engine COM API and each function that has a valid
# param will be called.
# Once setup is complete, calls encode() to produce an output file located in out
#   dir

function print_encoder_status()
{
    global $flfx, $version;

    print "<p>Encoder Status<br>";
}

```

```

$res= $flix->getEncoderState();
print "&nbsp;flix->getEncoderState: $res<br>";

if($version[0] == 4) {
    $flixerr= $flix->flixerrno();
    $syserr= $flix->syserrno();
} else {
    ## PHP5 can automatically detect byref parameters
    $flixerr = $syserr = 0;
    $flix->errno_($flixerr,$syserr);
}
printf("&nbsp;flix->errno_: flixerrno:%d syserrno:%d</p>", $flixerr,$syserr);
}

function process_hr($funcname, $sc)
{
    global $version, $flix, $hr;
    print "<td align=\"center\">$sc</td>";

    if ($hr || $sc) {
        if($version[0] == 4) {
            $flixerr= $flix->flixerrno();
            $syserr= $flix->syserrno();
        } else {
            ## PHP5 can automatically detect byref parameters
            $flixerr = $syserr = 0;
            $flix->errno_($flixerr,$syserr);
        }
        $esc= $flix->sc();

        printf("<td>hr:0x%8x $esc flix->errno_ flixerrno:%d syserrno:%d</td>",
            $hr,$flixerr,$syserr);
        die("Error calling $funcname, sc= $sc hr= $hr;".
            "esc= $esc flix->errno_($flixerr, $syserr)\n");
    }
    print "</tr>";
}

function parse_params()
{
    global $outdir;

    echo "<table border='1' cellpadding='5'>".
        "<caption>Flix Function Calls</caption>\n".
        "<tr><th>Function Name</th><th>Return Value</th></tr>\n";
    flush_output();

    foreach($_POST as $name=>$value) {
        if(strlen($value)) {
            #print "name= $name -> $value<br>";
            if(!strcmp($name,"setOutputFile")) {
                SimpleSet($name,$outdir.$value);
            } else if(preg_match('/^(?:a|v)codec:/', $name)) {
                init_codec($value);
            } else if(preg_match('/(?:a|v)filter:/', $value)) {
                init_filter($name);
            } else if(preg_match('/^muxer:/', $name)) {
                init_muxer($value);
            } else {
                $temp= explode(":", $name);
                if(!strcmp($temp[0], "codec")) {
                    codec_interface($temp[1], $temp[2], $value);
                } else if(!strcmp($temp[0], "filter")) {
                    filter_interface($temp[1], $temp[2], $value);
                } else if(!strcmp($temp[0], "muxer")) {
                    muxer_interface($temp[1], $temp[2], $value);
                }
            }
        }
    }
}

```

```

        } else {
            SimpleSet($name,$value);
        }
    }
}

}

function flush_output()
{
    flush(); if(ini_get("output_buffering")) ob_flush();
}

$process_sample= 1;
$version      = explode('.',phpversion());
$prefix       = "C:\\Inetpub\\";
$outdir       = $prefix."flxmedia\\out\\";

echo '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/htm
14/strict.dtd">';
echo "\r\n\r\n<html><head><title>Flix CGI Process Sample - PHP</title>\n".
    "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\"
    >\n".
    "</head><body>";
echo "<hr><p>process_sample.php version 1.7 (PHP v".phpversion().")<br>\n";
flush_output();

## report all errors
error_reporting(E_ALL);

##
## Manually load the Flix Engine type library to allow PHP to register
## constants contained within it. PHP4 seems to ignore com.autoregister_typelib.
## In addition PHP 5.2.0 will cause an exception on exit with autoregister.
## 5.0|1.x seem to show the opposite behavior.
##
if($version[0]<5 || ($version[0]==5 && $version[1]>1)) {
    ini_set("com.autoregister_typelib","0");
    com_load_typelib("On2.FlixEngine") or
        die("Error loading Flix Engine type library");
} else {
    ini_set("com.autoregister_typelib","1");
}

## allow encodes of long files to complete, barring any further
## server side timeout settings,
## e.g. Apache: Timeout Directive,
##      IIS5.x: Web Sites properties -> Home Dir. -> Conf. ->
##              Process Options -> CGI script timeout
##      IIS6.x: Modify the CGITimeout Metabase property using MetaEdit
ini_set("max_execution_time","-1");
ini_set("max_input_time","-1");

if($version[0] == 4) {
    include 'process_sample.php4';
} else {
    include 'process_sample.php5';
}

load_engine();

##verify outdir's presence and accessibility
if(!(is_dir($outdir) && is_writeable($outdir))) {
    $t=array("<i>'$outdir'</i>","<tt>$outdir</tt>"); $file=__FILE__;
    echo "<p>*****<br>\n".
        "<b>WARNING</b>: $t[0] MUST exist and be writeable by".
        " <i>flxengine_com.dll</i>.\n".

```

```

        "<br>Please make $t[0] accessible or modify the $t[1]".
        " value defined in '$file'.<br>\n";
$g=`find /N 'my \${outdir}' '$file'`;
if($g) {
    print "The definition of $t[1] can be found here:<br>\n".
        str_replace("\n", "<br>", $g). "<br>\n";
}
echo "*****</p>\n";
flush_output(); $hr=0;
}

parse_params(); flush_output();
encode();
print "</body>\n</html>";

# vim:expandtab
?>

```

6.15.3 process_sample.php4 - PHP4 processing

```

<?php
##=====
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##-----
##
## File:          $Workfile$
##               $Revision$
##
## Last Update:  $DateUTC$
##
##-----
##

function error_handler($errno, $errstr, $errfile, $errline, $errcontext)
{
    global $hr;

    switch($errno) {
    case E_ERROR:
    case E_USER_ERROR:
        $str= "Error:";
        break;
    case E_WARNING:
    case E_USER_WARNING:
        $str= "Warning:";
        break;
    case E_USER_NOTICE:
        $str= "Notice:";
        break;
    default:
        $str= "Unknown:";
        break;
    }
    $str .= " [errno: $errno] errstr: $errstr<br>".
        " file: $errfile line: $errline<br>";
    ##set a false hr, we'll allow process_hr to print any further
    ##information and exit
    print $str;
    $hr= -1;
}

function load_engine()
{
    global $flx;

```



```

##
## retrieve the main engine interface, IFlix
##
$flix= new COM("On2.FlixEngine") or die("Error loading Flix Engine COM!");

echo "Flix Engine COM library. Flix Engine v".$flix->version().
    " COM v".$flix->com_version()."<br>";
echo str_replace("\n", "<br>", $flix->copyright())."</p>";
}

function SimpleSet($func, $arg)
{
    global $flix;

    echo "<tr><td>flix->$func( $arg )</td>";
    if(!strcmp($func, "setInputFile") || !strcmp($func, "setOutputFile")) {
        $flix->$func($arg);
    } else {
        $flix->$func(constant("$arg"));
    }
    process_hr("flix->$func", $flix->sc);
}

function init_codec($name)
{
    global $flix, $codec;

    ##if name is a codec name, e.g. FE2_CODEC_VP6, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this codec until we
    ##hit the next codec name
    print "<tr><td>flix->addCodec( $name )</td>";
    $codec= $flix->addCodec($flix->$name);
    process_hr('flix->addCodec', $flix->sc);
}

function codec_interface($funcname, $name, $value)
{
    global $flix, $codec;
    print "<tr><td>codec->$funcname( $name, $value )</td>";

    $codec->$funcname($flix->$name,
        (preg_match('/AsStr/', $funcname) || preg_match('/[[:alpha:]]_/', $value))?$
        value:constant("$value"));
    process_hr("codec->$funcname", $codec->sc);
}

function init_filter($name)
{
    global $flix, $filter;

    ##if name is a filter name, e.g. FE2_FILTER_CUT, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this filter until we
    ##hit the next filter name
    print "<tr><td>flix->addFilter( $name )</td>";
    $filter= $flix->addFilter($flix->$name);
    process_hr('flix->addFilter', $flix->sc);
}

function filter_interface($funcname, $name, $value)
{
    global $flix, $filter;
    print "<tr><td>filter->$funcname( $name, $value )</td>";

    $filter->$funcname($flix->$name,
        (preg_match('/AsStr/', $funcname) || preg_match('/[[:alpha:]]_/', $value))?$
        value:constant("$value"));
}

```

```

        process_hr("filter->$funcname",$filter->sc);
    }

function init_muxer($name)
{
    global $flix, $muxer;

    ##if name is a muxer name, e.g. FE2_MUXER_FLV, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this muxer until we
    ##hit the next muxer name
    print "<tr><td>flix->addMuxer( $name )</td>";
    $muxer= $flix->addMuxer($flix->$name);
    process_hr('flix->addMuxer',$flix->sc);
}

function muxer_interface($funcname, $name, $value)
{
    global $flix, $muxer;
    print "<tr><td>muxer->$funcname( $name, $value )</td>";

    $muxer->$funcname($flix->$name,
        (preg_match('/AsStr/', $funcname) || !preg_match('/[[:alpha:]]_/', $value))?$
        value:constant("$value"));
    process_hr("muxer->$funcname",$muxer->sc);
}

function encode()
{
    global $flix;

    ##
    ## start the encode
    ##
    print "<tr><td>flix->encode()</td>";
    $flix->encode();
    process_hr('flix->encode',$flix->sc);

    ##
    ## retrieve the encoding status interface, IEncodingStatus
    ##
    print "<tr><td>flix->encodingStatus()</td>";
    $encstatus= $flix->encodingStatus(); process_hr('flix->encodingStatus',0);

    print "</table>\n";

    print "<p>Encoding...(video frames encoded, percent complete). ".
        "Total frames will reset when doing 2pass.<br>";
    do {
        $ier= $flix->isEncoderRunning();
        $tf= $encstatus->totalFrames();
        $p= $encstatus->percentComplete();
        print "($tf, $p%)<br>"; flush_output();
        sleep(1);
    } while($ier);
    print "<br>Done!</p>";
    print_encoder_status();
}

if(!$process_sample) {
    die("This script is meant to be included by process_sample.php not".
        " run directly.<br>");
}
set_error_handler('error_handler');

# vim:expandtab
?>

```

6.15.4 process_sample.php5 - PHP5 processing

```
<?php
##=====
##
## Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##-----
##
## File:          $Workfile$
##               $Revision$
##
## Last Update:  $DateUTC$
##
##-----
##

function handle_com_exception($e)
{
    global $hr;

    print_str_replace("\n", "<br>", $e->__toString())."<br>";
    $hr= $e->getCode();
}

function load_engine()
{
    global $flix;

    try {
        ##
        ## retrieve the main engine interface, IFlix
        ##
        $flix= new COM("On2.FlixEngine");
    } catch(com_exception $e) {
        handle_com_exception($e);
        die("Error loading Flix Engine COM!");
    }

    try {
        echo "Flix Engine COM library. Flix Engine v".$flix->version().
            " COM v".$flix->com_version()."<br>";
        echo str_replace("\n", "<br>", $flix->copyright())."</p>";
    } catch(com_exception $e) {
        handle_com_exception($e);
    }
}

function SimpleSet($func, $arg)
{
    global $flix;

    echo "<tr><td>flix->$func( $arg )</td>";
    try {
        if(!strcmp($func, "setInputFile")||!strcmp($func, "setOutputFile")) {
            $flix->$func($arg);
        } else {
            $flix->$func(constant("$arg"));
        }
    } catch(com_exception $e) {
        handle_com_exception($e);
    }
    process_hr("flix->$func", $flix->sc);
}

function init_codec($name)
{

```

```

global $flix, $codec;

##if name is a codec name, e.g. FE2_CODEC_VP6, add an instance
##we'll assume all IFlixPlgn::setParam's relate to this codec until we
##hit the next codec name
print "<tr><td>flix->addCodec( $name )</td>";
try {
    $codec= $flix->addCodec($flix->$name);
} catch(com_exception $e) {
    handle_com_exception($e);
}
process_hr('flix->addCodec',$flix->sc);
}

function codec_interface($funcname, $name, $value)
{
    global $flix, $codec;
    print "<tr><td>codec->$funcname( $name, $value )</td>";

    try {
        $codec->$funcname($flix->$name,
            (preg_match('/AsStr/', $funcname) || !preg_match('/[[:alpha:]]/', $value)
            )?$value:constant("$value"));
    } catch(com_exception $e) {
        handle_com_exception($e);
    }
    process_hr("codec->$funcname", $codec->sc);
}

function init_filter($name)
{
    global $flix, $filter;

    ##if name is a filter name, e.g. FE2_FILTER_CUT, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this filter until we
    ##hit the next filter name
    print "<tr><td>flix->addFilter( $name )</td>";
    try {
        $filter= $flix->addFilter($flix->$name);
    } catch(com_exception $e) {
        handle_com_exception($e);
    }
    process_hr('flix->addFilter',$flix->sc);
}

function filter_interface($funcname, $name, $value)
{
    global $flix, $filter;
    print "<tr><td>filter->$funcname( $name, $value )</td>";

    try {
        $filter->$funcname($flix->$name,
            (preg_match('/AsStr/', $funcname) || !preg_match('/[[:alpha:]]/', $value)
            )?$value:constant("$value"));
    } catch(com_exception $e) {
        handle_com_exception($e);
    }
    process_hr("filter->$funcname", $filter->sc);
}

function init_muxer($name)
{
    global $flix, $muxer;

    ##if name is a muxer name, e.g. FE2_MUXER_FLV, add an instance
    ##we'll assume all IFlixPlgn::setParam's relate to this muxer until we
    ##hit the next muxer name

```

```

    print "<tr><td>flix->addMuxer( $name )</td>";
    try {
        $muxer= $flix->addMuxer($flix->$name);
    } catch(com_exception $e) {
        handle_com_exception($e);
    }
    process_hr('flix->addMuxer',$flix->sc);
}

function muxer_interface($funcname, $name, $value)
{
    global $flix, $muxer;
    print "<tr><td>muxer->$funcname( $name, $value )</td>";

    try {
        $muxer->$funcname($flix->$name,
            (preg_match('/AsStr/', $funcname) || !preg_match('/[[:alpha:]]_/', $value)
            )?$value:constant("$value"));
    } catch(com_exception $e) {
        handle_com_exception($e);
    }
    process_hr("muxer->$funcname", $muxer->sc);
}

function encode()
{
    global $flix;

    ##
    ## start the encode
    ##
    print "<tr><td>flix->encode()</td>";
    $flix->encode();
    process_hr('flix->encode',$flix->sc);

    ##
    ## retrieve the encoding status interface, IEncodingStatus
    ##
    print "<tr><td>flix->encodingStatus()</td>";
    $encstatus= $flix->encodingStatus(); process_hr('flix->encodingStatus',0);

    print "</table>\n";

    print "<p>Encoding...(video frames encoded, percent complete). ".
        "Total frames will reset when doing 2pass.<br>";
    do {
        $ier= $flix->isEncoderRunning();
        $tf= $encstatus->totalFrames();
        $p= $encstatus->percentComplete();
        print "($tf, $p%)<br>"; flush_output();
        sleep(1);
    } while($ier);
    print "<br>Done!</p>";
    print_encoder_status();
}

if(!$process_sample) {
    die("This script is meant to be included by process_sample.php not".
        " run directly.\n");
}

# vim:expandtab
?>

```

6.16 Command Line

```

Attribute VB_Name = "cliMain"
'=====
'
' Copyright (c) On2 Technologies Inc. All Rights Reserved.
'
'-----
'
' File:          $Workfile$
'               $Revision$
'
' Last Update: $DateUTC$
'
'-----
Option Explicit

Dim flix As flixengine_com.flix
Dim encodingStatus As flixengine_com.IEncodingStatus

'Objects used to gain access to the Console StdOut
Dim FSO As New Scripting.FileSystemObject
Dim stdout As Scripting.TextStream

'Win32 Console functions
Private Declare Function AttachConsole Lib "kernel32" (ByVal pid As Long) As Boolean
Private Declare Function AllocConsole Lib "kernel32" () As Boolean
Private Declare Function FreeConsole Lib "kernel32" () As Boolean
Private Const ATTACH_PARENT_PROCESS As Long = -1
Dim bNewConsole As Boolean

'Use the Win32 Sleep function for progress updates.
'A Timer object would be used in a forms-based project.
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

'These file paths are altered in the ParseCommandLine function and used in Main
Dim inputFile As String, outputFile As String

Sub Main()
    InitConsole
    Set flix = New flixengine_com.flix

    'currentFlixProcedure is used here as a convenience to keep track of which
    'Flix Engine procedure has generated an error, since VB6 has no standalone
    'executable stack trace. Set before each Flix-related call.
    Dim currentFlixProcedure As String
    On Error GoTo ErrorHandler

    'enable logging, 0=none(disable) 1=info 2=error(asserts) 3=debug 4=heavy
    'CONOUT$ can be used as the log file name to send output to the console
    flix.setLogLevel (3)
    flix.setLogPath ("cli_encode_vb6.log")

    'Print some library information
    currentFlixProcedure = "flix.version"
    stdout.WriteLine ("Flix Engine COM library. Flix Engine v" & flix.version _
        & " COM v" & flix.com_version)
    currentFlixProcedure = "flix.copyright"
    stdout.WriteLine (flix.copyright & vbCrLf)

    If (ParseCommandLine = False) Then
        stdout.WriteLine ("usage: cli_encode_vb6.exe <infile> <outfile>")
        stdout.WriteLine ("")
        ReleaseConsole
        Set flix = Nothing
    
```

```

Exit Sub
End If

' Set the source file
stdioOut.WriteLine ("Input file : " & inputFile)
currentFlixProcedure = "flix.setInputFile"
flix.setInputFile (inputFile)

' Retrieve the video options interface, IVideoOptions, and display input
' file properties
Dim on2VidOpts As flixengine_com.IVideoOptions
currentFlixProcedure = "flix.videoOptions"
Set on2VidOpts = flix.videoOptions
currentFlixProcedure = "on2VidOpts.getSourceWidth"
stdioOut.WriteLine (String(14, " ") & "Width: " _
    & on2VidOpts.getSourceWidth)
currentFlixProcedure = "on2VidOpts.getSourceHeight"
stdioOut.WriteLine (String(14, " ") & "Height: " _
    & on2VidOpts.getSourceHeight)
currentFlixProcedure = "flix.getSourceDuration"
stdioOut.WriteLine (String(14, " ") & "Duration: " _
    & flix.getSourceDuration & "ms")
' Release the IVideoOption object, as we don't need it any more.
Set on2VidOpts = Nothing

' Set the destination file
stdioOut.WriteLine ("Output file : " & outputFile)
currentFlixProcedure = "flix.setOutputFile"
flix.setOutputFile (outputFile)

' Options may be set and codecs/filters/muxers may be added prior to encode

' Add the scale filter
' Dim scaleFilter As flixengine_com.IFlixPlgn
' currentFlixProcedure = "flix.addFilter"
' Set scaleFilter = flix.addFilter(flix.FE2_FILTER_SCALE)
' currentFlixProcedure = "scaleFilter.setParam"
' Call scaleFilter.setParam(flix.FE2_SCALE_WIDTH, 240)
' currentFlixProcedure = "scaleFilter.setParam"
' Call scaleFilter.setParam(flix.FE2_SCALE_HEIGHT, 160)
' Set scaleFilter = Nothing

' Add the VP6 codec. Though it is the default, you must add it in order to
' modify its settings.
' Dim vp6Codec As flixengine_com.IFlixPlgn
' currentFlixProcedure = "flix.addCodec"
' Set vp6Codec = flix.addCodec(flix.FE2_CODEC_VP6)
' currentFlixProcedure = "vp6Codec.setParam"
' Call vp6Codec.setParam(flix.FE2_VP6_RC_MODE, VBR_1PASSControl)
' Set vp6Codec = Nothing

' Use the FLV muxer (default)
' Dim muxer As flixengine_com.IFlixPlgn
' currentFlixProcedure = "flix.addMuxer"
' Set muxer = flix.addMuxer(flix.FE2_MUXER_FLV)
' Set muxer = Nothing

' start the encode
currentFlixProcedure = "flix.encode"
flix.encode
currentFlixProcedure = "flix.encodingStatus"
Set encodingStatus = flix.encodingStatus

' Start the progress updates
Dim isRunning As Long, percentString As String * 3
stdioOut.WriteBlankLines (1)
Do

```

```

        Sleep (500)
        currentFlixProcedure = "flix.isEncoderRunning"
        isRunning = flix.isEncoderRunning
        currentFlixProcedure = "on2EncodingStatus.percentComplete"
        RSet percentString = encodingStatus.percentComplete
        stdout.Write (vbCr & "Encoding..." & percentString & "% ")
    Loop Until (isRunning = 0)

    stdout.WriteLine ("Done!")
EndSummary

Exit Sub
ErrorHandler:
    Dim errorValue As Long
    errorValue = Err.Number
    Call HandleError(errorValue, currentFlixProcedure)
End Sub

'Activates and tests StdOut to the console
Sub InitConsole()
    bNewConsole = False
    If Not AttachConsole(ATTACH_PARENT_PROCESS) Then
        bNewConsole = AllocConsole()
    End If

    On Error GoTo IOError
    Set stdout = FSO.GetStandardStream(StdOut)

    stdout.Write (vbCr) 'Should throw an error if stdout is invalid
Exit Sub
IOError:
    MsgBox ("Error: cli_encode_vb6.exe was unable to attach to Console Standard O
        utput")
    End
End Sub

Sub ReleaseConsole()
    'Let the user know the program is exiting; technically not necessary if we
    'used the parent console, but we've overwritten the prompt, so having the
    'user press <Enter> generates a new prompt
    stdout.Write (vbLf & "Press any key to continue . . . ")

    If bNewConsole Then
        Dim temp As String
        Dim stdin As Scripting.TextStream
        Set stdin = FSO.GetStandardStream(StdIn)
        temp = stdin.Read(1)
    End If
    FreeConsole
End Sub

Private Function ParseCommandLine() As Boolean
    'Position in the command string which divides the infile and outfile paths
    Dim dividePosition As String, commandLine As String
    commandLine = Trim(Command)

    If (Len(commandLine) < 3) Then
        ParseCommandLine = False
        Exit Function
    End If

    'Check for long file paths with quotes
    If (InStr(1, commandLine, "\"") > 0) Then
        If (InStr(1, commandLine, "\"") = 1) Then 'infile is quoted
            dividePosition = InStr(2, commandLine, "\"") + 1
        Else 'infile is not quoted, but outfile is quoted
            dividePosition = InStr(1, commandLine, "\"") - 1
        End If
    End If
End Function

```



```

        End If
    Else 'Neither file path is quoted
        dividePosition = InStr(1, commandLine, " ")
    End If

    If (dividePosition > 0) Then
        inputFile = Trim$(Left$(commandLine, dividePosition - 1))
        outputFile = Trim$(Right$(commandLine, Len(commandLine) - dividePosition)
    )
    End If

    If (InStr(1, inputFile, "\"") > 0) Then
        inputFile = Mid$(inputFile, 2, Len(inputFile) - 2)
    End If

    If (InStr(1, outputFile, "\"") > 0) Then
        outputFile = Mid$(outputFile, 2, Len(outputFile) - 2)
    End If

    If ((Len(inputFile) > 0) And (Len(outputFile) > 0)) Then
        ParseCommandLine = True
    Else
        ParseCommandLine = False
    End If
End Function

Private Sub HandleError(errorNumber As Long, currentFlixProcedure As String)
    'Get current state of Flix Engine
    Dim flixerr As FE2_errno
    flixerr = ErrNone
    Dim syserr As Long
    Call flix.errno_(flixerr, syserr)
    If (flixerr = ErrNone) Then
        'If the error is not a Flix Engine error,
        'pass the error on to the default VB handler
        Err.Clear
        Err.Raise (errorNumber)
    Else
        stdout.WriteLine (currentFlixProcedure & " failed!")
        stdout.WriteLine ("HRESULT (" & errorNumber & ")")
        EndSummary
    End If
End Sub

'Prints a summary of Flix Engine return values and explicitly cleans references.
Private Sub EndSummary()
    'Print encoder status
    stdout.WriteLine (1)
    stdout.WriteLine ("Encoder Status")
    stdout.WriteLine (" flix.getEncoderState: " & flix.getEncoderState)

    Dim flixerr As FE2_errno
    flixerr = ErrNone
    Dim syserr As Long
    Call flix.errno_(flixerr, syserr)
    stdout.WriteLine (" flix.errno_: hr:" & Err.Number & " flixerrno:" _
        & flixerr & " syserrno:" & syserr)
    ReleaseConsole

    'Clean up
    If (flix.isEncoderRunning = 1) Then
        flix.stopEncoding
    End If
    Set encodingStatus = Nothing
    Set flix = Nothing

    End 'Exit program

```

End Sub

6.17 Visual Basic .NET

The Visual Basic .NET example requires that the Microsoft .NET Framework (1.1|2.0) be installed. A pre-built executable is included with the install and can be found under `<InstallDir>\sample\vb.net\bin`. Should you receive an error dialog containing the text "The application failed to initialize ...", it is likely that you do not have the Framework installed. This component is located on Windows Update under "Software, Optional".

6.18 Command Line

```
'=====
'
' Copyright (c) On2 Technologies Inc. All Rights Reserved.
'
'-----
'
' File:          $Workfile$
'               $Revision$
'
' Last Update:  $DateUTC$
'
'-----
Imports System
Imports System.Runtime.InteropServices 'COMException
Imports flixengine_com 'Flix Engine interfaces
Module cli_encode

    Sub PrintStackTrace(ByVal flix As Flix, ByVal e As COMException)
        Console.WriteLine(e)
        If Not flix Is Nothing
            PrintEncoderStatus(flix)
        End If
        Environment.Exit(e.ErrorCode)
    End Sub

    Sub Main()
        'Create an instance of the main engine interface, IFlix
        Dim flix
        Try
            flix = New Flix
        Catch e As COMException
            PrintStackTrace(Nothing,e)
        End Try

        'enable logging, 0=none(disable), 1=info, 2=error(asserts), 3=debug, 4=he
avy
        Try
            flix.setLogLevel(3)
            flix.setLogPath("\cli_encode_vbnet.log")
        Catch e As COMException
            PrintStackTrace(flix,e)
        End Try

        'Print some library information
        Try
            Console.WriteLine("Flix Engine Com Library. Flix Engine v" & _
                               flix.version & " COM v" & flix.com_version)
            Console.WriteLine(flix.copyright)
        End Try
    End Sub
End Module
```

```

        Console.WriteLine()
    Catch e As COMException
        PrintStackTrace(flix,e)
    End Try

    If (System.Environment.GetCommandLineArgs.GetLength(0) < 3) Then
        Console.Error.WriteLine("usage: cli_encode_vbnet.exe <infile> <outfil
e>")
        Environment.Exit(1)
    End If

    'Set the source file
    Try
        Console.WriteLine("Input File : " & _
            System.Environment.GetCommandLineArgs(1))
        flix.setInputFile(System.Environment.GetCommandLineArgs(1))
    Catch e As COMException
        PrintStackTrace(flix,e)
    End Try

    'Retrieve the video options interface, IVideoOptions
    Try
        Dim vidopts As IVideoOptions
        vidopts = flix.videoOptions
        'Print input file information
        Console.WriteLine("Width:      ".PadLeft(24) & vidopts.getSourceWidth)

        Console.WriteLine("Height:    ".PadLeft(24) & vidopts.getSourceHeight
)
        Console.WriteLine("Duration:  ".PadLeft(24) & flix.getSourceDuration
& "ms")
    Catch e As COMException
        PrintStackTrace(flix,e)
    End Try

    'Set the destination file
    Try
        Console.WriteLine("Output File : " & _
            System.Environment.GetCommandLineArgs(2))
        flix.setOutputFile(System.Environment.GetCommandLineArgs(2))
    Catch e As COMException
        PrintStackTrace(flix,e)
    End Try

    'Options may be set and codecs/filters/muxers may be added prior to encod
e

    'Add the scale filter
    'Try
    '    Dim scaleFilter As IFlixPlgn
    '    scaleFilter = flix.addFilter(flix.FE2_FILTER_SCALE)
    '    scaleFilter.setParam(flix.FE2_SCALE_WIDTH, 240)
    '    scaleFilter.setParam(flix.FE2_SCALE_HEIGHT, 160)
    'Catch e As COMException
    '    PrintStackTrace(flix,e)
    'End Try

    'Add the vp6 codec. Though it is the default,
    'you must add it in order to modify its settings.
    'Try
    '    Dim vp6Codec As IFlixPlgn
    '    vp6Codec = flix.addCodec(flix.FE2_CODEC_VP6)
    '    vp6Codec.setParam(flix.FE2_VP6_RC_MODE, _
    '        flixengine_com.FE2_VideoBitrateControls.VBR_1PASSControl)
    'Catch e As COMException
    '    PrintStackTrace(flix,e)
    'End Try

```

```

'Use the FLV muxer (default)
'Try
'    Dim muxer As IFlixPlgn
'    muxer = flix.addMuxer(flix.FE2_MUXER_FLV)
'Catch e As COMException
'    PrintStackTrace(flix,e)
'End Try

'Start the encode
Try
    flix.encode()

    'Retrieve the encoding status interface, IEncodingStatus
    Dim encStatus As IEncodingStatus
    encStatus = flix.encodingStatus

    Console.WriteLine()
    Dim isRunning As Integer
    Do
        System.Threading.Thread.Sleep(500)
        isRunning = flix.isEncoderRunning
        Console.Write(vbCr & "Encoding..." & _
            encStatus.percentComplete.ToString.PadLeft(3) & "%
    ")
        Loop Until (isRunning = 0)
    Catch e As COMException
        PrintStackTrace(flix,e)
    End Try

    Console.WriteLine("Done!")
    PrintEncoderStatus(flix)

    'Force the cleanup of IFlix.
    'Though this is not strictly necessary in this sample, as
    'it is about to exit, if the script is more involved it may be
    'necessary so the input file can be moved as destruction of
    'the underlying FLIX2HANDLE occurs within IFlix's destructor.
    'By explicitly defining a WeakReference and removing the strong
    'reference to the flix object we are guaranteeing that it will
    'not survive the garbage collection.
    Dim wkref = New WeakReference(flix)
    flix = Nothing
    System.GC.Collect(System.GC.GetGeneration(wkref))
    System.GC.WaitForPendingFinalizers()
End Sub

Sub PrintEncoderStatus(ByVal flix As IFlix)
    Console.WriteLine()
    Console.WriteLine("Encoder Status")
    Try
        Console.WriteLine(" flix.getEncoderState: " & flix.getEncoderState)
        Dim flixerr As FE2_errno
        Dim syserr As Integer
        flix.errno_(flixerr, syserr)
        Console.WriteLine(" flix.errno_: flixerrno:" & _
            flixerr & " syserrno:" & syserr)
    Catch e As COMException
        PrintStackTrace(Nothing,e)
    End Try
End Sub

End Module

```

Chapter 7

Deprecated List

Global [audio_options_GetBitrate](#) Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

Global [audio_options_GetFlvAudioFormat](#) Use the [Codec Interface](#). This function will be removed in a future release. See also: [Audio Codecs](#).

Global [audio_options_GetSamplingrate](#) Use the [Filter Interface](#) along with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

Global [audio_options_GetStereo](#) Use the [Codec Interface](#) along with the [FE2_LAME_CHANNELS](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

Global [audio_options_Reset](#) Use the [Codec](#) and [Filter](#) interfaces.

Global [audio_options_SetBitrate](#) Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

Global [audio_options_SetFlvAudioFormat](#) Use the [Codec Interface](#). This function will be removed in a future release. See also: [Audio Codecs](#).

Global [audio_options_SetSamplingrate](#) Use the [Filter Interface](#) along with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

Global [audio_options_SetStereo](#) Use the [Codec Interface](#) along with the [FE2_LAME_CHANNELS](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

Global [audio_options_Validate](#) Use the [Codec](#) and [Filter](#) interfaces.

Global [editor_options_GetBrightness](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

Global [editor_options_GetContrast](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

Global [editor_options_GetCrop](#) Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#). This function will be removed in a future release.

Global [editor_options_GetCropBounds](#) Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#) and the [FE2_CROP_TOP](#), [FE2_CROP_LEFT](#), [FE2_CROP_BOTTOM](#) and [FE2_CROP_RIGHT](#) parameters. This function will be removed in a future release.

Global `editor_options_GetCutStartTime` Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

Global `editor_options_GetCutStopTime` Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

Global `editor_options_GetHue` Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

Global `editor_options_GetSaturation` Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

Global `editor_options_GetUseBrightness` Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

Global `editor_options_GetUseContrast` Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

Global `editor_options_GetUseCut` Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flx engine cut filter.

Global [editor_options_GetUseHue](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

Global [editor_options_GetUseSaturation](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

Global [editor_options_Reset](#) Use the [Filter Interface](#)

Global [editor_options_SetBrightness](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

Global [editor_options_SetContrast](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

Global [editor_options_SetCrop](#) Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#). This function will be removed in a future release.

Global [editor_options_SetCropBounds](#) Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#) and the [FE2_CROP_TOP](#), [FE2_CROP_LEFT](#), [FE2_CROP_BOTTOM](#) and [FE2_CROP_RIGHT](#) parameters. This function will be removed in a future release.

Global [editor_options_SetCutStartTime](#) Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flx engine cut filter.

Global [editor_options_SetCutStopTime](#) Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)

- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

Global [editor_options_SetHue](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

Global [editor_options_SetSaturation](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

Global [editor_options_SetUseBrightness](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

Global [editor_options_SetUseContrast](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

Global [editor_options_SetUseCut](#) Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

Global [editor_options_SetUseHue](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

Global [editor_options_SetUseSaturation](#) Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

Global [editor_options_Validate](#) Use the [Filter Interface](#)

Global [FE2_AudioSamplingrates](#) With the addition of new [Muxers](#) the allowed sample rates will change. For this reason the desired value should be entered directly in [Flix2_FilterSetParam\(\)](#).

Global [FE2_CuePointType](#) Use the [Muxer Interface](#) along with [FE2_MUXER_FLV](#) and the [FE2_FLV_CUEPT_EVENT/FE2_FLV_CUEPT_NAV](#) parameter. This enumeration will be removed in a future release.

Global [FE2_ExportedVideoType](#) Use the [Muxer Interface](#). This enumeration will be removed in a future release. See also: [Muxers](#).

Global [FE2_FlvAudioFormat](#) Use the [Codec Interface](#). This enumeration will be removed in a future release. See also: [Audio Codecs](#).

Global [FE2_LAME_CHANNELS](#) Use the [Filter Interface](#) along with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

Global [FE2_VideoCodec](#) Use the [Codec Interface](#). This enumeration will be removed in a future release. See also: [Video Codecs](#).

Global [Flix2_GetExportVideoType](#) Use the [Muxer Interface](#). This function will be removed in a future release. See also: [Muxers](#).

Global [Flix2_SetExportVideoType](#) Use the [Muxer Interface](#). This function will be removed in a future release. See also: [Muxers](#).

Class [flixengine_com::ISwfOptions](#) Use the [flixengine_com::IFlixPlgn](#) interface along with the [SWF](#) muxer. This interface will be removed in a future release.

Global [flixengine_com::ISwfOptions::addVariable\(const BSTR name, const BSTR value\)](#) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADD_VARIABLE](#) parameter.

Global [flixengine_com::ISwfOptions::deleteVariable\(const on2s32 index\)](#) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_DELETE_VARIABLE](#) parameter.

Global [flixengine_com::ISwfOptions::getAdaptivePreloaderBufferFactor\(\[out, retval\] double *pPreloaderBufferFactor\)](#) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter.

Global [flixengine_com::ISwfOptions::getEmbeddedUrlType\(\[out, retval\] FE2_EmbeddedUrlType *embeddedUrlType\)](#) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TYPE](#) parameter.

Global `flixengine_com::ISwfOptions::getEnablePreloader`([out, retval] on2bool *pEnablePreloader)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_PRELOAD_TYPE` parameter.

Global `flixengine_com::ISwfOptions::getInsertBlankFrameOnStart`([out, retval] on2bool *pInsertBlankFrameOnStart)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_START_BLANK_FRAME` parameter.

Global `flixengine_com::ISwfOptions::getLoopCount`([out, retval] on2s32 *pLoopCount) Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_LOOP_COUNT` parameter.

Global `flixengine_com::ISwfOptions::getMovieOnEndOptions`([out, retval] FE2_SwfOnEndOptions *pOnEndOptions)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_ON_END_OPTION` parameter.

Global `flixengine_com::ISwfOptions::getMovieOnStartOptions`([out, retval] FE2_SwfOnStartOptions *pOnStartOptions)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_ON_START_OPTION` parameter.

Global `flixengine_com::ISwfOptions::getPercentToPreload`([out, retval] on2s32 *pPercentToPreload)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_FIXED_PRELOAD_PCT` parameter.

Global `flixengine_com::ISwfOptions::getPreloaderType`([out, retval] FE2_SwfPreloaderOptions *pPreloaderType)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_PRELOAD_TYPE` parameter.

Global `flixengine_com::ISwfOptions::getSwfFramerateAsDouble`([out, retval] double *pSwfFramerate)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_FRAMERATE` parameter.

Global `flixengine_com::ISwfOptions::getVariableCount`([out, retval] on2s32 *pVariableCount)
Use the `SWF` muxer.

Global `flixengine_com::ISwfOptions::getWaitTimeToStart`([out, retval] on2s32 *pWaitTimeToStart)
Use the `flixengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_START_WAIT_SEC` parameter.

Global `flixengine_com::ISwfOptions::reset`() Use the `flixengine_com::IFlixPlgn` interface along with the `SWF` muxer. This interface will be removed in a future release.

Global `flixengine_com::ISwfOptions::sc` Use the `flixengine_com::IFlixPlgn` interface.

Global **flixengine_com::ISwfOptions::setAdaptivePreloaderBufferFactor**(const double preloaderBufferFactor)

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter.

Global **flixengine_com::ISwfOptions::setEmbeddedUrl**(const BSTR embeddedUrl) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-EMBEDDED_URL](#) parameter.

Global **flixengine_com::ISwfOptions::setEmbeddedUrlTarget**(const BSTR embeddedUrlTarget) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-EMBEDDED_URL](#) parameter.

Global **flixengine_com::ISwfOptions::setEmbeddedUrlType**(const FE2_EmbeddedUrlType embeddedUrlType) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-EMBEDDED_URL_TYPE](#) parameter.

Global **flixengine_com::ISwfOptions::setEnablePreloader**(const on2bool enablePreloader) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-PRELOAD_TYPE](#) parameter.

Global **flixengine_com::ISwfOptions::setInsertBlankFrameOnStart**(const on2bool insertBlankFrameOnStart) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-START_BLANK_FRAME](#) parameter.

Global **flixengine_com::ISwfOptions::setLoadMovieOnEndUrl**(const BSTR loadMovieOnEndUrl) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-ON_END_URL](#) parameter.

Global **flixengine_com::ISwfOptions::setLoopCount**(const on2s32 loopCount) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_LOOP_COUNT](#) parameter.

Global **flixengine_com::ISwfOptions::setMovieOnEndOptions**(const FE2_SwfOnEndOptions onEndOptions) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-ON_END_OPTION](#) parameter.

Global **flixengine_com::ISwfOptions::setMovieOnStartOptions**(const FE2_SwfOnStartOptions onStartOptions) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-ON_START_OPTION](#) parameter.

Global **flixengine_com::ISwfOptions::setPercentToPreload**(const on2s32 percentToPreload) Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-FIXED_PRELOAD_PCT](#) parameter.

Global [flixengine_com::ISwfOptions::setPreloaderType](#)(const FE2_SwfPreloaderOptions preloaderType)

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_PRELOAD_TYPE](#) parameter.

Global [flixengine_com::ISwfOptions::setSwfFramerateAsDouble](#)(const double swfFramerate)

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FRAMERATE](#) parameter.

Global [flixengine_com::ISwfOptions::setWaitTimeToStart](#)(const on2s32 waitTimeToStart)

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_WAIT_SEC](#) parameter.

Global [flixengine_com::ISwfOptions::updateVariable](#)(const on2s32 index, const BSTR name, const BSTR value)

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADD_VARIABLE](#) parameter.

Global [flixengine_com::IVideoOptions::addFLVCuePoint](#)(const BSTR pName, const double time, const FE2_CuePoint)

See [video_options_AddFLVCuePoint\(\)](#) for details.

Global [flixengine_com::IVideoOptions::addFLVCuePointParameter](#)(const BSTR pCuePointName, const BSTR pName)

See [video_options_AddFLVCuePointParameter\(\)](#) for details.

Global [flixengine_com::IVideoOptions::getDecimateValue](#)([out, retval] on2u32 *pValue)

See [video_options_GetDecimateValue\(\)](#) for details.

Global [flixengine_com::IVideoOptions::getSwfHeight](#)([out, retval] on2s32 *lpSwfHeight)

See [video_options_GetSwfHeight\(\)](#) for details.

Global [flixengine_com::IVideoOptions::getSwfWidth](#)([out, retval] on2s32 *lpSwfWidth)

See [video_options_GetSwfWidth\(\)](#) for details.

Global [flixengine_com::IVideoOptions::getUseCustomSwfDimensions](#)([out, retval] on2bool *bpUseCustomSwfDimensions)

See [video_options_GetUseCustomSwfDimensions\(\)](#) for details.

Global [flixengine_com::IVideoOptions::getUseSourceFramerate](#)([out, retval] on2bool *bpUseSourceFramerate)

See [video_options_GetUseSourceFramerate\(\)](#) for details.

Global [flixengine_com::IVideoOptions::getVideoFramerateAsDouble](#)([out, retval] double *p_fps)

See [video_options_GetVideoFramerateAsDouble\(\)](#) for details.

Global [flixengine_com::IVideoOptions::setDecimateValue](#)(const on2u32 value)

See [video_options_SetDecimateValue\(\)](#) for details.

Global **flixengine_com::IVideoOptions::setSwfHeight**(const on2s32 lSwfHeight) See [video_options_SetSwfHeight\(\)](#) for details.

Global **flixengine_com::IVideoOptions::setSwfWidth**(const on2s32 lSwfWidth) See [video_options_SetSwfWidth\(\)](#) for details.

Global **flixengine_com::IVideoOptions::setUseCustomSwfDimensions**(const on2bool bUseCustomSwfDimensions) See [video_options_SetUseCustomSwfDimensions\(\)](#) for details.

Global **flixengine_com::IVideoOptions::setUseSourceFramerate**(const on2bool bUseSourceFramerate) See [video_options_SetUseSourceFramerate\(\)](#) for details.

Global **flixengine_com::IVideoOptions::setVideoFramerateAsDouble**(const double fps) See [video_options_SetVideoFramerateAsDouble\(\)](#) for details.

Global **overlay_options_GetMaskPixelRGB** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_R](#), [FE2_OVERLAY_MASK_G](#) and [FE2_OVERLAY_MASK_B](#) parameters. This function will be removed in a future release.

Global **overlay_options_GetMaskPixelXY** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_X](#) and [FE2_OVERLAY_MASK_Y](#) parameters. This function will be removed in a future release.

Global **overlay_options_GetOverlayPath** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_FILE](#) parameter. This function will be removed in a future release.

Global **overlay_options_GetOverlayPosition** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_POS](#), [FE2_OVERLAY_POS_X](#), [FE2_OVERLAY_POS_Y](#) parameters. This function will be removed in a future release.

Global **overlay_options_GetUseOverlay** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#). This function will be removed in a future release.

Global **overlay_options_Reset** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#). This function will be removed in a future release.

Global **overlay_options_SetMaskPixelRGB** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_R](#), [FE2_OVERLAY_MASK_G](#) and [FE2_OVERLAY_MASK_B](#) parameters. This function will be removed in a future release.

Global **overlay_options_SetMaskPixelXY** Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_X](#) and [FE2_OVERLAY_MASK_Y](#) parameters. This function will be removed in a future release.

Global [overlay_options_SetOverlayPath](#) Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_FILE](#) parameter. This function will be removed in a future release.

Global [overlay_options_SetOverlayPosition](#) Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_POS](#), [FE2_OVERLAY_POS_X](#), [FE2_OVERLAY_POS_Y](#) parameters. This function will be removed in a future release.

Global [overlay_options_SetUseOverlay](#) Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#). This function will be removed in a future release.

Global [swf_options_AddVariable](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADD_VARIABLE](#) parameter. This function will be removed in a future release.

Global [swf_options_DeleteVariable](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_DELETE_VARIABLE](#) parameter. This function will be removed in a future release.

Global [swf_options_GetAdaptivePreloaderBufferFactor](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter. This function will be removed in a future release.

Global [swf_options_GetEmbeddedUrl](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL](#) parameter. This function will be removed in a future release.

Global [swf_options_GetEmbeddedUrlTarget](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TARGET](#) parameter. This function will be removed in a future release.

Global [swf_options_GetEmbeddedUrlType](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TYPE](#) parameter. This function will be removed in a future release.

Global [swf_options_GetLoadMovieOnEndUrl](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_URL](#) parameter. This function will be removed in a future release.

Global [swf_options_GetLoopCount](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_LOOP_COUNT](#) parameter. This function will be removed in a future release.

Global [swf_options_GetMovieOnEndOptions](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_OPTION](#) parameter. This function will be removed in a future release.

Global [swf_options_GetMovieOnStartOptions](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_START_OPTION](#) parameter. This function will be removed in a future release.

Global [swf_options_GetPercentToPreload](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FIXED_PRELOAD_PCT](#) parameter. This function will be removed in a future release.

Global [swf_options_GetPreloaderType](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_PRELOAD_TYPE](#) parameter. This function will be removed in a future release.

Global [swf_options_GetSwfFramerate](#) This function has been deprecated in favor of [swf_options_GetSwfFramerateAsDouble\(\)](#) because this function can only handle integer framerates.

Global [swf_options_GetSwfFramerateAsDouble](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FRAMERATE](#) parameter. This function will be removed in a future release.

Global [swf_options_GetVariableCount](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#). This function will be removed in a future release.

Global [swf_options_GetWaitTimeToStart](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_WAIT_SEC](#) parameter. This function will be removed in a future release.

Global [swf_options_Reset](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#). This function will be removed in a future release.

Global [swf_options_SetAdaptivePreloaderBufferFactor](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter. This function will be removed in a future release.

Global [swf_options_SetEmbeddedUrl](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL](#) parameter. This function will be removed in a future release.

Global [swf_options_SetEmbeddedUrlTarget](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TARGET](#) parameter. This function will be removed in a future release.

Global [swf_options_SetEmbeddedUrlType](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TYPE](#) parameter. This function will be removed in a future release.

Global [swf_options_SetInsertBlankFrameOnStart](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_BLANK_FRAME](#) parameter. This function will be removed in a future release.

Global [swf_options_SetLoadMovieOnEndUrl](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_URL](#) parameter. This function will be removed in a future release.

Global [swf_options_SetLoopCount](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_LOOP_COUNT](#) parameter. This function will be removed in a future release.

Global [swf_options_SetMovieOnEndOptions](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_OPTION](#) parameter. This function will be removed in a future release.

Global [swf_options_SetMovieOnStartOptions](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_START_OPTION](#) parameter. This function will be removed in a future release.

Global [swf_options_SetPercentToPreload](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FIXED_PRELOAD_PCT](#) parameter. This function will be removed in a future release.

Global [swf_options_SetPreloaderType](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_PRELOAD_TYPE](#) parameter. This function will be removed in a future release.

Global [swf_options_SetSwfFramerate](#) This function has been deprecated in favor of [swf_options_SetSwfFramerateAsDouble\(\)](#) because this function can only handle integer framerates.

Global [swf_options_SetSwfFramerateAsDouble](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FRAMERATE](#) parameter. This function will be removed in a future release.

Global [swf_options_SetWaitTimeToStart](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_WAIT_SEC](#) parameter. This function will be removed in a future release.

Global [swf_options_UpdateVariable](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADD_VARIABLE](#) parameter. This function will be removed in a future release.

Global [video_options_AddFLVCuePoint](#) Use the [Muxer Interface](#) along with [FE2_MUXER_FLV](#) and the [FE2_FLV_CUEPT_EVENT/FE2_FLV_CUEPT_NAV](#) parameter. This function will be removed in a future release.

Global [video_options_AddFLVCuePointParameter](#) Use the [Muxer Interface](#) along with [FE2_MUXER_FLV](#) and the [FE2_FLV_CUEPT_PARAM](#) parameter. This function will be removed in a future release.

Global [video_options_GetAlphaPercentage](#) Use the [Codec Interface](#) along with [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6A_ALPHA_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetCompressMode](#) Use the [Codec Interface](#) along with [FE2_CODEC_VP6](#) or [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6_CXMODE](#) or [FE2_VP6A_CXMODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetDecimateValue](#) Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_DECIMATE](#) parameter. This function will be removed in a future release.

Global [video_options_GetDeinterlace](#) Use the [Filter Interface](#) along with [FE2_FILTER_ADAPTIVE_DEINTERLACE](#). This function will be removed in a future release.

Global [video_options_GetImageHeight](#) Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flx engine video scaling filter.

Global [video_options_GetImageQuality](#) The image quality functions have been deprecated in favor of the maximum bitrate functions. Use [video_options_GetMaximumBitrate\(\)](#) instead.

Global [video_options_GetImageWidth](#) Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flx engine video scaling filter.

Global [video_options_GetKeyframeInterval](#) Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFFREQ](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetKeyframeIntervalType](#) Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFINTTYPE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetMaximumBitrate](#) Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetRateControlType](#) Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_RC_MODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetSwfFramerate](#) Please use [swf_options_GetSwfFramerateAsDouble\(\)](#) to allow for non integer framerates.

Global [video_options_GetSwfHeight](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_HEIGHT](#) parameter. This function will be removed in a future release.

Global [video_options_GetSwfWidth](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH](#) parameter. This function will be removed in a future release.

Global [video_options_GetUseCustomSwfDimensions](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH/FE2_SWF_HEIGHT](#) parameters. This function will be removed in a future release.

Global [video_options_GetUseMaximumBitrate](#) Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetUseSourceDimensions](#) The flx engine encodes video using the video source dimensions by default. If [FE2_FILTER_SCALE](#) has not been added to the filter chain, video will be encoded using source dimensions.

Global [video_options_GetUseSourceFramerate](#) Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#). Not adding [FE2_FILTER_FRAMERATE](#) with [Flix2_AddFilter\(\)](#) implies use of the source frame rate. This function will be removed in a future release.

Global [video_options_GetVideoCodec](#) Use the [Codec Interface](#). This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_GetVideoFramerate](#) Please use [video_options_GetVideoFramerateAsDouble\(\)](#) to allow for non integer framerates.

Global [video_options_GetVideoFramerateAsDouble](#) Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_FPS](#) parameter. This function will be removed in a future release.

Global [video_options_SetAlphaPercentage](#) Use the [Codec Interface](#) along with [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6A_ALPHA_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetCompressMode](#) Use the [Codec Interface](#) along with [FE2_CODEC_VP6](#) or [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6_CXMODE](#) or [FE2_VP6A_CXMODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetDecimateValue](#) Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_DECIMATE](#) parameter. This function will be removed in a future release.

Global [video_options_SetDeinterlace](#) Use the [Filter Interface](#) along with [FE2_FILTER_ADAPTIVE_DEINTERLACE](#). This function will be removed in a future release.

Global [video_options_SetImageHeight](#) Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flx engine video scaling filter.

Global [video_options_SetImageQuality](#) The image quality functions have been deprecated in favor of the maximum bitrate functions. Use [video_options_GetMaximumBitrate\(\)](#) instead.

Global [video_options_SetImageWidth](#) Please use:

- [Flix2_AddFilter\(\)](#)

- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flx engine video scaling filter.

Global [video_options_SetKeyframeInterval](#) Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFFREQ](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetKeyframeIntervalType](#) Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFINTTYPE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetMaximumBitrate](#) Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetRateControlType](#) Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_RC_MODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetSwfFramerate](#) Please use [swf_options_SetSwfFramerateAsDouble\(\)](#) to allow for non integer framerates.

Global [video_options_SetSwfHeight](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_HEIGHT](#) parameter. This function will be removed in a future release.

Global [video_options_SetSwfWidth](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH](#) parameter. This function will be removed in a future release.

Global [video_options_SetUseCustomSwfDimensions](#) Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH](#)/[FE2_SWF_HEIGHT](#) parameters. This function will be removed in a future release.

Global [video_options_SetUseMaximumBitrate](#) Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetUseSourceDimensions](#) The flx engine encodes video using the video source dimensions by default. If [FE2_FILTER_SCALE](#) has not been added to the filter chain, video will be encoded using source dimensions.

Global [video_options_SetUseSourceFramerate](#) Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#). Not adding [FE2_FILTER_FRAMERATE](#) with [Flix2_AddFilter\(\)](#) implies use of the source frame rate. This function will be removed in a future release.

Global [video_options_SetVideoCodec](#) Use the [Codec Interface](#). This function will be removed in a future release. See also: [Video Codecs](#).

Global [video_options_SetVideoFramerate](#) Please use [video_options_SetVideoFramerateAsDouble\(\)](#) to allow for non integer framerates.

Global [video_options_SetVideoFramerateAsDouble](#) Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_FPS](#) parameter. This function will be removed in a future release.

Chapter 8

Module Index

8.1 Modules

Here is a list of all modules:

Flix Engine API	476
Audio Encoding Options	323
Deprecated	325
Codecs	332
Common Codec Parameters	333
Video Codecs	335
H263 - FFmpeg	343
H264	346
VP6	354
VP6 with Alpha	364
VP8	372
Audio Codecs	336
AAC	337
AAC+	339
AMR_NB - FFmpeg	341
MP3 - LAME	350
libvorbis - FFmpeg	353
Encoding Statistics	381
Filters	386
Video Filters	387
Deinterlace	389
Brightness/Contrast/Hue/Saturation	392
Blur	402
Crop	405
Denoise	427
Frame Rate	428
Mirror	438
Overlay (Watermark)	440
PNG Image Export (Thumbnail)	453
Rotate	462
Scale	463
Sharpen	475
Audio Filters	388

Highpass	434
Lowpass	436
Resample	461
Cut	421
Main Engine Interface	477
Filter Manipulation	491
Deprecated	503
Codec Manipulation	495
Muxer Manipulation	499
Muxers	504
FLV	505
FXM	510
3GPP - FFmpeg	513
3GPP2 - FFmpeg	515
MOV - FFmpeg	517
MP4 - FFmpeg	519
SWF	521
WebM - FFmpeg	553
Video Encoding Options	559
Deprecated	564
Base Types	554
Flix Engine COM	578

Chapter 9

Namespace Index

9.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[flixengine_com](#) 581

Chapter 10

Data Structure Index

10.1 Data Structures

Here are the data structures with brief descriptions:

flxengine_com::IEncodingStatus (Interface for accessing encoding statistics functions)	583
flxengine_com::IFlix (Main interface for accessing engine functions and obtaining instances of other COM interfaces)	586
flxengine_com::IFlixPlgn (Interface for accessing Codec , Filter and Muxer functions)	624
flxengine_com::ISwfOptions (Interface for accessing SWF creation functions)	626
flxengine_com::IVideoOptions (Interface for accessing non-deprecated video_options_* functions)	635

Chapter 11

File Index

11.1 File List

Here is a list of all files with brief descriptions:

flixengine_com.idl (Reference module(s): Flix Engine COM)	709
flixengine2/audio_options.h (Reference module(s): Audio Encoding Options)	642
flixengine2/codec_constants.h (Reference module(s): Codecs)	644
flixengine2/encoding_status.h (Reference module(s): Encoding Statistics)	659
flixengine2/filter_constants.h (Reference module(s): Filters)	660
flixengine2/flixengine2.h (Reference module(s): Main Engine Interface , Filter Manipulation , Codec Manipulation , Muxer Manipulation)	679
flixengine2/media_editor_options.h (Reference module(s): Filter Manipulation , Video Filters)	687
flixengine2/muxer_constants.h (Reference module(s): Muxers)	689
flixengine2/on2types.h (Reference module(s): Base Types)	698
flixengine2/overlay_options.h (Reference module(s): Overlay (Watermark))	701
flixengine2/swf_options.h (Reference module(s): SWF)	702
flixengine2/video_options.h (Reference module(s): Video Encoding Options , Frame Rate , Scale)	705
flixengine2/codecs/aac.h (Reference module(s): AAC , AAC+)	645
flixengine2/codecs/amr.h (Reference module(s): AMR_NB - FFmpeg)	646
flixengine2/codecs/codec_common.h	647
flixengine2/codecs/h263.h (Reference module(s): H263 - FFmpeg)	648
flixengine2/codecs/h264.h (Reference module(s): H264)	649
flixengine2/codecs/lame.h (Reference module(s): MP3 - LAME)	651
flixengine2/codecs/vorbis.h (Reference module(s): libvorbis - FFmpeg)	652
flixengine2/codecs/vp6.h (Reference module(s): VP6)	653
flixengine2/codecs/vp6_alpha.h (Reference module(s): VP6 with Alpha)	655
flixengine2/codecs/vp8.h (Reference module(s): VP8)	657
flixengine2/filters/adaptive_deinterlace.h (Reference module(s): Deinterlace)	661
flixengine2/filters/bchs.h (Reference module(s): Brightness/Contrast/Hue/Saturation)	662
flixengine2/filters/blur.h (Reference module(s): Blur)	663
flixengine2/filters/crop.h (Reference module(s): Crop)	664
flixengine2/filters/cut.h (Reference module(s): Cut)	665
flixengine2/filters/denise.h (Reference module(s): Denoise)	666
flixengine2/filters/framerate.h (Reference module(s): Frame Rate)	667
flixengine2/filters/highpass.h (Reference module(s): Highpass)	668
flixengine2/filters/lowpass.h (Reference module(s): Lowpass)	669
flixengine2/filters/mirror.h (Reference module(s): Mirror)	670

flxengine2/filters/ overlay.h (Reference module(s): Overlay (Watermark))	671
flxengine2/filters/ png_export.h (Reference module(s): PNG Image Export (Thumbnail))	673
flxengine2/filters/ resample.h (Reference module(s): Resample)	675
flxengine2/filters/ rotate.h (Reference module(s): Rotate)	676
flxengine2/filters/ scale.h (Reference module(s): Scale)	677
flxengine2/filters/ sharpen.h (Reference module(s): Sharpen)	678
flxengine2/muxers/ flv.h (Reference module(s): FLV Muxer)	690
flxengine2/muxers/ fxm.h (Reference module(s): FXM Muxer)	692
flxengine2/muxers/ isomedia.h (Reference module(s): 3GPP Muxer , 3GPP2 Muxer , MOV Muxer , MP4 Muxer)	693
flxengine2/muxers/ swf.h (Reference module(s): SWF Muxer)	695
flxengine2/muxers/ webm.h (Reference module(s): WebM Muxer)	697

Chapter 12

Module Documentation

12.1 Audio Encoding Options

Modules

- [Deprecated](#)

Enumerations

- `enum FE2_AudioBitrates {`
 [Bitrate8kbps](#),
 [Bitrate16kbps](#),
 [Bitrate24kbps](#),
 [Bitrate32kbps](#),
 [Bitrate40kbps](#),
 [Bitrate48kbps](#),
 [Bitrate56kbps](#),
 [Bitrate64kbps](#),
 [Bitrate80kbps](#),
 [Bitrate96kbps](#),
 [Bitrate112kbps](#),
 [Bitrate128kbps](#),
 [Bitrate144kbps](#),
 [Bitrate160kbps](#),
 [Bitrate192kbps](#),
 [Bitrate224kbps](#),
 [Bitrate256kbps](#),
 [Bitrate320kbps](#) }

Valid bitrates for [FE2_CODEC_LAME](#). For use in calls to [Flix2_CodecSetParam\(\)](#) and [Flix2_CodecGetParam\(\)](#).

12.1.1 Enumeration Type Documentation

12.1.1.1 enum FE2_AudioBitrates

Valid bitrates for [FE2_CODEC_LAME](#). For use in calls to [Flix2_CodecSetParam\(\)](#) and [Flix2_CodecGetParam\(\)](#).

Enumerator:

- Bitrate8kbps*
- Bitrate16kbps*
- Bitrate24kbps*
- Bitrate32kbps*
- Bitrate40kbps*
- Bitrate48kbps*
- Bitrate56kbps*
- Bitrate64kbps*
- Bitrate80kbps*
- Bitrate96kbps*
- Bitrate112kbps*
- Bitrate128kbps*
- Bitrate144kbps*
- Bitrate160kbps*
- Bitrate192kbps*
- Bitrate224kbps*
- Bitrate256kbps*
- Bitrate320kbps*

Definition at line 32 of file audio_options.h.

12.2 Deprecated

Enumerations

- enum [FE2_FlvAudioFormat](#) {
[FlvAudioUncompressed](#),
[FlvAudioMp3](#) }
Valid output audio formats for use in calls to [audio_options_GetFlvAudioFormat\(\)](#) and [audio_options_SetFlvAudioFormat\(\)](#).
- enum [FE2_AudioSamplingrates](#) {
[Hertz11025](#),
[Hertz22050](#),
[Hertz44100](#) }
Sample rates for use with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

Functions

- [on2sc audio_options_Reset](#) ([FLIX2HANDLE](#) flx)
Reset the audio encoding options to their default values.
- [on2sc audio_options_Validate](#) ([FLIX2HANDLE](#) flx)
Validate the current audio encoding options.
- [on2sc audio_options_SetBitrate](#) ([FLIX2HANDLE](#) flx, const [FE2_AudioBitrates](#) bitrate)
Set the bitrate to use in mp3 encoding.
- [on2sc audio_options_GetBitrate](#) (const [FLIX2HANDLE](#) flx, [FE2_AudioBitrates](#) *pBitrate)
Retrieve the current mp3 bitrate.
- [on2sc audio_options_SetFlvAudioFormat](#) ([FLIX2HANDLE](#) flx, const [FE2_FlvAudioFormat](#) format)
Set the output audio format.
- [on2sc audio_options_GetFlvAudioFormat](#) (const [FLIX2HANDLE](#) flx, [FE2_FlvAudioFormat](#) *pFormat)
Retrieve the current output audio format.
- [on2sc audio_options_SetSamplingrate](#) ([FLIX2HANDLE](#) flx, const [FE2_AudioSamplingrates](#) samplingrate)
Set the output audio sample rate.
- [on2sc audio_options_GetSamplingrate](#) (const [FLIX2HANDLE](#) flx, [FE2_AudioSamplingrates](#) *pSamplingrate)
Retrieve the current output sample rate.
- [on2sc audio_options_SetStereo](#) ([FLIX2HANDLE](#) flx, const [on2bool](#) stereo)

Set the number of output audio channels.

- `on2sc audio_options_GetStereo` (const [FLIX2HANDLE](#) *flx*, [on2bool](#) *pStereo)

Retrieve the current number of output audio channels.

12.2.1 Enumeration Type Documentation

12.2.1.1 enum FE2_AudioSamplingrates

Sample rates for use with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

Deprecated

With the addition of new [Muxers](#) the allowed sample rates will change. For this reason the desired value should be entered directly in [Flix2_FilterSetParam\(\)](#).

Enumerator:

Hertz11025

Hertz22050

Hertz44100

Definition at line 70 of file `audio_options.h`.

12.2.1.2 enum FE2_FlvAudioFormat

Valid output audio formats for use in calls to [audio_options_GetFlvAudioFormat\(\)](#) and [audio_options_SetFlvAudioFormat\(\)](#).

Deprecated

Use the [Codec Interface](#). This enumeration will be removed in a future release. See also: [Audio Codecs](#).

Enumerator:

FlvAudioUncompressed

FlvAudioMp3

Definition at line 59 of file `audio_options.h`.

12.2.2 Function Documentation

12.2.2.1 on2sc audio_options_GetBitrate (const [FLIX2HANDLE](#) *flx*, [FE2_AudioBitrates](#) **pBitrate*)

Retrieve the current mp3 bitrate. Only has meaning when outputting [FlvAudioMp3](#)

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pBitrate* Storage location to receive the result

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

pBitrate is not NULL

Note:

Default: [Bitrate64kbps](#)

Deprecated

Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

12.2.2.2 on2sc audio_options_GetFlvAudioFormat (const FLIX2HANDLE *flix*, FE2_FlvAudioFormat * *pFormat*)

Retrieve the current output audio format.

Parameters:

← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pFormat* Storage location to receive the result

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

pFormat is not NULL

Deprecated

Use the [Codec Interface](#). This function will be removed in a future release. See also: [Audio Codecs](#).

12.2.2.3 on2sc audio_options_GetSamplingrate (const FLIX2HANDLE *flix*, FE2_AudioSamplingrates * *pSamplingrate*)

Retrieve the current output sample rate.

Parameters:

← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pSamplingrate* Storage location to receive the result

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

pSamplingrate is not NULL

Note:

How the default output sample rate is determined is described by [audio_options_SetSamplingrate\(\)](#)

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

12.2.2.4 on2sc audio_options_GetStereo (const FLIX2HANDLE *flix*, on2bool * *pStereo*)

Retrieve the current number of output audio channels. Value indicates stereo (`on2true`) or mono (`on2false`)

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pStereo* Storage location to receive the result

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

pStereo is not NULL

Deprecated

Use the [Codec Interface](#) along with the [FE2_LAME_CHANNELS](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

12.2.2.5 on2sc audio_options_Reset (FLIX2HANDLE *flix*)

Reset the audio encoding options to their default values. Defaults are set as described by: [audio_options_SetBitrate\(\)](#), [audio_options_SetFlvAudioFormat\(\)](#), [audio_options_SetSamplingrate\(\)](#) and [audio_options_SetStereo\(\)](#)

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should the precondition fail

Precondition:

flix is not NULL

Deprecated

Use the [Codec](#) and [Filter](#) interfaces.

12.2.2.6 on2sc audio_options_SetBitrate (FLIX2HANDLE *flix*, const FE2_AudioBitrates *bitrate*)

Set the bitrate to use in mp3 encoding.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *bitrate* Bitrate to use encoding mp3 audio

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

bitrate is a valid member of [FE2_AudioBitrates](#)

Note:

Default: [Bitrate64kbps](#)

Deprecated

Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

12.2.2.7 on2sc audio_options_SetFlvAudioFormat (FLIX2HANDLE *flix*, const FE2_FlvAudioFormat *format*)

Set the output audio format.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *format* Output audio format

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

Precondition:

flix is not NULL
format is a valid member of [FE2_FlvAudioFormat](#)

Note:

Default: [FlvAudioMp3](#)

Deprecated

Use the [Codec Interface](#). This function will be removed in a future release. See also: [Audio Codecs](#).

12.2.2.8 on2sc audio_options_SetSamplingrate (FLIX2HANDLE *flix*, const FE2_AudioSamplingrates *samplingrate*)

Set the output audio sample rate.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 ← *samplingrate* Output audio sample rate

Return values:

[*ON2_OK*](#) on success
[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

Precondition:

flix is not NULL
samplingrate is a valid member of [FE2_AudioSamplingrates](#)

Remarks:

Flash video files are limited by what sample rate audio they can store. The default output sample rate is determined using the source file attributes in the following manner:

- source sample rate < 16000Hz, set default to [Hertz11025](#)
- source sample rate >= 16000Hz and < 32000Hz, set default to [Hertz22050](#)
- source sample rate >= 32000Hz, set default to [Hertz44100](#)

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

12.2.2.9 on2sc audio_options_SetStereo (FLIX2HANDLE *flix*, const on2bool *stereo*)

Set the number of output audio channels.

Parameters:

- ← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *stereo* Indicates whether to encode stereo (`on2true`) or mono (`on2false`)

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

Remarks:

Flash video files are limited by the number of output audio channels they can store. The default output number of channels is determined using the source file attributes in the following manner:

- input number of channels == 2, set default to: stereo (`on2true`)
- otherwise, set default to: mono (`on2false`)

Deprecated

Use the [Codec Interface](#) along with the [FE2_LAME_CHANNELS](#) parameter. This function will be removed in a future release. See also: [Audio Codecs](#).

12.2.2.10 on2sc audio_options_Validate (FLIX2HANDLE *flix*)

Validate the current audio encoding options. Ensures the current audio options are sufficient to start encoding. The checks currently performed are:

- sample rate is a valid member of [FE2_AudioSamplingrates](#)
- output format is a valid member of [FE2_FlvAudioFormat](#)

Parameters:

- ← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

Deprecated

Use the [Codec](#) and [Filter](#) interfaces.

12.3 Codecs

Modules

- [Common Codec Parameters](#)
- [Video Codecs](#)
- [Audio Codecs](#)

12.3.1 Detailed Description

Codecs are configurable both by the [codec interface](#) and individual functions (e.g., (audio|video)_options_SetXXX). The latter, however, are deprecated and will be removed in a future release. Therefore, any new code should use the [codec interface](#).

The currently available codecs are:

- Video Codecs
 - [H263 - FFmpeg](#)
 - [H264](#)
 - [VP6](#)
 - [VP6 with Alpha](#)
- Audio Codecs
 - [AAC](#)
 - [AAC+](#)
 - [AMR_NB - FFmpeg](#)
 - [MP3 - LAME](#)

12.4 Common Codec Parameters

Defines

- `#define FE2_CODECPARAM_BITRATE`
Codec parameter for stream bitrate.
- `#define FE2_VCODECPARAM_RC_MODE`
Video codec parameter for the rate control mode.
- `#define FE2_VCODECPARAM_KFINTTYPE`
Video codec parameter for the keyframe interval type.
- `#define FE2_VCODECPARAM_KFFREQ`
Video codec parameter for keyframe frequency.

Enumerations

- `enum FE2_VideoBitrateControls {`
 `CBR_1PASSControl,`
 `VBR_1PASSControl,`
 `CBR_2PASSControl,`
 `VBR_2PASSControl }`
Encoder rate control types, influences quality.
- `enum FE2_VideoKeyframeTypes {`
 `MAX_KEYFRAMES,`
 `FIXED_KEYFRAMES }`
Key frame interval types, influences quality.

12.4.1 Define Documentation

12.4.1.1 `#define FE2_CODECPARAM_BITRATE`

Codec parameter for stream bitrate. Compressed stream bitrate in kbits/sec.

Definition at line 47 of file `codec_common.h`.

12.4.1.2 `#define FE2_VCODECPARAM_KFFREQ`

Video codec parameter for keyframe frequency. Value is in video frames. The interpretation depends on the setting of `FE2_VCODECPARAM_KFINTTYPE`

See also:

[FE2_VideoKeyframeTypes](#)

Definition at line 67 of file `codec_common.h`.

12.4.1.3 #define FE2_VCODECPARAM_KFINTTYPE

Video codec parameter for the keyframe interval type. Valid values are defined by [FE2_VideoKeyframeTypes](#).

Definition at line 59 of file codec_common.h.

12.4.1.4 #define FE2_VCODECPARAM_RC_MODE

Video codec parameter for the rate control mode. Valid values are defined by [FE2_VideoBitrateControls](#).

Definition at line 53 of file codec_common.h.

12.4.2 Enumeration Type Documentation

12.4.2.1 enum FE2_VideoBitrateControls

Encoder rate control types, influences quality. For use in calls to [Flix2_CodecSetParam\(\)](#) and [Flix2_CodecGetParam\(\)](#) for the [FE2_VCODECPARAM_RC_MODE](#) parameter.

Enumerator:

- CBR_IPASSControl* single pass constant bitrate
- VBR_IPASSControl* single pass variable bitrate
- CBR_2PASSControl* two pass constant bitrate
- VBR_2PASSControl* two pass variable bitrate

Definition at line 26 of file codec_common.h.

12.4.2.2 enum FE2_VideoKeyframeTypes

Key frame interval types, influences quality. For use in calls to [Flix2_CodecSetParam\(\)](#) and [Flix2_CodecGetParam\(\)](#) for the [FE2_VCODECPARAM_KFINTTYPE](#) parameter.

Enumerator:

- MAX_KEYFRAMES* generate keyframes at MOST [FE2_VCODECPARAM_KFFREQ](#) apart
- FIXED_KEYFRAMES* generate keyframes at a fixed interval

Definition at line 37 of file codec_common.h.

12.5 Video Codecs

Modules

- [H263 - FFmpeg](#)
- [H264](#)
- [VP6](#)
- [VP6 with Alpha](#)
- [VP8](#)

12.6 Audio Codecs

Modules

- [AAC](#)
- [AAC+](#)
- [AMR_NB - FFmpeg](#)
- [MP3 - LAME](#)
- [libvorbis - FFmpeg](#)

Defines

- `#define FE2_CODEC_PCM`
'Codec' name for use with [Flix2_AddCodec\(\)](#)

12.6.1 Define Documentation

12.6.1.1 `#define FE2_CODEC_PCM`

'Codec' name for use with [Flix2_AddCodec\(\)](#) Allows for PCM audio output.

Definition at line 65 of file `codec_constants.h`.

12.7 AAC

Defines

- `#define FE2_CODEC_AAC`
AAC. Codec name for use with *Flix2_AddCodec()*.
- `#define FE2_AAC_BITRATE`
Alias for *FE2_CODECPARAM_BITRATE*.

12.7.1 Detailed Description

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_AAC);
// Use 128kbps
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_AAC_BITRATE, 128);
```

Valid Configurations

Channels te (kHz)	Bitrates (kbps)	Sampling Ra
=====	=====	=====
1ch	8,10	8,11.025,12
1ch	12	8,11.025,12
,16		
1ch	16	8 to 24
1ch	20	11.025 to 2
4		
1ch	24,28	11.025 to 3
2		
1ch	32	11.025 to 4
8		
1ch	40	16 to 48
1ch	48,56	22.05 to 48
1ch	64,80,96,112,128,160	32,44.1,48
2ch	16,20	8,11.025,12
2ch	24,28,32	11.025 to 2
4		
2ch	40	16 to 32
2ch	48	22.05 to 32
2ch	56	22.05 to 48
2ch	64,80,96,112,128,160,192,224,256,288,320	32,44.1,48

Attention:

Should the requested bitrate under/overshoot the above bounds it will be adjusted accordingly.

See also:

[Resample](#)

Additional References:

- [ISO - International Organization for Standardization](#)
- [ISO/IEC 14496-3](#):MPEG-4 Audio
- Coding Technologies [aacPlus documentation](#) for further details regarding AAC and aacPlus compatibility

12.7.2 Define Documentation**12.7.2.1 #define FE2_AAC_BITRATE**

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 64

Definition at line 75 of file aac.h.

12.7.2.2 #define FE2_CODEC_AAC

AAC. Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 69 of file aac.h.

12.8 AAC+

Defines

- `#define FE2_CODEC_AACPLUS`
AAC+ (HE-AAC). Codec name for use with *Flix2_AddCodec()*.
- `#define FE2_AACPLUS_BITRATE`
Alias for *FE2_CODECPARAM_BITRATE*.
- `#define FE2_AACPLUS_PARAMETRIC_STEREO`
Parameter for parametric stereo.

12.8.1 Detailed Description

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_AACPLUS);
// Use 128kbps
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_AACPLUS_BITRATE, 128);
```

Valid Configurations

Channels	Bitrates (kbps)	Sampling Rate (kHz)
1ch	8, 10, 12, 16	22.05, 24, 32, 44.1, 48
1ch	20, 24, 28, 32, 40, 48	32, 44.1, 48
1ch	56, 64	32, 44.1, 48, 64, 88.2, 96
1ch	80, 96, 112, 128	64, 88.2, 96
2ch	16, 20, 24, 28, 32, 40, 48, 56, 64	32, 44.1, 48
2ch	80, 96, 112, 128	32, 44.1, 48, 64, 88.2, 96
2ch	160, 192, 224, 256	64, 88.2, 96
2ch-Parametric-Stereo	16, 20, 24, 28, 32, 40, 48, 56	32, 44.1, 48

Attention:

Should the requested bitrate under/overshoot the above bounds it will be adjusted accordingly.

See also:

[Resample](#)

Additional References:

- [ISO - International Organization for Standardization](#)
- [ISO/IEC 14496-3:MPEG-4 Audio](#)
- Coding Technologies [aacPlus documentation](#) for further details regarding AAC and aacPlus compatibility

12.8.2 Define Documentation

12.8.2.1 `#define FE2_AACPLUS_BITRATE`

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 64

Definition at line 126 of file aac.h.

12.8.2.2 `#define FE2_AACPLUS_PARAMETRIC_STEREO`

Parameter for parametric stereo. The Parametric Stereo technology was standardized by ISO 14496-3. It is a method to code stereo information and thus allows for further enhanced coding efficiency for lowest bit rate stereo coding. Combining Parametric Stereo with SBR and AAC yields the aacPlus v2 codec.

Definition at line 135 of file aac.h.

12.8.2.3 `#define FE2_CODEC_AACPLUS`

AAC+ (HE-AAC). Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 120 of file aac.h.

12.9 AMR_NB - FFmpeg

Defines

- `#define FE2_CODEC_AMR_NB`
AMR Narrowband. Codec name for use with [Flix2_AddCodec\(\)](#).
- `#define FE2_AMR_BITRATE`
Alias for [FE2_CODECPARAM_BITRATE](#).

12.9.1 Detailed Description

AMR narrowband encoder.

Valid bitrates:

AMR narrowband only supports a fixed set of bitrates:

codec mode	kbps
=====	=====
MR475	4.75
MR515	5.15
MR59	5.90
MR67	6.70
MR74	7.40
MR795	7.95
MR102	10.2
MR122	12.2

[FE2_AMR_BITRATE](#) will be adjusted as follows, if necessary:

```
FE2_AMR_BITRATE = 0.0  <= FE2_AMR_BITRATE <  5.0,  MR475
                  5.0  <= FE2_AMR_BITRATE <  5.9,  MR515
                  5.9  <= FE2_AMR_BITRATE <  6.7,  MR59
                  6.7  <= FE2_AMR_BITRATE <= 7.0,  MR67
                  7.0  < FE2_AMR_BITRATE <  7.95, MR74
                  7.95 <= FE2_AMR_BITRATE < 10.0, MR795
                  10.0 <= FE2_AMR_BITRATE < 12.0, MR102
                  12.0 <= FE2_AMR_BITRATE,      MR122
```

Attention:

In addition to the bitrate restrictions above, the sample rate **MUST** be 8000Hz and the number of channels **MUST** be 1.

See also:

[Resample](#)

Example Usage:

```
sc = Flix2_AddCodec(&codec, flx, FE2_CODEC_AMR_NB);
// Use 12.2kbps
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_AMR_BITRATE, 12.2);
```

Additional References:

- [3GPP homepage](#)

- [AMR-NB Specification detail](#)
- [FFmpeg project homepage](#)

12.9.2 Define Documentation

12.9.2.1 `#define FE2_AMR_BITRATE`

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 12.2kbps

Definition at line 80 of file amr.h.

12.9.2.2 `#define FE2_CODEC_AMR_NB`

AMR Narrowband. Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 73 of file amr.h.

12.10 H263 - FFmpeg

Defines

- `#define FE2_CODEC_H263`
Codec name for use with `Flix2_AddCodec()`.
- `#define FE2_CODEC_H263_BASELINE`
Codec name for use with `Flix2_AddCodec()`.
- `#define FE2_H263_BITRATE`
Alias for `FE2_CODECPARAM_BITRATE`.
- `#define FE2_H263_KFINTTYPE`
Alias for `FE2_VCODECPARAM_KFINTTYPE`.
- `#define FE2_H263_KFFREQ`
Alias for `FE2_VCODECPARAM_KFFREQ`.
- `#define FE2_H263_RC_MODE`
Alias for `FE2_VCODECPARAM_RC_MODE`.
- `#define FE2_H263_MIN_Q`
Codec parameter for minimum quantizer.
- `#define FE2_H263_MAX_Q`
Codec parameter for maximum quantizer.

12.10.1 Detailed Description

The H263 codec

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_H263);  
// Use 450kbit  
if(sc == ON2_OK)  
    sc = Flix2_CodecSetParam(codec, FE2_H263_BITRATE, 450.0);
```

Additional References:

- [FFmpeg project homepage](#)

12.10.2 Define Documentation

12.10.2.1 `#define FE2_CODEC_H263`

Codec name for use with `Flix2_AddCodec()`. Specifically this codec refers to the H.263 profile used within FLV.

Definition at line 42 of file h263.h.

12.10.2.2 `#define FE2_CODEC_H263_BASELINE`

Codec name for use with [Flix2_AddCodec\(\)](#). Specifically this codec refers to the baseline H.263 profile (profile 0) supported within 3GPP files amongst others.

Attention:

The baseline profile only supports the following resolutions:

- sub-QCIF (128x96)
- QCIF (176x144)
- CIF (352x288)

See also:

[Scale](#)

ITU-T Recommendation H.263 (01/05): "Video coding for low bit rate communication".

Definition at line 56 of file h263.h.

12.10.2.3 `#define FE2_H263_BITRATE`

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 63 of file h263.h.

12.10.2.4 `#define FE2_H263_KFFREQ`

Alias for [FE2_VCODECPARAM_KFFREQ](#). Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of [FE2_VCODECPARAM_KFINTTYPE](#)

See also:

[FE2_VideoKeyframeTypes](#)

Note:

Default: $12.0 \times fps$ or 360 frames if the framerate is unknown

Definition at line 78 of file h263.h.

12.10.2.5 `#define FE2_H263_KFINTTYPE`

Alias for [FE2_VCODECPARAM_KFINTTYPE](#). Video codec parameter for the keyframe interval type.

Valid values are defined by [FE2_VideoKeyframeTypes](#).

Note:

Default: [MAX_KEYFRAMES](#)

Definition at line 70 of file h263.h.

12.10.2.6 #define FE2_H263_MAX_Q

Codec parameter for maximum quantizer.

Note:

Default: 31

Definition at line 94 of file h263.h.

12.10.2.7 #define FE2_H263_MIN_Q

Codec parameter for minimum quantizer.

Note:

Default: 2

Definition at line 89 of file h263.h.

12.10.2.8 #define FE2_H263_RC_MODE

Alias for [FE2_VCODECPARAM_RC_MODE](#).

Note:

Default: [VBR_2PASSControl](#)

Definition at line 84 of file h263.h.

12.11 H264

Defines

- #define [FE2_CODEC_H264](#)
Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_H264_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_H264_KFINTTYPE](#)
Alias for [FE2_VCODECPARAM_KFINTTYPE](#).
- #define [FE2_H264_KFFREQ](#)
Alias for [FE2_VCODECPARAM_KFFREQ](#).
- #define [FE2_H264_RC_MODE](#)
Alias for [FE2_VCODECPARAM_RC_MODE](#).
- #define [FE2_H264_PROFILE](#)
Codec parameter for encoding profile.
- #define [FE2_H264_B_FRAME_RATE](#)
Codec parameter for B frame rate.
- #define [FE2_H264_SPEED](#)
Controls frame analysis, affecting encoder speed and inversely output quality.

Typedefs

- typedef enum [h264profile](#) [h264profile_t](#)
Valid profiles for use with the [FE2_H264_PROFILE](#) parameter.

Enumerations

- enum [h264profile](#) {
 [BASE_H264PROFILE](#),
 [MAIN_H264PROFILE](#),
 [HIGH_H264PROFILE](#) }
Valid profiles for use with the [FE2_H264_PROFILE](#) parameter.

12.11.1 Detailed Description

The H264 video codec.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_H264);
// Use 450kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_H264_BITRATE, 450.0);
```

Attention:

Both width and height MUST be multiples of 4 for codec setup to succeed

Additional References:

- [ISO - International Organization for Standardization](#)
- [ISO/IEC 14496-10: Advanced Video Coding](#)
- [ITU - International Telecommunication Union](#)
- [ITU-T Rec. H.264: Recommendation H.264](#)

12.11.2 Apple device support

As of version 8.0.10.1 preliminary support for the iPhone has been added.

The following settings are required to enable video playback:

- [FE2_H264_PROFILE](#) set to [BASE_H264PROFILE](#)
- Resolution <= 640x480. iPhone native is 480x320, 480x360 is recommended for 4:3 content

See also:

[FE2_MUXER_MP4](#)
[Apple Developer Connection](#) video creation guide (registration required)

12.11.3 Define Documentation

12.11.3.1 #define FE2_CODEC_H264

Codec name for use with [Flix2_AddCodec\(\)](#).

Attention:

Both width and height MUST be multiples of 4

Definition at line 57 of file h264.h.

12.11.3.2 #define FE2_H264_B_FRAME_RATE

Codec parameter for B frame rate. Specifies number of B frames between I/P and next P frame.

Note:

Default: 0

Definition at line 102 of file h264.h.

12.11.3.3 `#define FE2_H264_BITRATE`

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 63 of file h264.h.

12.11.3.4 `#define FE2_H264_KFFREQ`

Alias for [FE2_VCODECPARAM_KFFREQ](#). Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of [FE2_VCODECPARAM_KFINTTYPE](#)

See also:

[FE2_VideoKeyframeTypes](#)

Note:

Default: $12.0 \times fps$ or 360 frames if the framerate is unknown

Definition at line 77 of file h264.h.

12.11.3.5 `#define FE2_H264_KFINTTYPE`

Alias for [FE2_VCODECPARAM_KFINTTYPE](#). Video codec parameter for the keyframe interval type.

Valid values are defined by [FE2_VideoKeyframeTypes](#).

Note:

Default: [MAX_KEYFRAMES](#)

Attention:

Only [MAX_KEYFRAMES](#) is supported at this time

Definition at line 70 of file h264.h.

12.11.3.6 `#define FE2_H264_PROFILE`

Codec parameter for encoding profile.

Note:

Valid values are defined by [h264profile_t](#)

Default: [MAIN_H264PROFILE](#)

Definition at line 89 of file h264.h.

12.11.3.7 #define FE2_H264_RC_MODE

Alias for [FE2_VCODECPARAM_RC_MODE](#).

Note:

Default: [VBR_1PASSControl](#)

Attention:

Only [VBR_1PASSControl](#) or [CBR_1PASSControl](#) are supported at this time

Definition at line 84 of file h264.h.

12.11.3.8 #define FE2_H264_SPEED

Controls frame analysis, affecting encoder speed and inversely output quality. Lower values will produce faster overall encode times, with the potential for quality loss, depending on the input material. A value of 1 provides a good balance between speed and quality, 2 or above will begin to perceptibly lengthen the encode.

Note:

Default:

- [BASE_H264PROFILE/MAIN_H264PROFILE](#): 1
- [HIGH_H264PROFILE](#): 3

Valid Range: [0,5]

Definition at line 116 of file h264.h.

12.11.4 Typedef Documentation

12.11.4.1 typedef enum h264profile h264profile_t

Valid profiles for use with the [FE2_H264_PROFILE](#) parameter.

12.11.5 Enumeration Type Documentation

12.11.5.1 enum h264profile

Valid profiles for use with the [FE2_H264_PROFILE](#) parameter.

Enumerator:

BASE_H264PROFILE

MAIN_H264PROFILE

HIGH_H264PROFILE

Definition at line 92 of file h264.h.

12.12 MP3 - LAME

Defines

- #define [FE2_CODEC_LAME](#)
Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_LAME_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_LAME_QUALITY](#)
Codec parameter for controlling LAME library's algorithm selection.
- #define [FE2_LAME_RC_MODE](#)
Codec parameter for controlling LAME library's rate control method.
- #define [FE2_LAME_CHANNELS](#)
Codec parameter for number of output channels.

Typedefs

- typedef enum [lame_rcmode](#) [lame_rcmode_t](#)
Rate control modes analogous to those found in `<lame/lame.h>`.

Enumerations

- enum [lame_rcmode](#) {
 [LAME_CBR](#),
 [LAME_ABR](#),
 [LAME_VBR_rh](#),
 [LAME_VBR_mtrh](#) }
Rate control modes analogous to those found in `<lame/lame.h>`.

12.12.1 Detailed Description

The LAME codec is used to produce MP3 audio streams.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flx, FE2_CODEC_LAME);  
// Use 128kbit  
if(sc == ON2_OK)  
    sc = Flix2_CodecSetParam(codec, FE2_LAME_BITRATE, Bitrate128kbps);
```

Additional References:

[LAME project homepage](#)

12.12.2 Define Documentation

12.12.2.1 #define FE2_CODEC_LAME

Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 41 of file lame.h.

12.12.2.2 #define FE2_LAME_BITRATE

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec. Valid bitrates are defined by [FE2_AudioBitrates](#)

Note:

Default: Bitrate64kbps

Attention:

Some combinations of bitrate and samplerate are invalid based on the MP3 specification. Setting an invalid pair will cause the codec setup to fail.

For further details please see: http://www.mpgedit.org/mpgedit/mpeg_format/mpeghdr.htm

Validity can be determined by inspecting the bitrate and samplerate indices.

Definition at line 55 of file lame.h.

12.12.2.3 #define FE2_LAME_CHANNELS

Codec parameter for number of output channels.

Note:

Default: input number of channels as described by [audio_options_SetStereo\(\)](#)

This parameter is temporary and will be removed when a proper filter is added in a future release

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

Definition at line 97 of file lame.h.

12.12.2.4 #define FE2_LAME_QUALITY

Codec parameter for controlling LAME library's algorithm selection. From `<lame/lame.h>`:

```
internal algorithm selection. True quality is determined by the bitrate
but this variable will effect quality by selecting expensive or cheap
algorithms.
quality=0..9. 0=best (very slow). 9=worst.
recommended: 2    near-best quality, not too slow
              5    good quality, fast
              7    ok quality, really fast
```

Note:

Valid range: [0,9]
Default: 5

Definition at line 72 of file lame.h.

12.12.2.5 #define FE2_LAME_RC_MODE

Codec parameter for controlling LAME library's rate control method.

Note:

Valid values are defined by [lame_rcmode_t](#)
Default: [LAME_CBR](#)

Definition at line 77 of file lame.h.

12.12.3 Typedef Documentation**12.12.3.1 typedef enum lame_rcmode lame_rcmode_t**

Rate control modes analogous to those found in `<lame/lame.h>`.

12.12.4 Enumeration Type Documentation**12.12.4.1 enum lame_rcmode**

Rate control modes analogous to those found in `<lame/lame.h>`.

Enumerator:

LAME_CBR constant bitrate
LAME_ABR average bitrate, produces predictable size output w/improved quality over CBR
LAME_VBR_rh variable bitrate, also referred to as vbr-old/standard
LAME_VBR_mtrh variable bitrate, also referred to as vbr-new/fast (over twice as fast as vbr-old)

Definition at line 80 of file lame.h.

12.13 libvorbis - FFmpeg

Defines

- `#define FE2_CODEC_VORBIS`
Vorbis. Codec name for use with [Flix2_AddCodec\(\)](#).
- `#define FE2_VORBIS_BITRATE`
Alias for [FE2_CODECPARAM_BITRATE](#).

12.13.1 Detailed Description

Example Usage:

```
sc = Flix2_AddCodec(&codec, flx, FE2_CODEC_VORBIS);  
// Use 128kbps  
if(sc == ON2_OK)  
    sc = Flix2_CodecSetParam(codec, FE2_VORBIS_BITRATE, 128);
```

12.13.2 Define Documentation

12.13.2.1 `#define FE2_CODEC_VORBIS`

Vorbis. Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 36 of file vorbis.h.

12.13.2.2 `#define FE2_VORBIS_BITRATE`

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 64kbps

Definition at line 43 of file vorbis.h.

12.14 VP6

General VP6 settings

- enum `FE2_CompressMode` {
 `COMPRESSMODE_GOOD`,
 `COMPRESSMODE_BEST` }
 Valid compress modes for VP6, influences encoder speed.
- enum `vp6profile` {
 `VP6_E`,
 `VP6_S` }
 Valid profiles for use with the `FE2_VP6_PROFILE` parameter.
- typedef enum `vp6profile` `vp6profile_t`
 Valid profiles for use with the `FE2_VP6_PROFILE` parameter.
- #define `FE2_VP6_BITRATE`
 Alias for `FE2_CODECPARAM_BITRATE`.
- #define `FE2_VP6_KFINTTYPE`
 Alias for `FE2_VCODECPARAM_KFINTTYPE`.
- #define `FE2_VP6_KFFREQ`
 Alias for `FE2_VCODECPARAM_KFFREQ`.
- #define `FE2_VP6_RC_MODE`
 Alias for `FE2_VCODECPARAM_RC_MODE`.
- #define `FE2_VP6_CXMODE`
 VP6 compress mode.
- #define `FE2_VP6_SHARPNESS`
 Codec parameter for sharpness.
- #define `FE2_VP6_NOISE_REDUCTION`
 Codec parameter for noise reduction.
- #define `FE2_VP6_PROFILE`
 Codec parameter for encoding profile.

Codec name

- #define `FE2_CODEC_VP6`
 Codec name for use with `Flix2_AddCodec()`.

Advanced VP6 settings

- #define [FE2_VP6_CONCURRENCY](#)
Codec parameter for concurrency level.
- #define [FE2_VP6_UNDERSHOOT_PCT](#)
Codec parameter for undershoot percentage.
- #define [FE2_VP6_MIN_Q](#)
Codec parameter for minimum quantizer.
- #define [FE2_VP6_MAX_Q](#)
Codec parameter for maximum quantizer.
- #define [FE2_VP6_TEMPORAL_RESAMPLING](#)
Codec parameter for temporal resampling.
- #define [FE2_VP6_TEMPORAL_DOWN_WATERMARK](#)
Codec parameter for temporal down watermark percentage.

CBR-Specific

- #define [FE2_VP6_STREAM_PEAK_BITRATE](#)
The maximum bitrate allowed in the stream.
- #define [FE2_VP6_STREAM_PREBUFFER](#)
Seconds of preload that are necessary before starting playback.
- #define [FE2_VP6_STREAM_OPTIMAL_BUFFER](#)
Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.
- #define [FE2_VP6_STREAM_MAX_BUFFER](#)
The maximum size of the buffer, in seconds.

VBR-Specific

- #define [FE2_VP6_2PASS_MIN_SECTION](#)
VBR_2PASSControl minimum section datarate
- #define [FE2_VP6_2PASS_MAX_SECTION](#)
VBR_2PASSControl maximum section datarate

12.14.1 Detailed Description

The VP6 (Flash 8) video codec.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flx, FE2_CODEC_VP6);
// Use 450kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP6_BITRATE, 450.0);
```

12.14.2 Defaults dependent on bits per pixel

Defaults for some encoder parameters are determined based on the bits per pixel, which is determined as follows:

$$bitsperpixel = \frac{FE2_VP6_BITRATE * 1024}{imagew * imageh * fps}$$

The current defaults are:

	<.03	<.07	<.12	<.20	>=.20
FE2_VP6_MIN_Q	25	15	15	5	5
FE2_VP6_MAX_Q	62	56	50	40	32
FE2_VP6_SHARPNESS	1	1	5	7	7
FE2_VP6_NOISE_REDUCTION	3	0	0	0	0
FE2_VP6_TEMPORAL_RESAMPLING	1	1	0	0	0

12.14.3 Notes on datarate control

Using the above defaults some material may fail to achieve the requested bitrate. This is highly material dependent, but the result will be a clip with a reduced datarate. If you are intent on achieving a specific datarate and find that your clips are missing it the following settings will help:

FE2_VP6_MIN_Q	4
FE2_VP6_NOISE_REDUCTION	0
FE2_VP6_TEMPORAL_RESAMPLING	1
FE2_VP6_STREAM_PREBUFFER	5
FE2_VP6_STREAM_OPTIMAL_BUFFER	5

Reasoning:

- **FE2_VP6_MIN_Q** setting - The default minimum quantizer settings are an attempt to save bits for harder sections. Unfortunately this means that on easy but high framerate and/or resolution clips the desired quality cannot be achieved. The downside to this change is that more frames may be dropped on harder sections.
- **FE2_VP6_NOISE_REDUCTION** setting - The usage of noise sensitivity set to non-0 is an attempt to save bits by blurring the source. Sometimes the material is so easy that this attempt to save bits may not only be unnecessary, it may actually hurt the visual quality by blurring. Again this may cause more frames to be dropped in harder sections.
- **FE2_VP6_TEMPORAL_RESAMPLING** setting - This parameter should be enabled to guarantee there are no extreme datarate spikes throughout the clip.

12.14.4 Define Documentation

12.14.4.1 #define FE2_CODEC_VP6

Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 84 of file vp6.h.

12.14.4.2 #define FE2_VP6_2PASS_MAX_SECTION

[VBR_2PASSControl](#) maximum section datarate This value is given as a percentage. The highest datarate, i.e., $FE2_VP6_2PASS_MAX_SECTION\% \times FE2_VP6_BITRATE$, that can be streamed, and also the highest datarate that the encoder will allow, no matter how difficult the section is.

Note:

Default: 400

Definition at line 336 of file vp6.h.

12.14.4.3 #define FE2_VP6_2PASS_MIN_SECTION

[VBR_2PASSControl](#) minimum section datarate This value is given as a percentage. The lowest datarate, i.e., $FE2_VP6_2PASS_MIN_SECTION\% \times FE2_VP6_BITRATE$, that the encoder will allow for any section, no matter how easy the section is. This value is used to prevent difficult sections from stealing too many bits from easy sections.

Note:

Default: 40

Definition at line 326 of file vp6.h.

12.14.4.4 #define FE2_VP6_BITRATE

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 124 of file vp6.h.

12.14.4.5 #define FE2_VP6_CONCURRENCY

Codec parameter for concurrency level. Setting this value to 1 will allow the encode process to take advantage of multiple cores/processors yielding a potentially [significant](#) gain in performance.

Usage Considerations:

The current implementation parallelizes the video encode on keyframe boundaries. To achieve this, each instance (1 per core/processor) must buffer both the raw input frames as well as those encoded. As a result, the memory requirement for each session increases dramatically.

The number of raw (YV12) input frames required per processor is [FE2_VP6_KFFREQ](#) or 60 (the current allowed maximum), whichever is smaller. The size of each raw frame can be calculated as follows:

$$framesiz_bytes = imagew \times imageh \times \frac{3}{2}$$

Making the input buffer requirements for each processor:

$$raw_bufreq = framesiz_bytes \times \min\{FE2_VP6_KFFREQ, 60\}$$

The encoded buffer requirement is the same, as this would be the worst (though unlikely) case, making the per-processor memory increase:

$$memreq_increase/processor = 2 \times raw_bufreq$$

The current maximum number of cores/processors utilized is 8 with the load evenly distributed across those available.

Because of this some thought should be given to queueing encodes to avoid possible failures due to exhaustion of available memory.

Using the above, with [FE2_VP6_KFFREQ](#)=60, the per-processor memory increase for some common resolutions are:

Resolution		framesiz_bytes		memreq_increase/processor		memreq_increase/2	
	rease/2		memreq_increase/4		memreq_increase/8		
=====							
320x240 (QVGA)		115200 (.11MiB)		13824000 (13.18MiB)		27648000 (27.37MiB)	(2)
640x480 (VGA)		55296000 (52.73MiB)		110592000 (105.47MiB)		221184000 (210.94MiB)	(1)
720x480 (NTSC)		460800 (.44MiB)		55296000 (52.73MiB)		110592000 (105.47MiB)	(1)
720x576 (PAL)		221184000 (210.94MiB)		442368000 (421.88MiB)		884736000 (843.76MiB)	(3)
1280x720 (720p)		518400 (.49MiB)		62208000 (59.33MiB)		124416000 (120.66MiB)	(1)
		248832000 (237.30MiB)		497664000 (474.61MiB)		995328000 (949.22MiB)	(3)
		622080 (.59MiB)		74649600 (71.19MiB)		149299200 (143.99MiB)	(1)
		298598400 (284.77MiB)		597196800 (569.53MiB)		1194393600 (1139.06MiB)	(3)
		1382400 (1.32MiB)		165888000 (158.2MiB)		331776000 (316.41MiB)	(1)
		663552000 (632.81MiB)		1327104000 (1265.63MiB)		2654208000 (2531.26MiB)	(3)

Note:

Default: 0 (disabled)

Attention:

Currently to enable this parameter:

- [FE2_VP6_KFINTTYPE](#) MUST be set to [FIXED_KEYFRAMES](#).
- [FE2_VP6_RC_MODE](#) MUST be set to [VBR_1PASSControl](#) or [CBR_1PASSControl](#).

Enabling this parameter on a processor that is solely Hyper-Threaded (e.g., early Pentium 4 models) MAY see a **decrease** in performance, especially under Linux.

See also:

[FE2_VP6_CONCURRENCY Performance](#) - Graphs comparing encode times with and without this parameter enabled.

Definition at line 228 of file vp6.h.

12.14.4.6 #define FE2_VP6_CXMODE

VP6 compress mode. Valid values are defined by [FE2_CompressMode](#)

Note:

Default: [COMPRESSMODE_GOOD](#)

Definition at line 149 of file vp6.h.

12.14.4.7 #define FE2_VP6_KFFREQ

Alias for [FE2_VCODECPARAM_KFFREQ](#). Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of [FE2_VCODECPARAM_KFINTTYPE](#)

See also:

[FE2_VideoKeyframeTypes](#)

Note:

Default: $12.0 \times output_{fps}$ or 360 frames if the framerate is unknown

Definition at line 137 of file vp6.h.

12.14.4.8 #define FE2_VP6_KFINTTYPE

Alias for [FE2_VCODECPARAM_KFINTTYPE](#). Video codec parameter for the keyframe interval type.

Valid values are defined by [FE2_VideoKeyframeTypes](#).

Note:

Default: [MAX_KEYFRAMES](#)

Definition at line 130 of file vp6.h.

12.14.4.9 #define FE2_VP6_MAX_Q

Codec parameter for maximum quantizer. Determines the quality of the output. A lower maximum number produces higher quality output.

Note:

Valid range: [0,63]

Default: [Bits per pixel dependent](#)

Attention:

Setting this value too low will likely cause the encoder to miss (i.e., overshoot) the target datarate specified by [FE2_VP6_BITRATE](#) and, should [FE2_VP6_TEMPORAL_RESAMPLING](#) be enabled, result in a large amount of dropped frames. The table mentioned above should be used as a guide in choosing a reasonable value based on the bitrate.

Definition at line 258 of file vp6.h.

12.14.4.10 `#define FE2_VP6_MIN_Q`

Codec parameter for minimum quantizer. Determines the quality of the output. A lower minimum number produces higher quality output.

Note:

Valid range: [0,63]

Default: [Bits per pixel dependent](#)

Definition at line 244 of file vp6.h.

12.14.4.11 `#define FE2_VP6_NOISE_REDUCTION`

Codec parameter for noise reduction. Determines the level of noise filtering to apply in the preprocessor. 0 is no preprocessing, 6 is extreme preprocessing.

Note:

Valid range: [0,6]

Setting this value to anything but 0 will result in slowing down the compression speed.

Default: [Bits per pixel dependent](#)

Definition at line 170 of file vp6.h.

12.14.4.12 `#define FE2_VP6_PROFILE`

Codec parameter for encoding profile.

Note:

Valid values are defined by [vp6profile_t](#) Default: [VP6_E](#)

Definition at line 175 of file vp6.h.

12.14.4.13 `#define FE2_VP6_RC_MODE`

Alias for [FE2_VCODECPARAM_RC_MODE](#). Video codec parameter for the rate control mode.

Valid values are defined by [FE2_VideoBitrateControls](#).

Note:

Default: [VBR_2PASSControl](#)

Definition at line 143 of file vp6.h.

12.14.4.14 `#define FE2_VP6_SHARPNESS`

Codec parameter for sharpness. Controls the sharpness of the image in the output. This setting does not impact any other setting and is largely a matter of personal preference. A low sharpness setting will result in fewer visible artifacts but may blur the image somewhat; a high sharpness will result in a sharper image but may result in more visible artifacts.

Note:

Valid range: [0,7]

Default: [Bits per pixel dependent](#)

Definition at line 160 of file vp6.h.

12.14.4.15 #define FE2_VP6_STREAM_MAX_BUFFER

The maximum size of the buffer, in seconds.

Attention:

Valid for CBR only

Definition at line 311 of file vp6.h.

12.14.4.16 #define FE2_VP6_STREAM_OPTIMAL_BUFFER

Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

Note:

Default: 10

Attention:

Valid for CBR only

Definition at line 307 of file vp6.h.

12.14.4.17 #define FE2_VP6_STREAM_PEAK_BITRATE

The maximum bitrate allowed in the stream. This value is given as a percentage. The peak bitrate is calculated as follows: $FE2_VP6_STREAM_PEAK_BITRATE\% \times FE2_VP6_BITRATE$

Note:

Default: 100

Attention:

Valid for CBR only

Definition at line 293 of file vp6.h.

12.14.4.18 #define FE2_VP6_STREAM_PREBUFFER

Seconds of preload that are necessary before starting playback. The buffer is used to maintain a consistent datarate and minimize playback interruption.

Note:

Default: 6

Attention:

Valid for CBR only

Definition at line 301 of file vp6.h.

12.14.4.19 #define FE2_VP6_TEMPORAL_DOWN_WATERMARK

Codec parameter for temporal down watermark percentage. Specifies the percentage of the datarate buffer remaining below which the encoder is allowed to start dropping frames. Only used if [FE2_VP6_TEMPORAL_RESAMPLING](#) is enabled.

Note:

Default: 20

A larger percentage will make it more likely frames will be dropped to achieve the requested [FE2_VP6_BITRATE](#). The converse is also true.

Definition at line 281 of file vp6.h.

12.14.4.20 #define FE2_VP6_TEMPORAL_RESAMPLING

Codec parameter for temporal resampling. In particularly difficult regions, if enabled, the encoder will drop frames to achieve the target the data rate. The value of [FE2_VP6_TEMPORAL_DOWN_WATERMARK](#) determines the percentage of the datarate buffer below which the encoder is enabled to start dropping frames.

Note:

Default: [Bits per pixel dependent](#)

Attention:

Setting [FE2_VP6_RC_MODE](#) to [CBR_1PASSControl](#) or [CBR_2PASSControl](#) will unconditionally set this value to 1. In this case adjusting [FE2_VP6_TEMPORAL_DOWN_WATERMARK](#) may improve results.

Definition at line 271 of file vp6.h.

12.14.4.21 #define FE2_VP6_UNDERSHOOT_PCT

Codec parameter for undershoot percentage. This value is given as a percentage. Creates output that targets a slightly lower datarate so there are bits available in the buffer to improve difficult sections.

Note:

Default: 90

Definition at line 236 of file vp6.h.

12.14.5 Typedef Documentation**12.14.5.1 typedef enum vp6profile vp6profile_t**

Valid profiles for use with the [FE2_VP6_PROFILE](#) parameter.

12.14.6 Enumeration Type Documentation

12.14.6.1 enum FE2_CompressMode

Valid compress modes for VP6, influences encoder speed. For use in calls to [Flix2_CodecSetParam\(\)](#) and [Flix2_CodecGetParam\(\)](#) for the [FE2_VP6_CXMODE](#) and [FE2_VP6A_CXMODE](#). For general transcoding (i.e. decoding from one compressed format and encoding to VP6) `COMPRESSMODE_GOOD` will provide adequate results and the best possible encode time. The quality gain using `COMPRESSMODE_BEST` will only be visible when using clean raw source or compressing to extremely low bitrates.

Enumerator:

COMPRESSMODE_GOOD The encoder balances quality with the amount of time it takes to encode the output with the goal of making the encoding as fast as possible without losing too much quality. This is the default.

COMPRESSMODE_BEST The encoder places priority on the quality of the output over encoding speed. The output is compressed at the highest possible quality. This option takes the longest amount of time to encode.

Definition at line 100 of file `vp6.h`.

12.14.6.2 enum vp6profile

Valid profiles for use with the [FE2_VP6_PROFILE](#) parameter.

Enumerator:

VP6_E Default profile used in flash video encoding

VP6_S Profile optimized for use with high resolution/datarate video. Decreases decoding complexity allowing playback on resource constrained machines.

Definition at line 113 of file `vp6.h`.

12.15 VP6 with Alpha

Codec name

- #define [FE2_CODEC_VP6ALPHA](#)
Codec name for use with `Flix2_AddCodec()`.

General VP6 settings

- #define [FE2_VP6A_BITRATE](#)
Alias for `FE2_CODECPARAM_BITRATE`.
- #define [FE2_VP6A_ALPHA_BITRATE](#)
Compressed bitrate of the alpha channel in kbit/s.
- #define [FE2_VP6A_KFINTTYPE](#)
Alias for `FE2_VP6_KFINTTYPE`.
- #define [FE2_VP6A_KFFREQ](#)
Alias for `FE2_VP6_KFFREQ`.
- #define [FE2_VP6A_RC_MODE](#)
Alias for `FE2_VCODECPARAM_RC_MODE`.
- #define [FE2_VP6A_CXMODE](#)
Alias for `FE2_VP6_CXMODE`.
- #define [FE2_VP6A_SHARPNESS](#)
Alias for `FE2_VP6_SHARPNESS`.
- #define [FE2_VP6A_ALPHA_SHARPNESS](#)
Sharpness for the alpha channel.
- #define [FE2_VP6A_NOISE_REDUCTION](#)
Alias for `FE2_VP6_NOISE_REDUCTION`.
- #define [FE2_VP6A_ALPHA_NOISE_REDUCTION](#)
Noise reduction for the alpha channel.

Advanced VP6 settings

- #define [FE2_VP6A_UNDERSHOOT_PCT](#)
Alias for `FE2_VP6_UNDERSHOOT_PCT`.
- #define [FE2_VP6A_MIN_Q](#)
Alias for `FE2_VP6_MIN_Q`.

- `#define FE2_VP6A_ALPHA_MIN_Q`
Minimum quantizer for the alpha channel.
- `#define FE2_VP6A_MAX_Q`
Alias for `FE2_VP6_MAX_Q`.
- `#define FE2_VP6A_ALPHA_MAX_Q`
Maximum quantizer for the alpha channel.
- `#define FE2_VP6A_TEMPORAL_RESAMPLING`
Alias for `FE2_VP6_TEMPORAL_RESAMPLING`.
- `#define FE2_VP6A_TEMPORAL_DOWN_WATERMARK`
Alias for `FE2_VP6_TEMPORAL_DOWN_WATERMARK`.
- `#define FE2_VP6A_STREAM_PEAK_BITRATE`
Alias for `FE2_VP6_STREAM_PEAK_BITRATE`.
- `#define FE2_VP6A_STREAM_PREBUFFER`
Alias for `FE2_VP6_STREAM_PREBUFFER`.
- `#define FE2_VP6A_STREAM_OPTIMAL_BUFFER`
Alias for `FE2_VP6_STREAM_OPTIMAL_BUFFER`.
- `#define FE2_VP6A_STREAM_MAX_BUFFER`
Alias for `FE2_VP6_STREAM_MAX_BUFFER`.
- `#define FE2_VP6A_2PASS_MIN_SECTION`
Alias for `FE2_VP6_2PASS_MIN_SECTION`.
- `#define FE2_VP6A_2PASS_MAX_SECTION`
Alias for `FE2_VP6_2PASS_MAX_SECTION`.

12.15.1 Detailed Description

The VP6 codec with support for an alpha channel.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flx, FE2_CODEC_VP6ALPHA);
//Use 380kbit for the video ..
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP6A_BITRATE, 380.0);
//..and 68kbit for the alpha channel
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP6A_ALPHA_BITRATE, 68.0);
```

As mentioned in the [VP6](#) section [Defaults dependent on bits per pixel](#) some of the codec parameters are determined by the bits per pixel in the output. Currently the alpha channel parameters, e.g. `FE2_VP6A_ALPHA_MIN_Q`, match their VP6 equivalents.

12.15.2 Define Documentation

12.15.2.1 `#define FE2_CODEC_VP6ALPHA`

Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 56 of file `vp6_alpha.h`.

12.15.2.2 `#define FE2_VP6A_2PASS_MAX_SECTION`

Alias for [FE2_VP6_2PASS_MAX_SECTION](#). [VBR_2PASSControl](#) maximum section datarate

This value is given as a percentage. The highest datarate, i.e., $FE2_VP6_2PASS_MAX_SECTION\% \times FE2_VP6_BITRATE$, that can be streamed, and also the highest datarate that the encoder will allow, no matter how difficult the section is.

Note:

Default: 400

Definition at line 202 of file `vp6_alpha.h`.

12.15.2.3 `#define FE2_VP6A_2PASS_MIN_SECTION`

Alias for [FE2_VP6_2PASS_MIN_SECTION](#). [VBR_2PASSControl](#) minimum section datarate

This value is given as a percentage. The lowest datarate, i.e., $FE2_VP6_2PASS_MIN_SECTION\% \times FE2_VP6_BITRATE$, that the encoder will allow for any section, no matter how easy the section is. This value is used to prevent difficult sections from stealing too many bits from easy sections.

Note:

Default: 40

Definition at line 196 of file `vp6_alpha.h`.

12.15.2.4 `#define FE2_VP6A_ALPHA_BITRATE`

Compressed bitrate of the alpha channel in kbit/s.

Note:

Default: 68 (15% of default 448kbps)

Definition at line 74 of file `vp6_alpha.h`.

12.15.2.5 `#define FE2_VP6A_ALPHA_MAX_Q`

Maximum quantizer for the alpha channel.

See also:

[FE2_VP6A_MAX_Q](#)

Definition at line 154 of file `vp6_alpha.h`.

12.15.2.6 #define FE2_VP6A_ALPHA_MIN_Q

Minimum quantizer for the alpha channel.

See also:

[FE2_VP6A_MIN_Q](#)

Definition at line 143 of file vp6_alpha.h.

12.15.2.7 #define FE2_VP6A_ALPHA_NOISE_REDUCTION

Noise reduction for the alpha channel.

See also:

[FE2_VP6A_NOISE_REDUCTION](#)

Definition at line 120 of file vp6_alpha.h.

12.15.2.8 #define FE2_VP6A_ALPHA_SHARPNESS

Sharpness for the alpha channel.

See also:

[FE2_VP6A_SHARPNESS](#)

Definition at line 109 of file vp6_alpha.h.

12.15.2.9 #define FE2_VP6A_BITRATE

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 380

Definition at line 69 of file vp6_alpha.h.

12.15.2.10 #define FE2_VP6A_CXMODE

Alias for [FE2_VP6_CXMODE](#). VP6 compress mode.

Valid values are defined by [FE2_CompressMode](#)

Note:

Default: [COMPRESSMODE_GOOD](#)

Definition at line 98 of file vp6_alpha.h.

12.15.2.11 #define FE2_VP6A_KFFREQ

Alias for [FE2_VP6_KFFREQ](#). Alias for [FE2_VCODECPARAM_KFFREQ](#).

Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of [FE2_VCODECPARAM_KFINTTYPE](#)

See also:

[FE2_VideoKeyframeTypes](#)

Note:

Default: $12.0 \times output_{fps}$ or 360 frames if the framerate is unknown

Definition at line 86 of file `vp6_alpha.h`.

12.15.2.12 #define FE2_VP6A_KFINTTYPE

Alias for [FE2_VP6_KFINTTYPE](#). Alias for [FE2_VCODECPARAM_KFINTTYPE](#).

Video codec parameter for the keyframe interval type.

Valid values are defined by [FE2_VideoKeyframeTypes](#).

Note:

Default: [MAX_KEYFRAMES](#)

Definition at line 80 of file `vp6_alpha.h`.

12.15.2.13 #define FE2_VP6A_MAX_Q

Alias for [FE2_VP6_MAX_Q](#). Codec parameter for maximum quantizer.

Determines the quality of the output. A lower maximum number produces higher quality output.

Note:

Valid range: [0,63]

Default: [Bits per pixel dependent](#)

Attention:

Setting this value too low will likely cause the encoder to miss (i.e., overshoot) the target datarate specified by [FE2_VP6_BITRATE](#) and, should [FE2_VP6_TEMPORAL_RESAMPLING](#) be enabled, result in a large amount of dropped frames. The table mentioned above should be used as a guide in choosing a reasonable value based on the bitrate.

Definition at line 149 of file `vp6_alpha.h`.

12.15.2.14 #define FE2_VP6A_MIN_Q

Alias for [FE2_VP6_MIN_Q](#). Codec parameter for minimum quantizer.

Determines the quality of the output. A lower minimum number produces higher quality output.

Note:

Valid range: [0,63]

Default: [Bits per pixel dependent](#)

Definition at line 138 of file vp6_alpha.h.

12.15.2.15 #define FE2_VP6A_NOISE_REDUCTION

Alias for [FE2_VP6_NOISE_REDUCTION](#). Codec parameter for noise reduction.

Determines the level of noise filtering to apply in the preprocessor. 0 is no preprocessing, 6 is extreme preprocessing.

Note:

Valid range: [0,6]

Setting this value to anything but 0 will result in slowing down the compression speed.

Default: [Bits per pixel dependent](#)

Definition at line 115 of file vp6_alpha.h.

12.15.2.16 #define FE2_VP6A_RC_MODE

Alias for [FE2_VCODECPARAM_RC_MODE](#).

Note:

Default: [VBR_2PASSControl](#)

Definition at line 92 of file vp6_alpha.h.

12.15.2.17 #define FE2_VP6A_SHARPNESS

Alias for [FE2_VP6_SHARPNESS](#). Codec parameter for sharpness.

Controls the sharpness of the image in the output. This setting does not impact any other setting and is largely a matter of personal preference. A low sharpness setting will result in fewer visible artifacts but may blur the image somewhat; a high sharpness will result in a sharper image but may result in more visible artifacts.

Note:

Valid range: [0,7]

Default: [Bits per pixel dependent](#)

Definition at line 104 of file vp6_alpha.h.

12.15.2.18 #define FE2_VP6A_STREAM_MAX_BUFFER

Alias for [FE2_VP6_STREAM_MAX_BUFFER](#). The maximum size of the buffer, in seconds.

Attention:

Valid for CBR only

Definition at line 190 of file vp6_alpha.h.

12.15.2.19 #define FE2_VP6A_STREAM_OPTIMAL_BUFFER

Alias for [FE2_VP6_STREAM_OPTIMAL_BUFFER](#). Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

Note:

Default: 10

Attention:

Valid for CBR only

Definition at line 184 of file vp6_alpha.h.

12.15.2.20 #define FE2_VP6A_STREAM_PEAK_BITRATE

Alias for [FE2_VP6_STREAM_PEAK_BITRATE](#). The maximum bitrate allowed in the stream.

This value is given as a percentage. The peak bitrate is calculated as follows:
 $FE2_VP6_STREAM_PEAK_BITRATE\% \times FE2_VP6_BITRATE$

Note:

Default: 100

Attention:

Valid for CBR only

Definition at line 172 of file vp6_alpha.h.

12.15.2.21 #define FE2_VP6A_STREAM_PREBUFFER

Alias for [FE2_VP6_STREAM_PREBUFFER](#). Seconds of preload that are necessary before starting playback.

The buffer is used to maintain a consistent datarate and minimize playback interruption.

Note:

Default: 6

Attention:

Valid for CBR only

Definition at line 178 of file vp6_alpha.h.

12.15.2.22 #define FE2_VP6A_TEMPORAL_DOWN_WATERMARK

Alias for [FE2_VP6_TEMPORAL_DOWN_WATERMARK](#). Codec parameter for temporal down watermark percentage.

Specifies the percentage of the datarate buffer remaining below which the encoder is allowed to start dropping frames. Only used if [FE2_VP6_TEMPORAL_RESAMPLING](#) is enabled.

Note:

Default: 20

A larger percentage will make it more likely frames will be dropped to achieve the requested [FE2_VP6_BITRATE](#). The converse is also true.

Definition at line 166 of file vp6_alpha.h.

12.15.2.23 #define FE2_VP6A_TEMPORAL_RESAMPLING

Alias for [FE2_VP6_TEMPORAL_RESAMPLING](#). Codec parameter for temporal resampling.

In particularly difficult regions, if enabled, the encoder will drop frames to achieve the target the data rate. The value of [FE2_VP6_TEMPORAL_DOWN_WATERMARK](#) determines the percentage of the datarate buffer below which the encoder is enabled to start dropping frames.

Note:

Default: [Bits per pixel dependent](#)

Attention:

Setting [FE2_VP6_RC_MODE](#) to [CBR_1PASSControl](#) or [CBR_2PASSControl](#) will unconditionally set this value to 1. In this case adjusting [FE2_VP6_TEMPORAL_DOWN_WATERMARK](#) may improve results.

Definition at line 160 of file vp6_alpha.h.

12.15.2.24 #define FE2_VP6A_UNDERSHOOT_PCT

Alias for [FE2_VP6_UNDERSHOOT_PCT](#). Codec parameter for undershoot percentage.

This value is given as a percentage. Creates output that targets a slightly lower datarate so there are bits available in the buffer to improve difficult sections.

Note:

Default: 90

Definition at line 132 of file vp6_alpha.h.

12.16 VP8

Defines

- #define [FE2_VP8_PROFILE](#)
Bitstream profile.
- #define [FE2_VP8_ALTREF](#)
Enable the use of alternate reference frames.
- #define [FE2_VP8_AR_MAX_FRAMES](#)
Max number of frames blurred creating alternate reference.
- #define [FE2_VP8_AR_TYPE](#)
Filter type to use w/alternate reference.
- #define [FE2_VP8_AR_STRENGTH](#)
Filter strength for the alternate reference.
- #define [FE2_VP8_MB_STATIC_THRESHOLD](#)
Threshold for macroblocks treated static.
- #define [FE2_VP8_TOKEN_PARTITIONS](#)
Number of token partitions.
- #define [FE2_VP8_LAG](#)
Allow lagged encoding If set, this value allows the encoder to consume a number of input frames before producing output frames. This allows the encoder to base decisions for the current frame on future frames.
- #define [FE2_VP8_THREADS](#)
Number of threads to use A reasonable selection would be the number of cores on the system.

Codec name

- #define [FE2_CODEC_VP8](#)
Codec name for use with [Flix2_AddCodec\(\)](#).

General VP8 settings

- #define [FE2_VP8_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_VP8_KFINTTYPE](#)
Alias for [FE2_VCODECPARAM_KFINTTYPE](#).
- #define [FE2_VP8_KFFREQ](#)
Alias for [FE2_VCODECPARAM_KFFREQ](#).

- #define [FE2_VP8_RC_MODE](#)
Alias for [FE2_VCODECPARAM_RC_MODE](#).
- #define [FE2_VP8_CXMODE](#)
VP8 compress mode.
- #define [FE2_VP8_SHARPNESS](#)
Codec parameter for sharpness.
- #define [FE2_VP8_NOISE_REDUCTION](#)
Controls encoder noise reduction preprocessing.

Advanced VP8 settings

- #define [FE2_VP8_UNDERSHOOT_PCT](#)
Codec parameter for undershoot percentage.
- #define [FE2_VP8_OVERSHOOT_PCT](#)
Codec parameter for undershoot percentage.
- #define [FE2_VP8_MIN_Q](#)
Codec parameter for minimum quantizer.
- #define [FE2_VP8_MAX_Q](#)
Codec parameter for maximum quantizer.
- #define [FE2_VP8_DROP_THRESH](#)
Threshold controlling encoder frame dropping.

CBR-Specific

- #define [FE2_VP8_STREAM_INITIAL_BUFFER](#)
Seconds of preload that are necessary before starting playback.
- #define [FE2_VP8_STREAM_OPTIMAL_BUFFER](#)
Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.
- #define [FE2_VP8_STREAM_MAX_BUFFER](#)
The maximum size of the buffer, in seconds.

VBR-Specific

- #define [FE2_VP8_2PASS_MIN_SECTION](#)
[VBR_2PASSControl](#) minimum section datarate

- `#define FE2_VP8_2PASS_MAX_SECTION`
VBR_2PASSControl maximum section datarate

12.16.1 Detailed Description

The VP8 video codec.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flx, FE2_CODEC_VP8);
// Use 450kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP8_BITRATE, 450.0);
```

12.16.2 Define Documentation

12.16.2.1 `#define FE2_CODEC_VP8`

Codec name for use with [Flix2_AddCodec\(\)](#).

Definition at line 40 of file vp8.h.

12.16.2.2 `#define FE2_VP8_2PASS_MAX_SECTION`

[VBR_2PASSControl](#) maximum section datarate This value is given as a percentage. The highest datarate, i.e., $FE2_VP8_2PASS_MAX_SECTION\% \times FE2_VP8_BITRATE$, that can be streamed, and also the highest datarate that the encoder will allow, no matter how difficult the section is.

Note:

Default: 400

Definition at line 198 of file vp8.h.

12.16.2.3 `#define FE2_VP8_2PASS_MIN_SECTION`

[VBR_2PASSControl](#) minimum section datarate This value is given as a percentage. The lowest datarate, i.e., $FE2_VP8_2PASS_MIN_SECTION\% \times FE2_VP8_BITRATE$, that the encoder will allow for any section, no matter how easy the section is. This value is used to prevent difficult sections from stealing too many bits from easy sections.

Note:

Default: 40

Definition at line 188 of file vp8.h.

12.16.2.4 `#define FE2_VP8_ALTREF`

Enable the use of alternate reference frames.

Note:

Default: 0

Definition at line 211 of file vp8.h.

12.16.2.5 #define FE2_VP8_AR_MAX_FRAMES

Max number of frames blurred creating alternate reference.

Note:

Valid range: [0,25]

Default: 0

Definition at line 216 of file vp8.h.

12.16.2.6 #define FE2_VP8_AR_STRENGTH

Filter strength for the alternate reference.

Note:

Valid range: [0,6]

Default: 0

Definition at line 225 of file vp8.h.

12.16.2.7 #define FE2_VP8_AR_TYPE

Filter type to use w/alternate reference.

Note:

Default: 0

Definition at line 220 of file vp8.h.

12.16.2.8 #define FE2_VP8_BITRATE

Alias for [FE2_CODECPARAM_BITRATE](#). Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 52 of file vp8.h.

12.16.2.9 #define FE2_VP8_CXMODE

VP8 compress mode. Valid values are defined by [FE2_CompressMode](#)

Note:

Default: [COMPRESSMODE_GOOD](#)

Definition at line 77 of file vp8.h.

12.16.2.10 #define FE2_VP8_DROP_THRESH

Threshold controlling encoder frame dropping. A value of 0 disables frame dropping. Larger values will increase the likelihood frames will be dropped to achieve data rate constraints. Recommended values are 0 for VBR mode and 70 for CBR mode.

Note:

Valid range: [0,100]

Default: 0

Definition at line 151 of file vp8.h.

12.16.2.11 #define FE2_VP8_KFFREQ

Alias for [FE2_VCODECPARAM_KFFREQ](#). Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of [FE2_VCODECPARAM_KFINTTYPE](#)

See also:

[FE2_VideoKeyframeTypes](#)

Note:

Default: $12.0 \times output_{fps}$ or 360 frames if the framerate is unknown

Definition at line 65 of file vp8.h.

12.16.2.12 #define FE2_VP8_KFINTTYPE

Alias for [FE2_VCODECPARAM_KFINTTYPE](#). Video codec parameter for the keyframe interval type.

Valid values are defined by [FE2_VideoKeyframeTypes](#).

Note:

Default: [MAX_KEYFRAMES](#)

Definition at line 58 of file vp8.h.

12.16.2.13 #define FE2_VP8_LAG

Allow lagged encoding If set, this value allows the encoder to consume a number of input frames before producing output frames. This allows the encoder to base decisions for the current frame on future frames.

Note:

Valid range: [0,25]
Default: 0

Definition at line 245 of file vp8.h.

12.16.2.14 #define FE2_VP8_MAX_Q

Codec parameter for maximum quantizer. Determines the quality of the output. A lower maximum number produces higher quality output.

Note:

Valid range: [0,63]
Default: 63

Attention:

Setting this value too low will likely cause the encoder to miss (i.e., overshoot) the target datarate specified by [FE2_VP8_BITRATE](#) and, should [FE2_VP8_DROP_THRESH](#) be enabled, result in a large amount of dropped frames. The table mentioned above should be used as a guide in choosing a reasonable value based on the bitrate.

Definition at line 140 of file vp8.h.

12.16.2.15 #define FE2_VP8_MB_STATIC_THRESHOLD

Threshold for macroblocks treated static.

Note:

Default: 0

Definition at line 229 of file vp8.h.

12.16.2.16 #define FE2_VP8_MIN_Q

Codec parameter for minimum quantizer. Determines the quality of the output. A lower minimum number produces higher quality output.

Note:

Valid range: [0,63]
Default: 4

Definition at line 126 of file vp8.h.

12.16.2.17 #define FE2_VP8_NOISE_REDUCTION

Controls encoder noise reduction preprocessing. Determines the level of noise filtering to apply in the preprocessor. 0 is no preprocessing, 6 is extreme preprocessing.

Note:

Valid range: [0,6]
Setting this value to anything but 0 will result in slowing down the compression speed.
Default:

Definition at line 98 of file vp8.h.

12.16.2.18 #define FE2_VP8_OVERSHOOT_PCT

Codec parameter for undershoot percentage. This value is given as a percentage.

Note:

Default: 200

Definition at line 118 of file vp8.h.

12.16.2.19 #define FE2_VP8_PROFILE

Bitstream profile. Larger values may be more appropriate for HD material when targeting playback on lower end machines. Similar to [VP6_S](#)

Note:

Valid range: [0,3]
Default: 0

Definition at line 207 of file vp8.h.

12.16.2.20 #define FE2_VP8_RC_MODE

Alias for [FE2_VCODECPARAM_RC_MODE](#). Video codec parameter for the rate control mode.

Valid values are defined by [FE2_VideoBitrateControls](#).

Note:

Default: [VBR_2PASSControl](#)

Definition at line 71 of file vp8.h.

12.16.2.21 #define FE2_VP8_SHARPNESS

Codec parameter for sharpness. Controls the sharpness of the image in the output. This setting does not impact any other setting and is largely a matter of personal preference. A low sharpness setting will result in fewer visible artifacts but may blur the image somewhat; a high sharpness will result in a sharper image but may result in more visible artifacts.

Note:

Valid range: [0,7]
Default: 0

Definition at line 88 of file vp8.h.

12.16.2.22 #define FE2_VP8_STREAM_INITIAL_BUFFER

Seconds of preload that are necessary before starting playback. The buffer is used to maintain a consistent datarate and minimize playback interruption.

Note:

Default: 6

Attention:

Valid for CBR only

Definition at line 163 of file vp8.h.

12.16.2.23 #define FE2_VP8_STREAM_MAX_BUFFER

The maximum size of the buffer, in seconds.

Attention:

Valid for CBR only

Definition at line 173 of file vp8.h.

12.16.2.24 #define FE2_VP8_STREAM_OPTIMAL_BUFFER

Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

Note:

Default: 10

Attention:

Valid for CBR only

Definition at line 169 of file vp8.h.

12.16.2.25 #define FE2_VP8_THREADS

Number of threads to use A reasonable selection would be the number of cores on the system.

Note:

Default: 1

Definition at line 250 of file vp8.h.

12.16.2.26 #define FE2_VP8_TOKEN_PARTITIONS

Number of token partitions. This defines VP8 partitioning mode for compressed data, i.e., the number of sub-streams in the bitstream. Used for parallelized decoding.

Note:

Valid range: {1,2,4,8}
Default: 1

Definition at line 237 of file vp8.h.

12.16.2.27 #define FE2_VP8_UNDERSHOOT_PCT

Codec parameter for undershoot percentage. This value is given as a percentage. Creates output that targets a slightly lower datarate so there are bits available in the buffer to improve difficult sections.

Note:

Default: 95

Definition at line 112 of file vp8.h.

12.17 Encoding Statistics

Functions

- `on2sc_encoding_status_GetAverageBitrate` (const `FLIX2HANDLE` `flx`, `int32_t` `*pBitrate`)
Retrieve the encoder's average (video) bitrate.
- `on2sc_encoding_status_GetAverageFramesize` (const `FLIX2HANDLE` `flx`, `int32_t` `*pFramesize`)
Retrieve the encoder's average (video) frame size.
- `on2sc_encoding_status_GetMaximumFramesize` (const `FLIX2HANDLE` `flx`, `int32_t` `*pMaxFramesize`)
Retrieve the encoder's maximum (video) frame size.
- `on2sc_encoding_status_GetMinimumFramesize` (const `FLIX2HANDLE` `flx`, `int32_t` `*pMinFramesize`)
Retrieve the encoder's minimum (video) frame size.
- `on2sc_encoding_status_GetTotalFrames` (const `FLIX2HANDLE` `flx`, `int32_t` `*pTotalFrames`)
Retrieve the total number of (video) frames encoded.
- `on2sc_encoding_status_GetElapsedTime` (const `FLIX2HANDLE` `flx`, `int32_t` `*pElapsedTime`)
Retrieve the total elapsed encode time.
- `on2sc_encoding_status_GetEndTime` (const `FLIX2HANDLE` `flx`, `int32_t` `*pEndTime`)
Retrieve the encode completion time.
- `on2sc_encoding_status_GetStartTime` (const `FLIX2HANDLE` `flx`, `int32_t` `*pStartTime`)
Retrieve the encode start time.
- `on2sc_encoding_status_PercentComplete` (const `FLIX2HANDLE` `flx`, `int32_t` `*percent`)
Retrieve the percent of the encode complete.

12.17.1 Function Documentation

12.17.1.1 `on2sc_encoding_status_GetAverageBitrate` (const `FLIX2HANDLE` `flx`, `int32_t` `*pBitrate`)

Retrieve the encoder's average (video) bitrate. Value is in bits per second (bps).

Parameters:

- ← `flx` Handle to the flx engine returned from `Flix2_Create()` or `Flix2_CreateEx()`
- `pBitrate` Storage location to receive the result

Return values:

`ON2_OK` on success

`ON2_INVALID_PARAMS` should one or more of the preconditions fail

Precondition:

flx is not NULL
pBitrate is not NULL

12.17.1.2 on2sc encoding_status_GetAverageFramesize (const FLIX2HANDLE *flx*, int32_t * *pFramesize*)

Retrieve the encoder's average (video) frame size. Value is in bytes.

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *pFramesize* Storage location to receive the result

Return values:

[ON2_OK](#) on success
[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flx is not NULL
pFramesize is not NULL

12.17.1.3 on2sc encoding_status_GetElapsedTime (const FLIX2HANDLE *flx*, int32_t * *pElapsedTime*)

Retrieve the total elapsed encode time. Value is in milliseconds.

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *pElapsedTime* Storage location to receive the result

Return values:

[ON2_OK](#) on success
[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flx is not NULL
pElapsedTime is not NULL

12.17.1.4 on2sc encoding_status_GetEndTime (const FLIX2HANDLE *flx*, int32_t * *pEndTime*)

Retrieve the encode completion time. Value is in milliseconds.

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pEndTime* Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flx is not NULL

pEndTime is not NULL

12.17.1.5 on2sc encoding_status_GetMaximumFramesize (const FLIX2HANDLE *flx*, int32_t * *pMaxFramesize*)

Retrieve the encoder's maximum (video) frame size. Value is in bytes.

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pMaxFramesize* Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flx is not NULL

pMaxFramesize is not NULL

12.17.1.6 on2sc encoding_status_GetMinimumFramesize (const FLIX2HANDLE *flx*, int32_t * *pMinFramesize*)

Retrieve the encoder's minimum (video) frame size. Value is in bytes

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pMinFramesize* Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flx is not NULL

pMinFramesize is not NULL

12.17.1.7 on2sc encoding_status_GetStartTime (const FLIX2HANDLE *flix*, int32_t * *pStartTime*)

Retrieve the encode start time. Value is in milliseconds.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pStartTime* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pStartTime* is not NULL

12.17.1.8 on2sc encoding_status_GetTotalFrames (const FLIX2HANDLE *flix*, int32_t * *pTotalFrames*)

Retrieve the total number of (video) frames encoded.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pTotalFrames* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pTotalFrames* is not NULL

12.17.1.9 on2sc encoding_status_PercentComplete (const FLIX2HANDLE *flix*, int32_t * *percent*)

Retrieve the percent of the encode complete. Percent is returned as a whole number, e.g. 45.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *percent* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL
percent is not NULL

Note:

If [Flix2_GetSourceDuration\(\)](#) returns -1, percent complete will remain 0 until the first pass of the encode completes.

This value is only valid after [Flix2_Encode\(\)](#) has been called.

12.18 Filters

Modules

- [Video Filters](#)
- [Audio Filters](#)
- [Cut](#)

12.18.1 Detailed Description

The Flix API supports a variety of filters. Filters are used to apply user controllable effects to the source material. For example, these include resizing, color adjustment, frame rate conversion, cuts, overlays, etc. All effect filters in Flix will be configured using this API. In a future release, all existing features will be available via this interface as well.

For those filters which are configurable both by this interface and individual functions (e.g., `video_options_SetXXX`), this method is preferred.

Filters are generally processed in a fixed order, regardless of the order they are loaded in an encoding session. The order is:

- Video Filters
 - Preprocessing Filters
 1. [Cut](#)
 - Processing Filters
 1. [Crop](#)
 2. [Deinterlace](#)
 3. [Framerate](#)
 4. [Denoise](#)
 5. [Blur/Sharpen](#)
 6. [Mirror](#)
 7. [Rotate](#)
 8. [Scale](#)
 9. [Brightness/Contrast/Hue/Saturation](#)
 - Postprocessing Filters
 1. [Overlay](#)
 2. [PNG Exporter](#)
- Audio Filters
 - Preprocessing Filters
 1. [Cut](#)
 - Processing Filters
 1. [Lowpass/Highpass](#)
 2. [Resample](#)

12.19 Video Filters

Modules

- [Deinterlace](#)
- [Brightness/Contrast/Hue/Saturation](#)
- [Blur](#)
- [Crop](#)
- [Denoise](#)
- [Frame Rate](#)
- [Mirror](#)
- [Overlay \(Watermark\)](#)
- [PNG Image Export \(Thumbnail\)](#)
- [Rotate](#)
- [Scale](#)
- [Sharpen](#)

12.20 Audio Filters

Modules

- [Highpass](#)
- [Lowpass](#)
- [Resample](#)

12.21 Deinterlace

Defines

- `#define FE2_FILTER_ADAPTIVE_DEINTERLACE`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_ADAPTIVE_DEINTERLACE_MODE`
Specifies deinterlace mode to be applied to source image.

Typedefs

- `typedef enum deinterlacemode deintmode_t`

Enumerations

- `enum deinterlacemode {`
 `DEINTERLACE_NONE,`
 `DEINTERLACE_1_2_1_BLUR,`
 `DEINTERLACE_DROP_FIELD,`
 `DEINTERLACE_ADAPTIVE }`

Deprecated functions

- `on2sc video_options_GetDeinterlace` (const `FLIX2HANDLE` flix, `on2bool` *lpDeinterlace)
Determine if the deinterlace filter is enabled.
- `on2sc video_options_SetDeinterlace` (`FLIX2HANDLE` flix, const `on2bool` lDeinterlace)
Enable/disable the deinterlace filter.

12.21.1 Detailed Description

The adaptive deinterlace filter implements a deinterlacer with three different modes - 1:2:1 blur, drop field, and adaptive

Filter Parameters:

Name	Type	Opt/Reqd	Range
=====			=====
FE2_ADAPTIVE_DEINTERLACE_MODE	Numeric	Required	deintmode_t

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_ADAPTIVE_DEINTERLACE);
// use adaptive mode
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_ADAPTIVE_DEINTERLACE_MODE,
        DEINTERLACE_ADAPTIVE);
```

Additional References:

Deinterlace entry in the [Wikipedia](#)

12.21.2 Define Documentation**12.21.2.1 #define FE2_ADAPTIVE_DEINTERLACE_MODE**

Specifies deinterlace mode to be applied to source image. The mode may be selected from [deintmode_t](#).

Note:

Default: [DEINTERLACE_NONE](#)

Definition at line 65 of file adaptive_deinterlace.h.

12.21.2.2 #define FE2_FILTER_ADAPTIVE_DEINTERLACE

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 58 of file adaptive_deinterlace.h.

12.21.3 Typedef Documentation**12.21.3.1 typedef enum deinterlacemode deintmode_t****12.21.4 Enumeration Type Documentation****12.21.4.1 enum deinterlacemode****Enumerator:**

DEINTERLACE_NONE
DEINTERLACE_1_2_1_BLUR
DEINTERLACE_DROP_FIELD
DEINTERLACE_ADAPTIVE

Definition at line 50 of file adaptive_deinterlace.h.

12.21.5 Function Documentation**12.21.5.1 on2sc video_options_GetDeinterlace (const FLIX2HANDLE *flix*, on2bool *
lpDeinterlace)**

Determine if the deinterlace filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *lpDeinterlace* Variable to update with the current deinterlace enable status

Return values:

[ON2_OK](#) The deinterlace filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_ADAPTIVE_DEINTERLACE](#). This function will be removed in a future release.

12.21.5.2 on2sc video_options_SetDeinterlace (FLIX2HANDLE *flix*, const on2bool *lDeinterlace*)

Enable/disable the deinterlace filter.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *lDeinterlace* New deinterlace enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The deinterlace enable status was successfully set in the engine.

Note:

By default, the deinterlace filter is disabled.

Using this function will force the use of 1:2:1 deinterlace.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_ADAPTIVE_DEINTERLACE](#). This function will be removed in a future release.

12.22 Brightness/Contrast/Hue/Saturation

Defines

- `#define FE2_FILTER_BCHS`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_BCHS_BRIGHTNESS`
Parameter for the brightness adjustment factor.
- `#define FE2_BCHS_CONTRAST`
Parameter for the contrast adjustment factor.
- `#define FE2_BCHS_HUE`
Parameter for the hue adjustment factor.
- `#define FE2_BCHS_SATURATION`
Parameter for the saturation adjustment factor.

Deprecated functions

- `on2sc editor_options_GetBrightness` (const `FLIX2HANDLE` flix, `int32_t` *pBrightness)
Get the current brightness adjustment factor.
- `on2sc editor_options_SetBrightness` (`FLIX2HANDLE` flix, const `int32_t` brightness)
Set the brightness adjustment factor.
- `on2sc editor_options_GetUseBrightness` (const `FLIX2HANDLE` flix, `on2bool` *pUseBrightness)
Determine if the brightness filter is enabled.
- `on2sc editor_options_SetUseBrightness` (`FLIX2HANDLE` flix, const `on2bool` bUseBrightness)
Enable/disable the brightness filter.
- `on2sc editor_options_GetContrast` (const `FLIX2HANDLE` flix, double *pContrast)
Get the current contrast adjustment factor.
- `on2sc editor_options_SetContrast` (`FLIX2HANDLE` flix, const double contrast)
Set the contrast adjustment factor.
- `on2sc editor_options_GetUseContrast` (const `FLIX2HANDLE` flix, `on2bool` *pUseContrast)
Determine if the contrast filter is enabled.
- `on2sc editor_options_SetUseContrast` (`FLIX2HANDLE` flix, const `on2bool` bUseContrast)
Enable/disable the contrast filter.
- `on2sc editor_options_GetHue` (const `FLIX2HANDLE` flix, `int32_t` *pHue)
Get the current hue adjustment factor.

- `on2sc editor_options_SetHue` (FLIX2HANDLE flx, const int32_t hue)
Set the hue adjustment factor.
- `on2sc editor_options_GetUseHue` (const FLIX2HANDLE flx, on2bool *pUseHue)
Determine if the hue filter is enabled.
- `on2sc editor_options_SetUseHue` (FLIX2HANDLE flx, const on2bool bUseHue)
Enable/disable the hue filter.
- `on2sc editor_options_GetSaturation` (const FLIX2HANDLE flx, double *pSaturation)
Get the current saturation adjustment factor.
- `on2sc editor_options_SetSaturation` (FLIX2HANDLE flx, const double saturation)
Set the saturation adjustment factor.
- `on2sc editor_options_GetUseSaturation` (const FLIX2HANDLE flx, on2bool *pUseSaturation)
Determine if the saturation filter is enabled.
- `on2sc editor_options_SetUseSaturation` (FLIX2HANDLE flx, const on2bool bUseSaturation)
Enable/disable the saturation filter.

12.22.1 Detailed Description

The BCHS video filter is a filter used to modify the brightness, contrast, hue and/or saturation of the source image.

Filter Parameters:

Name	Type	Opt/Reqd	Range
=====			
FE2_BCHS_BRIGHTNESS	Numeric	Optional	[-255,255]
FE2_BCHS_CONTRAST	Numeric	Optional	[-255,255]
FE2_BCHS_HUE	Numeric	Optional	[-180,180]
FE2_BCHS_SATURATION	Numeric	Optional	[-255,255]

Example Usage:

```
sc = Flix2_AddFilter(&filter, flx, FE2_FILTER_BCHS);
//up the contrast by ~10%
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_BCHS_CONTRAST, 25.5);
```

12.22.2 Define Documentation

12.22.2.1 #define FE2_BCHS_BRIGHTNESS

Parameter for the brightness adjustment factor. Determines the level by which to adjust the brightness. Brightness is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative brightness of the video, use a positive number. To decrease the relative brightness of the video, use a negative number.

Note:

Valid range: [-255,255]

Default: 0 (no change)

Definition at line 58 of file bchs.h.

12.22.2.2 #define FE2_BCHS_CONTRAST

Parameter for the contrast adjustment factor. Determines the level by which to adjust the contrast. Contrast is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative contrast of the video, use a positive number. To decrease the relative contrast of the video, use a negative number.

Note:

Valid range: [-255,255]

Default: 0 (no change)

Definition at line 69 of file bchs.h.

12.22.2.3 #define FE2_BCHS_HUE

Parameter for the hue adjustment factor. Determines the level by which to adjust the hue. Hue is adjusted on a scale of -180 to 180, with 0 indicating no change. To adjust the hue of the video towards red, use a negative number. To adjust the hue of the video towards green, use a positive number.

Note:

Valid range: [-180,180]

Default: 0 (no change)

Definition at line 79 of file bchs.h.

12.22.2.4 #define FE2_BCHS_SATURATION

Parameter for the saturation adjustment factor. Determines the level by which to adjust the saturation. Saturation is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative saturation of the video, use a positive number. To decrease the relative saturation of the video, use a negative number.

Note:

Valid range: [-255,255] Default: 0

Definition at line 90 of file bchs.h.

12.22.2.5 #define FE2_FILTER_BCHS

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 47 of file bchs.h.

12.22.3 Function Documentation

12.22.3.1 on2sc editor_options_GetBrightness (const FLIX2HANDLE *flix*, int32_t * *pBrightness*)

Get the current brightness adjustment factor.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pBrightness* Brightness factor (-255..255)

Return values:

[ON2_OK](#) The brightness value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.22.3.2 on2sc editor_options_GetContrast (const FLIX2HANDLE *flix*, double * *pContrast*)

Get the current contrast adjustment factor.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pContrast* Contrast factor (-255..255)

Return values:

[ON2_OK](#) The contrast value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

12.22.3.3 on2sc editor_options_GetHue (const FLIX2HANDLE *flix*, int32_t * *pHue*)

Get the current hue adjustment factor.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pHue* Hue factor (-180..180)

Return values:

ON2_OK The hue value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

12.22.3.4 on2sc editor_options_GetSaturation (const FLIX2HANDLE *flix*, double * *pSaturation*)

Get the current saturation adjustment factor.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pSaturation* Saturation factor (-255..255)

Return values:

ON2_OK The saturation value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.22.3.5 on2sc editor_options_GetUseBrightness (const FLIX2HANDLE *flix*, on2bool * *pUseBrightness*)

Determine if the brightness filter is enabled.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pUseBrightness* Variable to update with the current brightness enable status

Return values:

ON2_OK The brightness filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.22.3.6 on2sc editor_options_GetUseContrast (const FLIX2HANDLE *flix*, on2bool * *pUseContrast*)

Determine if the contrast filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pUseContrast* Variable to update with the current contrast enable status

Return values:

[ON2_OK](#) The contrast filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

12.22.3.7 on2sc editor_options_GetUseHue (const FLIX2HANDLE *flix*, on2bool * *pUseHue*)

Determine if the hue filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pUseHue* Variable to update with the current hue enable status

Return values:

[ON2_OK](#) The hue filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

12.22.3.8 on2sc editor_options_GetUseSaturation (const FLIX2HANDLE *flix*, on2bool * *pUseSaturation*)

Determine if the saturation filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pUseSaturation* Variable to update with the current saturation enable status

Return values:

ON2_OK The saturation filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.22.3.9 on2sc editor_options_SetBrightness (FLIX2HANDLE *flix*, const int32_t *brightness*)

Set the brightness adjustment factor. Determines the level by which to adjust the brightness. Brightness is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative brightness of the video, use a positive number. To decrease the relative brightness of the video, use a negative number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *brightness* Brightness Factor (-255 to 255)

Return values:

ON2_OK The brightness value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Note:

This call will have no effect on the encoded video unless the brightness filter is enabled with [editor_options_SetUseBrightness\(\)](#). The default value is 0 (no change).

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.22.3.10 on2sc editor_options_SetContrast (FLIX2HANDLE *flix*, const double *contrast*)

Set the contrast adjustment factor. Determines the level by which to adjust the contrast. Contrast is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative contrast of the video, use a positive number. To decrease the relative contrast of the video, use a negative number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *contrast* Contrast Factor (-255 to 255)

Return values:

ON2_OK The contrast value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Note:

This call will have no effect on the encoded video unless the contrast filter is enabled with `editor_options_SetUseContrast()`. The default value is 1.

Deprecated

Use the [Filter Interface](#) along with `FE2_FILTER_BCHS` and the `FE2_BCHS_CONTRAST` parameter. This function will be removed in a future release.

12.22.3.11 on2sc editor_options_SetHue (FLIX2HANDLE *flix*, const int32_t *hue*)

Set the hue adjustment factor. Determines the level by which to adjust the hue. Hue is adjusted on a scale of -180 to 180, with 0 indicating no change. To adjust the hue of the video towards red, use a negative number. To adjust the hue of the video towards green, use a positive number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or `Flix2_CreateEx()`
- ← *hue* Hue Factor (-180 to 180)

Return values:

- ON2_OK** The hue value was successfully set in the engine.
- ON2_INVALID_PARAMS** The value is out of range.

Note:

This call will have no effect on the encoded video unless the hue filter is enabled with `editor_options_SetUseHue()`. The default value is 0.

Deprecated

Use the [Filter Interface](#) along with `FE2_FILTER_BCHS` and the `FE2_BCHS_HUE` parameter. This function will be removed in a future release.

12.22.3.12 on2sc editor_options_SetSaturation (FLIX2HANDLE *flix*, const double *saturation*)

Set the saturation adjustment factor. Determines the level by which to adjust the saturation. Saturation is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative saturation of the video, use a positive number. To decrease the relative saturation of the video, use a negative number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or `Flix2_CreateEx()`
- ← *saturation* Saturation Factor (-255 to 255)

Return values:

- ON2_OK** The saturation value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Note:

This call will have no effect on the encoded video unless the saturation filter is enabled with [editor_options_SetUseSaturation\(\)](#). The default value is 1.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.22.3.13 on2sc editor_options_SetUseBrightness (FLIX2HANDLE *flix*, const on2bool *bUseBrightness*)

Enable/disable the brightness filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseBrightness* New brightness enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

ON2_OK The brightness enable status was successfully set in the engine.

Note:

By default, the brightness filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.22.3.14 on2sc editor_options_SetUseContrast (FLIX2HANDLE *flix*, const on2bool *bUseContrast*)

Enable/disable the contrast filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseContrast* New contrast enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

ON2_OK The contrast enable status was successfully set in the engine.

Note:

By default, the contrast filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

12.22.3.15 on2sc editor_options_SetUseHue (FLIX2HANDLE *flix*, const on2bool *bUseHue*)

Enable/disable the hue filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseHue* New hue enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The hue enable status was successfully set in the engine.

Note:

By default, the hue filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

12.22.3.16 on2sc editor_options_SetUseSaturation (FLIX2HANDLE *flix*, const on2bool *bUseSaturation*)

Enable/disable the saturation filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseSaturation* New saturation enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The saturation enable status was successfully set in the engine.

Note:

By default, the saturation filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.23 Blur

Defines

- #define [FE2_FILTER_BLUR](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_BLUR_FILTER](#)
Specifies blur filter to be applied to source.
- #define [FE2_BLUR_MASKSIZE](#)
Specifies the kernel/mask size to be used.

Typedefs

- typedef enum [masksiz](#) [masksiz_t](#)
Filter mask/kernel size.
- typedef enum [blurfilter](#) [blurfilter_t](#)

Enumerations

- enum [masksiz](#) {
 [MASK_3x3](#),
 [MASK_5x5](#) }
Filter mask/kernel size.
- enum [blurfilter](#) {
 [BLUR_LOWPASS](#),
 [BLUR_GAUSS](#) }

12.23.1 Detailed Description

Blurs the source image

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_BLUR_FILTER	Numeric	Optional	blurfilter_t
FE2_BLUR_MASKSIZE	Numeric	Optional	masksiz_t

Example Usage:

```
sc = Flix2_AddFilter(&filter, flx, FE2_FILTER_BLUR);

//blur the image using a Gaussian kernel
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(&filter, FE2_BLUR_FILTER, BLUR_GAUSS);
```

12.23.2 Define Documentation

12.23.2.1 #define FE2_BLUR_FILTER

Specifies blur filter to be applied to source. The blur filter may be selected from [blurfilter_t](#).

Note:

Default: [BLUR_LOWPASS](#)

Definition at line 88 of file blur.h.

12.23.2.2 #define FE2_BLUR_MASKSIZE

Specifies the kernel/mask size to be used. The mask size may be selected from [masksiz_t](#).

Note:

Default: [MASK_3x3](#)

See also:

[blurfilter_t](#) for kernel details

Definition at line 95 of file blur.h.

12.23.2.3 #define FE2_FILTER_BLUR

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 82 of file blur.h.

12.23.3 Typedef Documentation

12.23.3.1 typedef enum blurfilter blurfilter_t

12.23.3.2 typedef enum masksiz masksiz_t

Filter mask/kernel size.

12.23.4 Enumeration Type Documentation

12.23.4.1 enum blurfilter

Enumerator:

BLUR_LOWPASS Filter image using a lowpass kernel.

```

1/9 1/9 1/9
MASK_3x3: 1/9 1/9 1/9
1/9 1/9 1/9

1/25 1/25 1/25 1/25 1/25
1/25 1/25 1/25 1/25 1/25
MASK_5x5: 1/25 1/25 1/25 1/25 1/25
1/25 1/25 1/25 1/25 1/25
1/25 1/25 1/25 1/25 1/25
```

BLUR_GAUSS Filter image using a Gaussian kernel.

```

                1/16 2/16 1/16
MASK_3x3:      2/16 4/16 2/16
                1/16 2/16 1/16

                2/571 7/571 12/571 7/571 2/571
                7/571 31/571 52/571 31/571 7/571
MASK_5x5:      12/571 52/571 127/571 52/571 12/571
                7/571 31/571 52/571 31/571 7/571
                2/571 7/571 12/571 7/571 2/571

```

Definition at line 52 of file blur.h.

12.23.4.2 enum masksize

Filter mask/kernel size.

Enumerator:

MASK_3x3 kernel/mask will be a matrix of size 3x3

MASK_5x5 kernel/mask will be a matrix of size 5x5

Definition at line 47 of file blur.h.

12.24 Crop

Defines

- `#define FE2_FILTER_CROP`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_CROP_TOP`
Filter parameter for setting bounding box's top coordinate.
- `#define FE2_CROP_BOTTOM`
Filter parameter for setting bounding box's bottom coordinate.
- `#define FE2_CROP_LEFT`
Filter parameter for setting bounding box's left coordinate.
- `#define FE2_CROP_RIGHT`
Filter parameter for setting bounding box's right coordinate.

Deprecated functions

- `on2sc editor_options_GetCrop` (const `FLIX2HANDLE` flx, `on2bool` *pCrop)
Determine if the crop filter is enabled.
- `on2sc editor_options_SetCrop` (`FLIX2HANDLE` flx, const `on2bool` crop)
Enable/disable the crop filter.
- `on2sc editor_options_GetCropBounds` (const `FLIX2HANDLE` flx, `int32_t` *pTop, `int32_t` *pLeft, `int32_t` *pBottom, `int32_t` *pRight)
Get the current bounding box used for cropping.
- `on2sc editor_options_SetCropBounds` (`FLIX2HANDLE` flx, const `int32_t` top, const `int32_t` left, const `int32_t` bottom, const `int32_t` right)
Set the current bounding box used for cropping.
- `on2sc editor_options_GetBrightness` (const `FLIX2HANDLE` flx, `int32_t` *pBrightness)
Get the current brightness adjustment factor.
- `on2sc editor_options_SetBrightness` (`FLIX2HANDLE` flx, const `int32_t` brightness)
Set the brightness adjustment factor.
- `on2sc editor_options_GetUseBrightness` (const `FLIX2HANDLE` flx, `on2bool` *pUseBrightness)
Determine if the brightness filter is enabled.
- `on2sc editor_options_SetUseBrightness` (`FLIX2HANDLE` flx, const `on2bool` bUseBrightness)
Enable/disable the brightness filter.
- `on2sc editor_options_GetContrast` (const `FLIX2HANDLE` flx, double *pContrast)

Get the current contrast adjustment factor.

- [on2sc editor_options_SetContrast](#) (FLIX2HANDLE flx, const double contrast)
Set the contrast adjustment factor.
- [on2sc editor_options_GetUseContrast](#) (const FLIX2HANDLE flx, on2bool *pUseContrast)
Determine if the contrast filter is enabled.
- [on2sc editor_options_SetUseContrast](#) (FLIX2HANDLE flx, const on2bool bUseContrast)
Enable/disable the contrast filter.
- [on2sc editor_options_GetHue](#) (const FLIX2HANDLE flx, int32_t *pHue)
Get the current hue adjustment factor.
- [on2sc editor_options_SetHue](#) (FLIX2HANDLE flx, const int32_t hue)
Set the hue adjustment factor.
- [on2sc editor_options_GetUseHue](#) (const FLIX2HANDLE flx, on2bool *pUseHue)
Determine if the hue filter is enabled.
- [on2sc editor_options_SetUseHue](#) (FLIX2HANDLE flx, const on2bool bUseHue)
Enable/disable the hue filter.
- [on2sc editor_options_GetSaturation](#) (const FLIX2HANDLE flx, double *pSaturation)
Get the current saturation adjustment factor.
- [on2sc editor_options_SetSaturation](#) (FLIX2HANDLE flx, const double saturation)
Set the saturation adjustment factor.
- [on2sc editor_options_GetUseSaturation](#) (const FLIX2HANDLE flx, on2bool *pUseSaturation)
Determine if the saturation filter is enabled.
- [on2sc editor_options_SetUseSaturation](#) (FLIX2HANDLE flx, const on2bool bUseSaturation)
Enable/disable the saturation filter.
- [on2sc editor_options_GetUseCut](#) (const FLIX2HANDLE flx, on2bool *pUseCut)
Determine if the cut filter is enabled.
- [on2sc editor_options_SetUseCut](#) (FLIX2HANDLE flx, const on2bool bUseCut)
Enable/disable the cut filter.
- [on2sc editor_options_GetCutStartTime](#) (const FLIX2HANDLE flx, double *pStartTime)
Get the current cut start time.
- [on2sc editor_options_SetCutStartTime](#) (FLIX2HANDLE flx, const double start_time)
Set the cut start time.
- [on2sc editor_options_GetCutStopTime](#) (const FLIX2HANDLE flx, double *pEndTime)
Get the current cut stop time.

- `on2sc editor_options_SetCutStopTime` (FLIX2HANDLE flx, const double end_time)

Set the cut stop time.

12.24.1 Detailed Description

The crop video filter is a filter used to isolate a subsection of an input image.

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_CROP_TOP	Numeric	Optional	[0,video height]
FE2_CROP_BOTTOM	Numeric	Optional	[0,video height]
FE2_CROP_LEFT	Numeric	Optional	[0,video width]
FE2_CROP_RIGHT	Numeric	Optional	[0,video width]

Example Usage:

```
sc = Flix2_AddFilter(&filter, flx, FE2_FILTER_CROP);
//set crop bounding box to (0,0,240,320) (t,l,b,r)
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_CROP_BOTTOM, 240);
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_CROP_RIGHT, 320);
```

12.24.2 Define Documentation

12.24.2.1 #define FE2_CROP_BOTTOM

Filter parameter for setting bounding box's bottom coordinate.

Note:

Default: input image height

Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 61 of file crop.h.

12.24.2.2 #define FE2_CROP_LEFT

Filter parameter for setting bounding box's left coordinate.

Note:

Default: 0

Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 67 of file crop.h.

12.24.2.3 #define FE2_CROP_RIGHT

Filter parameter for setting bounding box's right coordinate.

Note:

Default: input image width

Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 73 of file crop.h.

12.24.2.4 #define FE2_CROP_TOP

Filter parameter for setting bounding box's top coordinate.

Note:

Default: 0

Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 55 of file crop.h.

12.24.2.5 #define FE2_FILTER_CROP

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 49 of file crop.h.

12.24.3 Function Documentation**12.24.3.1 on2sc editor_options_GetBrightness (const FLIX2HANDLE *flix*, int32_t * *pBrightness*)**

Get the current brightness adjustment factor.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pBrightness* Brightness factor (-255..255)

Return values:

[ON2_OK](#) The brightness value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.24.3.2 on2sc editor_options_GetContrast (const FLIX2HANDLE *flix*, double * *pContrast*)

Get the current contrast adjustment factor.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pContrast* Contrast factor (-255..255)

Return values:

[ON2_OK](#) The contrast value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

12.24.3.3 on2sc editor_options_GetCrop (const FLIX2HANDLE *flix*, on2bool * *pCrop*)

Determine if the crop filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pCrop* Variable to update with the current crop enable status

Return values:

[ON2_OK](#) The crop enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#). This function will be removed in a future release.

12.24.3.4 on2sc editor_options_GetCropBounds (const FLIX2HANDLE *flix*, int32_t * *pTop*, int32_t * *pLeft*, int32_t * *pBottom*, int32_t * *pRight*)

Get the current bounding box used for cropping.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pTop* Topmost row of the cropped area
- *pLeft* Leftmost column of the cropped area
- *pBottom* Bottommost row of the cropped area
- *pRight* Rightmost row of the cropped area

Return values:

ON2_OK The crop bounding box was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#) and the [FE2_CROP_TOP](#), [FE2_CROP_LEFT](#), [FE2_CROP_BOTTOM](#) and [FE2_CROP_RIGHT](#) parameters. This function will be removed in a future release.

12.24.3.5 on2sc editor_options_GetCutStartTime (const FLIX2HANDLE *flix*, double * *pStartTime*)

Get the current cut start time.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *pStartTime* Clip start time (in seconds)

Return values:

ON2_OK The cut start time was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.24.3.6 on2sc editor_options_GetCutStopTime (const FLIX2HANDLE *flix*, double * *pEndTime*)

Get the current cut stop time.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pEndTime* Clip stop time (in seconds, -1 for end)

Return values:

ON2_OK The cut stop time was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.24.3.7 on2sc editor_options_GetHue (const FLIX2HANDLE *flix*, int32_t * *pHue*)

Get the current hue adjustment factor.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pHue* Hue factor (-180..180)

Return values:

ON2_OK The hue value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

12.24.3.8 on2sc editor_options_GetSaturation (const FLIX2HANDLE *flix*, double * *pSaturation*)

Get the current saturation adjustment factor.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pSaturation* Saturation factor (-255..255)

Return values:

ON2_OK The saturation value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.24.3.9 on2sc editor_options_GetUseBrightness (const FLIX2HANDLE *flix*, on2bool * *pUseBrightness*)

Determine if the brightness filter is enabled.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pUseBrightness* Variable to update with the current brightness enable status

Return values:

ON2_OK The brightness filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.24.3.10 on2sc editor_options_GetUseContrast (const FLIX2HANDLE *flix*, on2bool * *pUseContrast*)

Determine if the contrast filter is enabled.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pUseContrast* Variable to update with the current contrast enable status

Return values:

ON2_OK The contrast filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

12.24.3.11 on2sc editor_options_GetUseCut (const FLIX2HANDLE *flix*, on2bool * *pUseCut*)

Determine if the cut filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pUseCut* Variable to update with the current saturation enable status

Return values:

[ON2_OK](#) The cut filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.24.3.12 on2sc editor_options_GetUseHue (const FLIX2HANDLE *flix*, on2bool * *pUseHue*)

Determine if the hue filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pUseHue* Variable to update with the current hue enable status

Return values:

[ON2_OK](#) The hue filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

12.24.3.13 on2sc editor_options_GetUseSaturation (const FLIX2HANDLE *flix*, on2bool * *pUseSaturation*)

Determine if the saturation filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pUseSaturation* Variable to update with the current saturation enable status

Return values:

[ON2_OK](#) The saturation filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.24.3.14 on2sc editor_options_SetBrightness (FLIX2HANDLE *flix*, const int32_t *brightness*)

Set the brightness adjustment factor. Determines the level by which to adjust the brightness. Brightness is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative brightness of the video, use a positive number. To decrease the relative brightness of the video, use a negative number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *brightness* Brightness Factor (-255 to 255)

Return values:

[ON2_OK](#) The brightness value was successfully set in the engine.

[ON2_INVALID_PARAMS](#) The value is out of range.

Note:

This call will have no effect on the encoded video unless the brightness filter is enabled with [editor_options_SetUseBrightness\(\)](#). The default value is 0 (no change).

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.24.3.15 on2sc editor_options_SetContrast (FLIX2HANDLE *flix*, const double *contrast*)

Set the contrast adjustment factor. Determines the level by which to adjust the contrast. Contrast is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative contrast of the video, use a positive number. To decrease the relative contrast of the video, use a negative number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *contrast* Contrast Factor (-255 to 255)

Return values:

- [ON2_OK](#) The contrast value was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

This call will have no effect on the encoded video unless the contrast filter is enabled with [editor_options_SetUseContrast\(\)](#). The default value is 1.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

12.24.3.16 on2sc editor_options_SetCrop (FLIX2HANDLE *flix*, const on2bool *crop*)

Enable/disable the crop filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *crop* New crop enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

- [ON2_OK](#) The crop enable status was successfully set in the engine.

Note:

By default, the crop filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#). This function will be removed in a future release.

12.24.3.17 on2sc editor_options_SetCropBounds (FLIX2HANDLE *flix*, const int32_t *top*, const int32_t *left*, const int32_t *bottom*, const int32_t *right*)

Set the current bounding box used for cropping.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *top* Topmost row of the cropped area
- ← *left* Leftmost column of the cropped area
- ← *bottom* Bottommost row of the cropped area

← *right* Rightmost row of the cropped area

Return values:

ON2_OK The crop bounding box was successfully set in the engine.

Note:

This call will have no effect on the encoded video unless the crop filter is enabled with [editor_options_-SetCrop\(\)](#)

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_CROP](#) and the [FE2_CROP_TOP](#), [FE2_CROP_LEFT](#), [FE2_CROP_BOTTOM](#) and [FE2_CROP_RIGHT](#) parameters. This function will be removed in a future release.

12.24.3.18 on2sc editor_options_SetCutStartTime (FLIX2HANDLE *flix*, const double *start_time*)

Set the cut start time. Sets the time (in seconds from the beginning of the source) to begin the media cut. 0 indicates start immediately.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 ← *start_time* Start time (in seconds, relative to source)

Return values:

ON2_OK The start time value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Note:

This call will have no effect on the encoded video unless the cut filter is enabled with [editor_options_-SetUseCut\(\)](#). The default value is 0.0.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.24.3.19 on2sc editor_options_SetCutStopTime (FLIX2HANDLE *flix*, const double *end_time*)

Set the cut stop time. Sets the time (in seconds from the beginning of the source) to end the media cut. -1 indicates end of source.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *end_time* End time (in seconds, relative to source)

Return values:

- [ON2_OK](#) The end time value was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

This call will have no effect on the encoded video unless the cut filter is enabled with [editor_options_SetUseCut\(\)](#). The default value is -1.0 (end of source).

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.24.3.20 on2sc editor_options_SetHue (FLIX2HANDLE *flix*, const int32_t *hue*)

Set the hue adjustment factor. Determines the level by which to adjust the hue. Hue is adjusted on a scale of -180 to 180, with 0 indicating no change. To adjust the hue of the video towards red, use a negative number. To adjust the hue of the video towards green, use a positive number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *hue* Hue Factor (-180 to 180)

Return values:

- [ON2_OK](#) The hue value was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

This call will have no effect on the encoded video unless the hue filter is enabled with [editor_options_SetUseHue\(\)](#). The default value is 0.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

12.24.3.21 on2sc editor_options_SetSaturation (FLIX2HANDLE *flix*, const double *saturation*)

Set the saturation adjustment factor. Determines the level by which to adjust the saturation. Saturation is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative saturation of the video, use a positive number. To decrease the relative saturation of the video, use a negative number.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *saturation* Saturation Factor (-255 to 255)

Return values:

- [ON2_OK](#) The saturation value was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

This call will have no effect on the encoded video unless the saturation filter is enabled with [editor_options_SetUseSaturation\(\)](#). The default value is 1.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.24.3.22 on2sc editor_options_SetUseBrightness (FLIX2HANDLE *flix*, const on2bool *bUseBrightness*)

Enable/disable the brightness filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseBrightness* New brightness enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

- [ON2_OK](#) The brightness enable status was successfully set in the engine.

Note:

By default, the brightness filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_BRIGHTNESS](#) parameter. This function will be removed in a future release.

12.24.3.23 on2sc editor_options_SetUseContrast (FLIX2HANDLE *flix*, const on2bool *bUseContrast*)

Enable/disable the contrast filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseContrast* New contrast enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The contrast enable status was successfully set in the engine.

Note:

By default, the contrast filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_CONTRAST](#) parameter. This function will be removed in a future release.

12.24.3.24 on2sc editor_options_SetUseCut (FLIX2HANDLE *flix*, const on2bool *bUseCut*)

Enable/disable the cut filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseCut* New cut enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The cut enable status was successfully set in the engine.

Note:

By default, the cut filter is disabled.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.24.3.25 on2sc editor_options_SetUseHue (FLIX2HANDLE *flix*, const on2bool *bUseHue*)

Enable/disable the hue filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseHue* New hue enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The hue enable status was successfully set in the engine.

Note:

By default, the hue filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_HUE](#) parameter. This function will be removed in a future release.

12.24.3.26 on2sc editor_options_SetUseSaturation (FLIX2HANDLE *flix*, const on2bool *bUseSaturation*)

Enable/disable the saturation filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseSaturation* New saturation enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The saturation enable status was successfully set in the engine.

Note:

By default, the saturation filter is disabled.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_BCHS](#) and the [FE2_BCHS_SATURATION](#) parameter. This function will be removed in a future release.

12.25 Cut

Defines

- `#define FE2_FILTER_CUT`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_CUT_START_SEC`
Filter parameter for cut start time.
- `#define FE2_CUT_STOP_SEC`
Filter parameter for setting the cut stop time.
- `#define FE2_CUT_USE_SEEK`
Set `FE2_CUT_START_SEC` application method.

Deprecated functions

- `on2sc editor_options_GetUseCut` (const `FLIX2HANDLE` flix, `on2bool` *pUseCut)
Determine if the cut filter is enabled.
- `on2sc editor_options_SetUseCut` (`FLIX2HANDLE` flix, const `on2bool` bUseCut)
Enable/disable the cut filter.
- `on2sc editor_options_GetCutStartTime` (const `FLIX2HANDLE` flix, double *pStartTime)
Get the current cut start time.
- `on2sc editor_options_SetCutStartTime` (`FLIX2HANDLE` flix, const double start_time)
Set the cut start time.
- `on2sc editor_options_GetCutStopTime` (const `FLIX2HANDLE` flix, double *pEndTime)
Get the current cut stop time.
- `on2sc editor_options_SetCutStopTime` (`FLIX2HANDLE` flix, const double end_time)
Set the cut stop time.

12.25.1 Detailed Description

The cut filter allows for the encoding of a user specified range of time from the input media file.

Filter Parameters:

Name	Type	Opt/Reqd	Range
=====			
FE2_CUT_START_SEC	Numeric	Optional	[0, media duration]
FE2_CUT_STOP_SEC	Numeric	Optional	[-1, media duration]
FE2_CUT_USE_SEEK	Numeric	Optional	[0, 1]

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_CUT);  
// start the cut at 5 seconds  
if(sc == ON2_OK)  
    sc = Flix2_FilterSetParam(filter, FE2_CUT_START_SEC, 5.0);  
// end the cut at 10 seconds  
if (sc == ON2_OK)  
    sc = Flix2_FilterSetParam(filter, FE2_CUT_STOP_SEC, 10.0);
```

12.25.2 Define Documentation**12.25.2.1 #define FE2_CUT_START_SEC**

Filter parameter for cut start time. Sets the time (in seconds relative to the beginning of the source) to begin the media cut. A value of 0 indicates that the cut should start at the beginning of the input media.

Note:

Default: 0

Definition at line 64 of file cut.h.

12.25.2.2 #define FE2_CUT_STOP_SEC

Filter parameter for setting the cut stop time. Sets the time (in seconds relative to the beginning of the input media) to end the media cut. A value of -1 indicates the cut should stop at the end of the input media.

Note:

Default: -1

Definition at line 75 of file cut.h.

12.25.2.3 #define FE2_CUT_USE_SEEK

Set FE2_CUT_START_SEC application method. When [FE2_CUT_START_SEC](#) is non-zero this parameter determines how the cut is applied. A value of 0 indicates that the entire source file should be read and discarded until [FE2_CUT_START_SEC](#) is reached. This method will increase the time necessary to process the source media as it must be decoded before it be discarded. This was the default behavior prior to 8.0.7.0. A value of 1 indicates that the engine should attempt to perform a seek on the source media to arrive at [FE2_CUT_START_SEC](#). Should this method fail the engine will fall back to the legacy method.

Note:

Default: 1

If output is produced with the value set to 1, but the results are undesirable the only workaround is to re-encode with the value set to 0.

Definition at line 95 of file cut.h.

12.25.2.4 #define FE2_FILTER_CUT

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 52 of file cut.h.

12.25.3 Function Documentation

12.25.3.1 on2sc editor_options_GetCutStartTime (const FLIX2HANDLE *flx*, double * *pStartTime*)

Get the current cut start time.

Parameters:

- ← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *pStartTime* Clip start time (in seconds)

Return values:

[ON2_OK](#) The cut start time was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.25.3.2 on2sc editor_options_GetCutStopTime (const FLIX2HANDLE *flx*, double * *pEndTime*)

Get the current cut stop time.

Parameters:

- ← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *pEndTime* Clip stop time (in seconds, -1 for end)

Return values:

[ON2_OK](#) The cut stop time was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.25.3.3 on2sc editor_options_GetUseCut (const FLIX2HANDLE *flix*, on2bool * *pUseCut*)

Determine if the cut filter is enabled.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pUseCut* Variable to update with the current saturation enable status

Return values:

[ON2_OK](#) The cut filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.25.3.4 on2sc editor_options_SetCutStartTime (FLIX2HANDLE *flix*, const double *start_time*)

Set the cut start time. Sets the time (in seconds from the beginning of the source) to begin the media cut. 0 indicates start immediately.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *start_time* Start time (in seconds, relative to source)

Return values:

ON2_OK The start time value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Note:

This call will have no effect on the encoded video unless the cut filter is enabled with `editor_options_-SetUseCut()`. The default value is 0.0.

Deprecated

Please use:

- `Flix2_AddFilter()`
- `Flix2_FilterGetParam()`
- `Flix2_FilterSetParam()`

With the constants:

- `FE2_FILTER_CUT`
- `FE2_CUT_START_SEC`
- `FE2_CUT_STOP_SEC`

for access to the flx engine cut filter.

12.25.3.5 on2sc editor_options_SetCutStopTime (FLIX2HANDLE *flx*, const double *end_time*)

Set the cut stop time. Sets the time (in seconds from the beginning of the source) to end the media cut. -1 indicates end of source.

Parameters:

- ← *flx* Handle to the Flix Engine returned from `Flix2_Create()` or `Flix2_CreateEx()`
- ← *end_time* End time (in seconds, relative to source)

Return values:

ON2_OK The end time value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Note:

This call will have no effect on the encoded video unless the cut filter is enabled with `editor_options_-SetUseCut()`. The default value is -1.0 (end of source).

Deprecated

Please use:

- `Flix2_AddFilter()`
- `Flix2_FilterGetParam()`
- `Flix2_FilterSetParam()`

With the constants:

- `FE2_FILTER_CUT`
- `FE2_CUT_START_SEC`
- `FE2_CUT_STOP_SEC`

for access to the flx engine cut filter.

12.25.3.6 on2sc editor_options_SetUseCut (FLIX2HANDLE *flix*, const on2bool *bUseCut*)

Enable/disable the cut filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseCut* New cut enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

[ON2_OK](#) The cut enable status was successfully set in the engine.

Note:

By default, the cut filter is disabled.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_CUT](#)
- [FE2_CUT_START_SEC](#)
- [FE2_CUT_STOP_SEC](#)

for access to the flix engine cut filter.

12.26 Denoise

Defines

- `#define FE2_FILTER_DENOISE`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_DENOISE_NOISE_LEVEL`
Specifies noise level of the source.

12.26.1 Detailed Description

Removes noise from source image

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_DENOISE_NOISE_LEVEL	Numeric	Optional	[0.0,1.0)

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_DENOISE);
if(sc == ON2_OK)
    ; //adaptive noise removal will be applied to the source image
```

12.26.2 Define Documentation

12.26.2.1 `#define FE2_DENOISE_NOISE_LEVEL`

Specifies noise level of the source. Any value greater than 0 implies a constant and known level of noise within the source. A value of 0 will cause the filter to estimate the noise level, allowing it to vary from frame to frame. Larger noise levels indicate a noisier source resulting in increased removal by the filter, at the cost of sharpness.

Note:

Default: 0 (adaptive)

Definition at line 57 of file denoise.h.

12.26.2.2 `#define FE2_FILTER_DENOISE`

Filter name for use with `Flix2_AddFilter()`.

Definition at line 45 of file denoise.h.

12.27 Frame Rate

Defines

- `#define FE2_FILTER_FRAMERATE`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_FRAMERATE_FPS`
Filter parameter name for frames per second value.
- `#define FE2_FRAMERATE_DECIMATE`
Filter parameter for the decimation interval.

Deprecated functions

- `on2sc video_options_GetVideoFramerate (const FLIX2HANDLE flx, int32_t *lpVideoFramerate)`
Get the current video framerate.
- `on2sc video_options_SetVideoFramerate (FLIX2HANDLE flx, const int32_t lVideoFramerate)`
Set the video framerate.
- `on2sc video_options_GetVideoFramerateAsDouble (const FLIX2HANDLE flx, double *p_fps)`
Get the current video framerate.
- `on2sc video_options_SetVideoFramerateAsDouble (FLIX2HANDLE flx, const double fps)`
Set the video framerate.
- `on2sc video_options_GetUseSourceFramerate (FLIX2HANDLE flx, on2bool *bpUseSourceFramerate)`
Determine if the source's framerate will be used instead of the modified framerate.
- `on2sc video_options_SetUseSourceFramerate (FLIX2HANDLE flx, const on2bool bUseSourceFramerate)`
Switch between source and scaled framerate.
- `on2sc video_options_GetDecimateValue (const FLIX2HANDLE flx, uint32_t *pValue)`
Get the current decimation of the video framerate.
- `on2sc video_options_SetDecimateValue (FLIX2HANDLE flx, const uint32_t value)`
Set the decimation of the video framerate.

12.27.1 Detailed Description

The frame rate filter modifies the source frame rate either by applying a specific frame rate (e.g. 12.0) or a decimation interval (e.g. 2 = 1/2 source frame rate).

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_FRAMERATE_FPS	Numeric	Optional	[0.0,)
FE2_FRAMERATE_DECIMATE	Numeric	Optional	[1,)

Example Usage:

```

sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_FRAMERATE);
// decimate by 2. ex. 29.97fps becomes 14.985
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_FRAMERATE_DECIMATE, 2);

```

12.27.2 Define Documentation**12.27.2.1 #define FE2_FILTER_FRAMERATE**

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 49 of file framerate.h.

12.27.2.2 #define FE2_FRAMERATE_DECIMATE

Filter parameter for the decimation interval. Decimation removes frames from the source video at regular intervals. This is useful for dropping the rate without introducing jerkiness due to uneven frame rate.

Note:

Must be a positive integer
Default: disabled

Definition at line 67 of file framerate.h.

12.27.2.3 #define FE2_FRAMERATE_FPS

Filter parameter name for frames per second value. Allows the frame rate to be adjusted to a specific rate.

Note:

Default: disabled

Definition at line 56 of file framerate.h.

12.27.3 Function Documentation**12.27.3.1 on2sc video_options_GetDecimateValue (const FLIX2HANDLE *flix*, uint32_t * *pValue*)**

Get the current decimation of the video framerate.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *pValue* Decimate value

Return values:

[*ON2_OK*](#) Success.

[*ON2_INVALID_PARAMS*](#) Should one or more of the preconditions fail.

Precondition:

flix is not NULL

pValue is not NULL

Note:

If a value other than *ON2_OK* is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_DECIMATE](#) parameter. This function will be removed in a future release.

12.27.3.2 `on2sc video_options_GetUseSourceFramerate (FLIX2HANDLE flix, on2bool * bpUseSourceFramerate)`

Determine if the source's framerate will be used instead of the modified framerate.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *bpUseSourceFramerate* Variable to update with the current framerate selection

Return values:

[*ON2_OK*](#) The framerate status was successfully retrieved from the engine.

Note:

If a value other than *ON2_OK* is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#). Not adding [FE2_FILTER_FRAMERATE](#) with [Flix2_AddFilter\(\)](#) implies use of the source frame rate. This function will be removed in a future release.

12.27.3.3 `on2sc video_options_GetVideoFramerate (const FLIX2HANDLE flix, int32_t * lpVideoFramerate)`

Get the current video framerate.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *lpVideoFramerate* Frames per second

Return values:

[*ON2_OK*](#) The framerate was successfully retrieved from the engine.

Deprecated

Please use [video_options_GetVideoFramerateAsDouble\(\)](#) to allow for non integer framerates.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

12.27.3.4 on2sc video_options_GetVideoFramerateAsDouble (const FLIX2HANDLE *flix*, double * *p_fps*)

Get the current video framerate.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *p_fps* Frames per second

Return values:

[ON2_OK](#) The framerate was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_FPS](#) parameter. This function will be removed in a future release.

12.27.3.5 on2sc video_options_SetDecimateValue (FLIX2HANDLE *flix*, const uint32_t *value*)

Set the decimation of the video framerate. This value will be used to drop frames according to the video framerate. For example if the framerate is 30fps and the decimate value is 3 then the encoded framerate will be 10fps.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 ← *value* Decimate value

Return values:

[ON2_OK](#) The target framerate was successfully set in the engine.

[ON2_INVALID_PARAMS](#) Should one or more of the preconditions fail.

Precondition:

flix is not NULL

Note:

Default value is 0.

A value of ≤ 1 means to not drop any frames.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_DECIMATE](#) parameter. This function will be removed in a future release.

12.27.3.6 on2sc video_options_SetUseSourceFramerate (FLIX2HANDLE *flix*, const on2bool *bUseSourceFramerate*)

Switch between source and scaled framerate.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseSourceFramerate* New framerate status. [on2true](#) to use source framerate, [on2false](#) to use scaled framerate.

Return values:

[ON2_OK](#) The framerate status was successfully set in the engine.

Note:

By default, the source framerate will be used, unless a call to [video_options_SetVideoFramerate\(\)](#), [video_options_SetVideoFramerateAsDouble\(\)](#) or [video_options_SetDecimateValue\(\)](#) has been made.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#). Not adding [FE2_FILTER_FRAMERATE](#) with [Flix2_AddFilter\(\)](#) implies use of the source frame rate. This function will be removed in a future release.

12.27.3.7 on2sc video_options_SetVideoFramerate (FLIX2HANDLE *flix*, const int32_t *IVideoFramerate*)

Set the video framerate. Sets the framerate to be used when encoding the video. Frames will be duplicated or dropped as necessary to achieve the desired framerate.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *IVideoFramerate* Framerate target (in frames per second)

Return values:

[ON2_OK](#) The target framerate was successfully set in the engine.

[ON2_INVALID_PARAMS](#) The value is out of range.

Deprecated

Please use [video_options_SetVideoFramerateAsDouble\(\)](#) to allow for non integer framerates.

Note:

The default is to use the source framerate.

12.27.3.8 on2sc video_options_SetVideoFramerateAsDouble (FLIX2HANDLE *flix*, const double *fps*)

Set the video framerate. Sets the framerate to be used when encoding the video. Frames will be duplicated or dropped as necessary to achieve the desired framerate.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *fps* Framerate target (in frames per second)

Return values:

- [ON2_OK](#) The target framerate was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

The default is to use the source framerate.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_FPS](#) parameter. This function will be removed in a future release.

12.28 Highpass

Defines

- `#define FE2_FILTER_HIGHPASS`
Filter name for use with [Flix2_AddFilter\(\)](#).
- `#define FE2_HIGHPASS_Q`
Filter parameter for shape constant ("Q" coefficient).
- `#define FE2_HIGHPASS_CUTOFF`
Filter parameter for cutoff frequency.

12.28.1 Detailed Description

The highpass audio filter is a filter to attenuate sounds in the audio track that are lower than the cutoff frequency. In other words, high frequencies are passed by the filter, and low frequencies are stopped.

Example Usage:

```
sc = Flix2_AddFilter(&filter, flx, FE2_FILTER_HIGHPASS);
// Cut off frequencies below 2khz
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_HIGHPASS_CUTOFF, 2000.0);
```

Additional References:

[Cookbook formulae for audio EQ biquad filter coefficients by Robert Bristow-Johnson](#)

12.28.2 Define Documentation

12.28.2.1 `#define FE2_FILTER_HIGHPASS`

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 46 of file `highpass.h`.

12.28.2.2 `#define FE2_HIGHPASS_CUTOFF`

Filter parameter for cutoff frequency. This is the filter's "corner" frequency. Components of the sound track with frequencies lower than this frequency will be attenuated. If very precise control of the cutoff frequency is required, note that the response of the filter at this frequency is -3db. Note that the maximum frequency is always half of the sampling rate of the produced file (5512.5/11025/22050 for sample rates [Hertz11025/Hertz22050/Hertz44100](#) respectively).

Definition at line 69 of file `highpass.h`.

12.28.2.3 `#define FE2_HIGHPASS_Q`

Filter parameter for shape constant ("Q" coefficient). The Q coefficient controls the "shape" of the filter. In general, higher numbers mean a sharper response curve. The default value is 0.707, and should be

appropriate in most cases. If you feel the default value is inadequate, iterative experimentation is the best way to select a new value. Alternatively, a mathematical description of this filter is linked in the references above.

Definition at line 57 of file `highpass.h`.

12.29 Lowpass

Defines

- `#define FE2_FILTER_LOWPASS`
Filter name for use with [Flix2_AddFilter\(\)](#).
- `#define FE2_LOWPASS_Q`
Filter parameter for shape constant ("Q" coefficient).
- `#define FE2_LOWPASS_CUTOFF`
Filter parameter for cutoff frequency.

12.29.1 Detailed Description

The lowpass audio filter is a filter to attenuate sounds in the audio track that are higher than the cutoff frequency. In other words, low frequencies are passed by the filter, and high frequencies are stopped.

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_LOWPASS);
// Cut off frequencies above 10khz
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_LOWPASS_CUTOFF, 10000.0);
```

Additional References:

[Cookbook formulae for audio EQ biquad filter coefficients by Robert Bristow-Johnson](#)

12.29.2 Define Documentation

12.29.2.1 `#define FE2_FILTER_LOWPASS`

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 46 of file lowpass.h.

12.29.2.2 `#define FE2_LOWPASS_CUTOFF`

Filter parameter for cutoff frequency. This is the filter's "corner" frequency. Components of the sound track with frequencies higher than this frequency will be attenuated. If very precise control of the cutoff frequency is required, note that the response of the filter at this frequency is -3db. Note that the maximum frequency is always half of the sampling rate of the produced file (5512.5/11025/22050 for sample rates [Hertz11025/Hertz22050/Hertz44100](#) respectively).

Definition at line 69 of file lowpass.h.

12.29.2.3 `#define FE2_LOWPASS_Q`

Filter parameter for shape constant ("Q" coefficient). The Q coefficient controls the "shape" of the filter. In general, higher numbers mean a sharper response curve. The default value is 0.707, and should be

appropriate in most cases. If you feel the default value is inadequate, iterative experimentation is the best way to select a new value. Alternatively, a mathematical description of this filter is linked in the references above.

Definition at line 57 of file lowpass.h.

12.30 Mirror

Defines

- `#define FE2_FILTER_MIRROR`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_MIRROR_HORIZONTAL`
Specifies horizontal disposition.
- `#define FE2_MIRROR_VERTICAL`
Specifies vertical disposition.

12.30.1 Detailed Description

The mirror filter mirrors (flips) the source along the horizontal axis, vertical axis or both.

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_MIRROR_HORIZONTAL	Numeric	Optional	{on2false,on2true}
FE2_MIRROR_VERTICAL	Numeric	Optional	{on2false,on2true}

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_MIRROR);
// flip the image horizontally
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_MIRROR_HORIZONTAL, on2true);
```

Note:

Setting both `FE2_MIRROR_HORIZONTAL` and `FE2_MIRROR_VERTICAL` is equivalent to setting `FE2_ROTATE_ANGLE` to 180

See also:

`FE2_FILTER_ROTATE`

12.30.2 Define Documentation

12.30.2.1 `#define FE2_FILTER_MIRROR`

Filter name for use with `Flix2_AddFilter()`.

Definition at line 49 of file mirror.h.

12.30.2.2 `#define FE2_MIRROR_HORIZONTAL`

Specifies horizontal disposition. Mirror (flip) the image along the horizontal axis

Note:

Default: 0 (disabled)
non-zero: enabled

Definition at line 56 of file mirror.h.

12.30.2.3 #define FE2_MIRROR_VERTICAL

Specifies vertical disposition. Mirror (flip) the image along the vertical axis

Note:

Default: 0 (disabled)
non-zero: enabled

Definition at line 63 of file mirror.h.

12.31 Overlay (Watermark)

Defines

- `#define FE2_FILTER_OVERLAY`
Filter name for use with `Flix2_AddFilter()`.
- `#define FE2_OVERLAY_FILE`
Set the path to the overlay image file.
- `#define FE2_OVERLAY_MASK_XY`
Use the pixel at coordinates (`FE2_OVERLAY_MASK_X`, `FE2_OVERLAY_MASK_Y`) to determine the transparent color.
- `#define FE2_OVERLAY_MASK_X`
X coordinate of pixel to use for transparency.
- `#define FE2_OVERLAY_MASK_Y`
Y coordinate of pixel to use for transparency.
- `#define FE2_OVERLAY_MASK_RGB`
Use the RGB value (`FE2_OVERLAY_MASK_R`, `FE2_OVERLAY_MASK_G`, `FE2_OVERLAY_MASK_B`) as the transparency color.
- `#define FE2_OVERLAY_MASK_R`
Red component of the transparency color.
- `#define FE2_OVERLAY_MASK_G`
Green component of the transparency color.
- `#define FE2_OVERLAY_MASK_B`
Blue component of the transparency color.
- `#define FE2_OVERLAY_POS`
Set the overlay position. Valid modes are defined by `FE2_OverlayPositionMode`.
- `#define FE2_OVERLAY_POS_X`
X coordinate of overlay top left on video.
- `#define FE2_OVERLAY_POS_Y`
Y coordinate of overlay top left on video.

Enumerations

- `enum FE2_OverlayPositionMode {`
 `FE2_OVERLAY_POS_MODE_TOPLEFT,`
 `FE2_OVERLAY_POS_MODE_BOTLEFT,`
 `FE2_OVERLAY_POS_MODE_CENTER,`

```
FE2_OVERLAY_POS_MODE_TOPRIGHT,
FE2_OVERLAY_POS_MODE_BOTRIGHT,
FE2_OVERLAY_POS_MODE_XY }
```

Position modes for use with `FE2_OVERLAY_POS`.

Deprecated functions

- `on2sc overlay_options_Reset` (const `FLIX2HANDLE` flix)
Resets the overlay options.
- `on2sc overlay_options_GetUseOverlay` (const `FLIX2HANDLE` flix, `on2bool` *pUseOverlay)
Determines if an overlay is to be used.
- `on2sc overlay_options_SetUseOverlay` (`FLIX2HANDLE` flix, const `on2bool` bUseOverlay)
Enables or disables overlay usage.
- `on2sc overlay_options_GetOverlayPath` (const `FLIX2HANDLE` flix, `on2tc` *pOverlayFilePath, `int32_t` *pLen)
Returns the path to the overlay image file.
- `on2sc overlay_options_SetOverlayPath` (`FLIX2HANDLE` flix, const `on2tc` *pOverlayFilePath)
Set the path to the overlay image file.
- `on2sc overlay_options_GetMaskPixelXY` (const `FLIX2HANDLE` flix, `int32_t` *pMaskPixelX, `int32_t` *pMaskPixelY)
Return the X and Y coordinates of the mask pixel.
- `on2sc overlay_options_SetMaskPixelXY` (`FLIX2HANDLE` flix, `int32_t` maskPixelX, `int32_t` maskPixelY)
Set the X and Y coordinates of the mask pixel.
- `on2sc overlay_options_GetMaskPixelRGB` (`FLIX2HANDLE` flix, `uint8_t` *pMaskPixelR, `uint8_t` *pMaskPixelG, `uint8_t` *pMaskPixelB)
Return the Red, Green, and Blue component values of the mask pixel.
- `on2sc overlay_options_SetMaskPixelRGB` (`FLIX2HANDLE` flix, `uint8_t` maskPixelR, `uint8_t` maskPixelG, `uint8_t` maskPixelB)
Set the Red, Green, and Blue component values of the mask pixel.
- `on2sc overlay_options_GetOverlayPosition` (`FLIX2HANDLE` flix, `FE2_OverlayPositionMode` *pMode, `uint32_t` *pX, `uint32_t` *pY)
Return the overlay position.
- `on2sc overlay_options_SetOverlayPosition` (`FLIX2HANDLE` flix, `FE2_OverlayPositionMode` mode, `uint32_t` x, `uint32_t` y)
Set the overlay position.

12.31.1 Detailed Description

The overlay video filter is a filter used to apply a PNG image to the output video. In addition this filter supports positioning the overlay and treating sections of the image as transparent either by specifying a pixel of the image to use as a reference, specifying a RGB value to use as a reference or using the alpha channel of the image.

Filter Parameters:

Name	Type	Opt/Reqd	Range
=====			
FE2_OVERLAY_FILE	String	Required	N/A
FE2_OVERLAY_MASK_XY	Boolean	Optional	[on2false,on2true]
FE2_OVERLAY_MASK_X	Numeric	Optional	[0,video width)
FE2_OVERLAY_MASK_Y	Numeric	Optional	[0,video height)
FE2_OVERLAY_MASK_RGB	Boolean	Optional	[on2false,on2true]
FE2_OVERLAY_MASK_R	Numeric	Optional	[0,255]
FE2_OVERLAY_MASK_G	Numeric	Optional	[0,255]
FE2_OVERLAY_MASK_B	Numeric	Optional	[0,255]
FE2_OVERLAY_POS	Numeric	Optional	[
FE2_OverlayPositionMode]			
FE2_OVERLAY_POS_X	Numeric	Optional	[0,video width)
FE2_OVERLAY_POS_Y	Numeric	Optional	[0,video height)

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_OVERLAY);
//apply image.png to the top left of the output video
if(sc == ON2_OK)
    sc = Flix2_FilterSetParamAsStr(filter, FE2_OVERLAY_FILE, "/tmp/image.png")
    ;
```

12.31.2 Define Documentation

12.31.2.1 #define FE2_FILTER_OVERLAY

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 76 of file overlay.h.

12.31.2.2 #define FE2_OVERLAY_FILE

Set the path to the overlay image file.

Remarks:

Only PNG images, 24 bit bitmaps, and 32 bit bitmaps are supported. Use of PNG images is highly recommended.

Images with an uneven width or uneven height will be **cropped**.

Overlay images larger than the video image will be **cropped**.

When both the video and overlay images contain an alpha channel, the overlay and video alpha will be averaged. For example, if the video at the overlay position was fully transparent, and the overlay was fully opaque, the region of the exported video where the overlay was placed would become 50% opaque.

Definition at line 96 of file overlay.h.

12.31.2.3 #define FE2_OVERLAY_MASK_B

Blue component of the transparency color.

Remarks:

Default value: 0

Note:

Only used if [FE2_OVERLAY_MASK_RGB](#) is set to [on2true](#)

Definition at line 156 of file overlay.h.

12.31.2.4 #define FE2_OVERLAY_MASK_G

Green component of the transparency color.

Remarks:

Default value: 0

Note:

Only used if [FE2_OVERLAY_MASK_RGB](#) is set to [on2true](#)

Definition at line 150 of file overlay.h.

12.31.2.5 #define FE2_OVERLAY_MASK_R

Red component of the transparency color.

Remarks:

Default value: 0

Note:

Only used if [FE2_OVERLAY_MASK_RGB](#) is set to [on2true](#)

Definition at line 144 of file overlay.h.

12.31.2.6 #define FE2_OVERLAY_MASK_RGB

Use the RGB value ([FE2_OVERLAY_MASK_R](#), [FE2_OVERLAY_MASK_G](#), [FE2_OVERLAY_MASK_B](#)) as the transparency color.

Remarks:

The color specified will become transparent when the overlay image is drawn on the video image.

Setting this parameter will disable [FE2_OVERLAY_MASK_XY](#) if it has been set.

Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.

If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Definition at line 138 of file overlay.h.

12.31.2.7 `#define FE2_OVERLAY_MASK_X`

X coordinate of pixel to use for transparency.

Remarks:

Default value: 0

Definition at line 117 of file overlay.h.

12.31.2.8 `#define FE2_OVERLAY_MASK_XY`

Use the pixel at coordinates ([FE2_OVERLAY_MASK_X](#),[FE2_OVERLAY_MASK_Y](#)) to determine the transparent color.

Remarks:

The color at the specified coordinate will become transparent when the overlay image is drawn on the video image.

Setting this parameter will disable [FE2_OVERLAY_MASK_RGB](#) if it has been set.

Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.

If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Definition at line 112 of file overlay.h.

12.31.2.9 `#define FE2_OVERLAY_MASK_Y`

Y coordinate of pixel to use for transparency.

Remarks:

Default value: 0

Definition at line 122 of file overlay.h.

12.31.2.10 `#define FE2_OVERLAY_POS`

Set the overlay position. Valid modes are defined by [FE2_OverlayPositionMode](#).

Remarks:

If the mode is equal to [FE2_OVERLAY_POS_MODE_XY](#), and either coordinate value is invalid ($x > \text{video width}$ or $y > \text{video height}$), the invalid coordinate will be set to 0.

If the value of the x and/or y coordinates cause any portion of the overlay to be outside of the video image bounds, the portion of the overlay image that is out of bounds will be cropped. Default: [FE2_OVERLAY_POS_MODE_TOPLEFT](#)

Definition at line 169 of file overlay.h.

12.31.2.11 #define FE2_OVERLAY_POS_X

X coordinate of overlay top left on video.

Note:

value is only used when [FE2_OVERLAY_POS](#) has been set to [FE2_OVERLAY_POS_MODE_XY](#)

Definition at line 175 of file overlay.h.

12.31.2.12 #define FE2_OVERLAY_POS_Y

Y coordinate of overlay top left on video.

Note:

value is only used when [FE2_OVERLAY_POS](#) has been set to [FE2_OVERLAY_POS_MODE_XY](#)

Definition at line 181 of file overlay.h.

12.31.3 Enumeration Type Documentation**12.31.3.1 enum FE2_OverlayPositionMode**

Position modes for use with [FE2_OVERLAY_POS](#).

Enumerator:

FE2_OVERLAY_POS_MODE_TOPLEFT draw overlay image on the top left corner of the video
FE2_OVERLAY_POS_MODE_BOTLEFT draw overlay image on the bottom left corner of the video
FE2_OVERLAY_POS_MODE_CENTER draw overlay image on the center of the video
FE2_OVERLAY_POS_MODE_TOPRIGHT draw overlay image on the top right corner of the video
FE2_OVERLAY_POS_MODE_BOTRIGHT draw overlay image on the bottom right corner of the video
FE2_OVERLAY_POS_MODE_XY draw overlay image on the video at the x and y coordinates specified

Definition at line 57 of file overlay.h.

12.31.4 Function Documentation**12.31.4.1 on2sc overlay_options_GetMaskPixelRGB (FLIX2HANDLE *flix*, uint8_t * *pMaskPixelR*, uint8_t * *pMaskPixelG*, uint8_t * *pMaskPixelB*)**

Return the Red, Green, and Blue component values of the mask pixel.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

- *pMaskPixelR* pointer to variable that will receive the R value
- *pMaskPixelG* pointer to variable that will receive the G value
- *pMaskPixelB* pointer to variable that will receive the B value

Return values:

- [ON2_OK](#) the color component values were retrieved successfully
- [ON2_INVALID_PARAMS](#) a precondition failed

Precondition:

flx is not NULL
pMaskPixelR is not NULL
pMaskPixelG is not NULL
pMaskPixelB is not NULL

Remarks:

This method returns the values set in previous call to [overlay_options_SetMaskPixelRGB\(\)](#), if no coordinates were previously set, the Red, Green, and Blue component values will be set to 0.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_R](#), [FE2_OVERLAY_MASK_G](#) and [FE2_OVERLAY_MASK_B](#) parameters. This function will be removed in a future release.

12.31.4.2 on2sc overlay_options_GetMaskPixelXY (const FLIX2HANDLE *flx*, int32_t * *pMaskPixelX*, int32_t * *pMaskPixelY*)

Return the X and Y coordinates of the mask pixel.

Parameters:

- ← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pMaskPixelX* pointer to variable that will receive the X value
- *pMaskPixelY* pointer to variable that will receive the Y value

Return values:

- [ON2_OK](#) the coordinate values were retrieved successfully
- [ON2_INVALID_PARAMS](#) a precondition failed

Precondition:

flx is not NULL
pMaskPixelX is not NULL
pMaskPixelY is not NULL

Remarks:

This method returns the values set in previous call to [overlay_options_SetMaskPixelXY\(\)](#), if no coordinates were previously set, the X and Y coordinate values will be set to 0.
 The color at the specified coordinate will become transparent when the overlay image is drawn on the video image.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_X](#) and [FE2_OVERLAY_MASK_Y](#) parameters. This function will be removed in a future release.

12.31.4.3 on2sc overlay_options_GetOverlayPath (const FLIX2HANDLE *flx*, on2tc * *pOverlayFilePath*, int32_t * *pLen*)

Returns the path to the overlay image file.

Parameters:

- ← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pOverlayFilePath* string that will receive the file path
- ↔ *pLen* length of *pOverlayFilePath*

Return values:

- [ON2_OK](#) success
- [ON2_INVALID_PARAMS](#) a precondition failed

Precondition:

- flx* is not NULL
- pLen* is not NULL

Remarks:

if *pOverlayFilePath* is not NULL, the value pointed to by *pLen* must be set to the maximum number of characters that can be written to *pOverlayFilePath*.
If *pOverlayFilePath* is NULL, *pLen* will be set to the length of the path to the overlay image

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_FILE](#) parameter. This function will be removed in a future release.

12.31.4.4 on2sc overlay_options_GetOverlayPosition (FLIX2HANDLE *flx*, FE2_OverlayPositionMode * *pMode*, uint32_t * *pX*, uint32_t * *pY*)

Return the overlay position.

Parameters:

- ← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pMode* pointer to variable that will receive the mode value
- *pX* pointer to variable that will receive the X value
- *pY* pointer to variable that will receive the Y value

Return values:

- [ON2_OK](#) the values were retrieved successfully

ON2_INVALID_PARAMS a precondition failed

Precondition:

flix is not NULL
pMode is not NULL
pX is not NULL
pY is not NULL

Remarks:

This method returns the values set in previous call to `overlay_options_SetOverlayPosition()`, if no call was made previously, mode will default to `FE2_OVERLAY_POS_MODE_TOPLEFT`. The X/Y coordinates will default to 0.

Deprecated

Use the [Filter Interface](#) along with `FE2_FILTER_OVERLAY` and the `FE2_OVERLAY_POS`, `FE2_OVERLAY_POS_X`, `FE2_OVERLAY_POS_Y` parameters. This function will be removed in a future release.

12.31.4.5 on2sc overlay_options_GetUseOverlay (const FLIX2HANDLE *flix*, on2bool **pUseOverlay*)

Determines if an overlay is to be used.

Parameters:

← *flix* Handle to the flix engine returned from `Flix2_Create()` or `Flix2_CreateEx()`
 → *pUseOverlay* pointer to the `on2bool` variable that will receive the usage state

Return values:

ON2_OK *pUseOverlay* was set successfully
ON2_INVALID_PARAMS a precondition failed

Precondition:

flix is not NULL
pUseOverlay is not NULL

Remarks:

Call this function to determine if an overlay is to be drawn on the encoded video output.

Deprecated

Use the [Filter Interface](#) along with `FE2_FILTER_OVERLAY`. This function will be removed in a future release.

12.31.4.6 on2sc overlay_options_Reset (const FLIX2HANDLE *flix*)

Resets the overlay options.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

[ON2_OK](#) reset of overlay options was successful

[ON2_INVALID_PARAMS](#) the precondition failed

Precondition:

flix is not NULL

Remarks:

Call this function if you wish to reset all overlay options to their default values.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#). This function will be removed in a future release.

12.31.4.7 on2sc overlay_options_SetMaskPixelRGB (FLIX2HANDLE *flix*, uint8_t *maskPixelR*, uint8_t *maskPixelG*, uint8_t *maskPixelB*)

Set the Red, Green, and Blue component values of the mask pixel.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *maskPixelR* the R component value

← *maskPixelG* the G component value

← *maskPixelB* the B component value

Return values:

[ON2_OK](#) the component values were set successfully

[ON2_INVALID_PARAMS](#) the precondition failed

Precondition:

flix is not NULL

Remarks:

The color specified will become transparent when the overlay image is drawn on the video image.

Once this function has been called successfully, [overlay_options_Reset\(\)](#) must be called to turn transparency off.

Calling this method will disable masking based on XY coordinate.

Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.

If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_R](#), [FE2_OVERLAY_MASK_G](#) and [FE2_OVERLAY_MASK_B](#) parameters. This function will be removed in a future release.

12.31.4.8 on2sc overlay_options_SetMaskPixelXY (FLIX2HANDLE *flix*, int32_t *maskPixelX*, int32_t *maskPixelY*)

Set the X and Y coordinates of the mask pixel.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *maskPixelX* the X coordinate value
- ← *maskPixelY* the Y coordinate value

Return values:

- [ON2_OK](#) the coordinate values were set successfully
- [ON2_INVALID_PARAMS](#) the precondition failed

Precondition:

flix is not NULL

Remarks:

The color at the specified coordinate will become transparent when the overlay image is drawn on the video image.
 Once this function has been called successfully, [overlay_options_Reset\(\)](#) must be called to turn transparency off.
 Calling this method will disable masking based on RGB components.
 Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.
 If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_MASK_X](#) and [FE2_OVERLAY_MASK_Y](#) parameters. This function will be removed in a future release.

12.31.4.9 on2sc overlay_options_SetOverlayPath (FLIX2HANDLE *flix*, const on2tc * *pOverlayFilePath*)

Set the path to the overlay image file.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *pOverlayFilePath* the path to the overlay image

Return values:

[*ON2_OK*](#) the path to the overlay image was set successfully

[*ON2_INVALID_PARAMS*](#) a precondition failed

Precondition:

flix is not NULL

pOverlayFilePath is not NULL

Remarks:

This method does **not** check for the existence of the file at the specified path.

Only PNG images, 24 bit bitmaps, and 32 bit bitmaps are supported. Use of PNG images is highly recommended.

Images with an uneven width or uneven height will be **cropped**.

Overlay images larger than the video image will be **cropped**.

When both the video and overlay images contain an alpha channel, the overlay and video alpha will be averaged. For example, if the video at the overlay position was fully transparent, and the overlay was fully opaque, the region of the exported video where the overlay was placed would become 50% opaque.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_FILE](#) parameter. This function will be removed in a future release.

12.31.4.10 on2sc overlay_options_SetOverlayPosition (FLIX2HANDLE *flix*, FE2_OverlayPositionMode *mode*, uint32_t *x*, uint32_t *y*)

Set the overlay position.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *mode* the position mode

→ *x* the x coordinate value

→ *y* the y coordinate value

Return values:

[*ON2_OK*](#) success

[*ON2_INVALID_PARAMS*](#) the precondition failed

Precondition:

flix is not NULL

Remarks:

The x and y values are ignored unless *mode* is equal to [FE2_OVERLAY_POS_MODE_XY](#).

If *mode* is equal to [FE2_OVERLAY_POS_MODE_XY](#), and either coordinate value is invalid (*x* > video width or *y* > video height), the invalid coordinate will be set to 0.

If the the value of the *x* and/or *y* coordinates cause any portion of the overlay to be outside of the video image bounds, the portion of the overlay image that is out of bounds will be cropped.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#) and the [FE2_OVERLAY_POS](#), [FE2_OVERLAY_POS_X](#), [FE2_OVERLAY_POS_Y](#) parameters. This function will be removed in a future release.

12.31.4.11 on2sc overlay_options_SetUseOverlay (FLIX2HANDLE *flix*, const on2bool *bUseOverlay*)

Enables or disables overlay usage.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseOverlay* [on2bool](#) value set to the desired overlay state

Return values:

- [ON2_OK](#) the overlay status was changed successfully
- [ON2_INVALID_PARAMS](#) the precondition failed

Precondition:

flix is NULL

Remarks:

Set *bUseOverlay* to [on2true](#) to enable overlay usage. Set *bUseOverlay* to [on2false](#) to disable usage.

Only PNG images, 24 bit bitmaps, and 32 bit bitmaps are supported. Use of PNG images is highly recommended.

Images with an uneven width or uneven height will be **cropped**.

Overlay images larger than the video image will be **cropped**.

When both the video and overlay images contain an alpha channel, the overlay and video alpha will be averaged. For example, if the video at the overlay position was fully transparent, and the overlay was fully opaque, the region of the exported video where the overlay was placed would become 50% opaque.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_OVERLAY](#). This function will be removed in a future release.

12.32 PNG Image Export (Thumbnail)

Defines

- #define `FE2_FILTER_PNGEX`
Filter name for use with `Flix2_AddFilter()`.
- #define `FE2_PNGEX_OUTPUT_DIRECTORY`
Filter parameter for setting PNG image output directory.
- #define `FE2_PNGEX_FILENAME_PREFIX`
Filter parameter for setting PNG image file name prefix.
- #define `FE2_PNGEX_FILENAME_SUFFIX`
Filter parameter for setting PNG image file name suffix.
- #define `FE2_PNGEX_ENABLE_ALPHA`
Filter parameter for preserving source video alpha channel data in exported PNG images.
- #define `FE2_PNGEX_EXPORT_INTERVAL`
Filter parameter for enabling PNG image export at a millisecond interval.
- #define `FE2_PNGEX_EXPORT_TIME_STRING`
Filter parameter for setting a string of PNG export times in milliseconds.
- #define `FE2_PNGEX_EXPORT_CUE_POINTS`
Filter parameter for enabling export of PNG images at cue points.
- #define `FE2_PNGEX_AUTO_EXPORT_COUNT`
Filter parameter for enabling automatic PNG image generation.
- #define `FE2_PNGEX_AUTO_EXPORT_START_TIME`
Filter parameter for setting PNG auto generation start time.
- #define `FE2_PNGEX_AUTO_EXPORT_END_TIME`
Filter parameter for setting PNG auto generation end time.
- #define `FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD`
Filter parameter for randomizing auto generated times.
- #define `FE2_PNGEX_EXPORT_FIRST_FRAME_PNG`
Filter parameter for enabling PNG image creation using the first video frame.
- #define `FE2_PNGEX_COMPRESSION_LEVEL`
Filter parameter for setting compression level used by libpng.
- #define `FE2_PNGEX_WIDTH`
Filter parameter for setting PNG image width.
- #define `FE2_PNGEX_HEIGHT`
Filter parameter for setting PNG image height.

Enumerations

- enum [FE2_PNGEX_CuePtMode](#) {
[FE2_PNGEX_CP_EVENT](#),
[FE2_PNGEX_CP_NAV](#),
[FE2_PNGEX_CP_ALL](#) }

Constants for use with [FE2_PNGEX_EXPORT_CUE_POINTS](#).

12.32.1 Detailed Description

The PNG image export filter creates PNG images from the source video data of the input file passed to [Flix2_SetInputFile\(\)](#).

Filter Parameters:

Name	Type	Opt/Reqd	Range
=====			
FE2_PNGEX_OUTPUT_DIRECTORY	String	Optional	N/A
FE2_PNGEX_FILENAME_PREFIX	String	Optional	N/A
FE2_PNGEX_FILENAME_SUFFIX	String	Optional	N/A
FE2_PNGEX_ENABLE_ALPHA	Boolean	Optional	[0,1]
FE2_PNGEX_EXPORT_INTERVAL	Numeric	Optional	[1,video stream duration]
FE2_PNGEX_EXPORT_TIME_STRING	String	Optional	N/A
FE2_PNGEX_EXPORT_CUE_POINTS	Numeric	Optional	[0,3]
FE2_PNGEX_AUTO_EXPORT_COUNT	Numeric	Optional	[0,10000]
FE2_PNGEX_AUTO_EXPORT_START_TIME	Numeric	Optional	[0,video stream duration]
FE2_PNGEX_AUTO_EXPORT_END_TIME	Numeric	Optional	[0,video stream duration]
FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD	Numeric	Optional	[1,)
FE2_PNGEX_EXPORT_FIRST_FRAME_PNG	Boolean	Optional	[0,1]
FE2_PNGEX_COMPRESSION_LEVEL	Numeric	Optional	[-1,9]
FE2_PNGEX_WIDTH	Numeric	Optional	FE2_SCALE_WIDTH
FE2_PNGEX_HEIGHT	Numeric	Optional	FE2_SCALE_HEIGHT

Example Usage:

```
sc = Flix2_AddFilter(&filter, flx, FE2_FILTER_PNGEX);
// Automatically export 10 PNG images spaced evenly through the clip
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_PNGEX_AUTO_EXPORT_COUNT, 10);
```

Notes:

- All PNG exporter parameters are optional, but at least one of the following parameters must be used in order to export images:
 - [FE2_PNGEX_AUTO_EXPORT_COUNT](#)
 - [FE2_PNGEX_EXPORT_CUE_POINTS](#)
 - [FE2_PNGEX_EXPORT_FIRST_FRAME_PNG](#)
 - [FE2_PNGEX_EXPORT_INTERVAL](#)
 - [FE2_PNGEX_EXPORT_TIME_STRING](#)

- Image file naming:
 - The general form of an output image name is:
`FE2_PNGEX_OUTPUT_DIRECTORY '/' FE2_PNGEX_FILENAME_PREFIX nnnn FE2_PNGEX_FILENAME_SUFFIX '.png'`
 using values supplied or defaults as described by each parameter.
 - Image file numbering starts at 0, the numeric portion of the file name will be expanded to 4 columns (e.g., 0 will be 0000). This limits the export total to 10000 images. If 10000 images per encoded file is inadequate, please contact [support](#).
- Duplicate PNG images:
 - If any of `FE2_PNGEX_AUTO_EXPORT_COUNT`, `FE2_PNGEX_EXPORT_CUE_POINTS` or `FE2_PNGEX_EXPORT_TIME_STRING` are enabled and a time is duplicated, the duplicate time will be discarded.
 - If `FE2_PNGEX_EXPORT_INTERVAL` is enabled and at least one of `FE2_PNGEX_AUTO_EXPORT_COUNT`, `FE2_PNGEX_EXPORT_CUE_POINTS` or `FE2_PNGEX_EXPORT_TIME_STRING` is enabled, the time interval based PNG image will be generated using the frame subsequent to the frame that would otherwise have been duplicated if times overlap.
- Incorrect duration:
 - If the duration is unavailable (i.e., `Flix2_GetSourceDuration()` returns -1) the first `FE2_PNGEX_AUTO_EXPORT_COUNT` frames will be output.
 - If the duration is misreported (i.e., >actual) and not all requested PNGs have been output, one additional PNG will be generated on the last frame.

Additional References:

For information about the PNG file format, please visit: www.libpng.org

12.32.2 Define Documentation

12.32.2.1 #define FE2_FILTER_PNGEX

Filter name for use with `Flix2_AddFilter()`.

Definition at line 99 of file `png_export.h`.

12.32.2.2 #define FE2_PNGEX_AUTO_EXPORT_COUNT

Filter parameter for enabling automatic PNG image generation. If enabled, the PNG exporter will export the specified number of PNG images at times spaced evenly throughout the video stream. See the notes section for important information about this feature and single pass encoding. Time slice duration is calculated using the following:

```
time_slice_duration = video_stream_duration / FE2_PNGEX_AUTO_EXPORT_COUNT
```

The first auto generated PNG image time is calculated using the following equation:

```
first_png_time = FE2_PNGEX_AUTO_EXPORT_START_TIME > 0 ?  
    FE2_PNGEX_AUTO_EXPORT_START_TIME : time_slice_duration / 2
```

Additional PNG time(s) is/are calculated using the following equation:

```

n: 0 based PNG image index
png_time = first_png_time + (time_slice_duration * n)

sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // auto export 10 PNG images
    sc = Flix2_FilterSetParam(png_exporter_plgn,
        FE2_PNGEX_AUTO_EXPORT_COUNT, 10);

```

Definition at line 263 of file png_export.h.

12.32.2.3 #define FE2_PNGEX_AUTO_EXPORT_END_TIME

Filter parameter for setting PNG auto generation end time. The PNG exporter will use this value as the video stream duration in auto export time calculations. The value specified **MUST** be in milliseconds.

```

sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // set the auto generation end time to 10 seconds
    sc = Flix2_FilterSetParam(png_exporter_plgn,
        FE2_PNGEX_AUTO_EXPORT_END_TIME, 10000);

```

Definition at line 305 of file png_export.h.

12.32.2.4 #define FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD

Filter parameter for randomizing auto generated times. When set to a value greater than zero, PNG image generation will occur at random offsets from the times at which images would normally be generated when [FE2_PNGEX_AUTO_EXPORT_COUNT](#) is enabled. The value of this parameter controls the time period (offset range) within which the image generation will occur. The duration of each time slice is calculated using the following equation:

$$\text{time_slice_duration} = \frac{\text{video_stream_duration}}{\text{FE2_PNGEX_AUTO_EXPORT_COUNT}}$$

[FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD](#) is used to calculate the range of time within which to randomize the PNG generation time:

$$\text{random_offset_range} = \frac{\text{time_slice_duration}}{\text{FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD}}$$

The time of each automatically generated PNG is then calculated:

```

f: first png time
n: 0 based PNG image index
r: random offset in the range 0-random_range
png_time = f + (n * time_slice_duration) + r;

```

Definition at line 336 of file png_export.h.

12.32.2.5 #define FE2_PNGEX_AUTO_EXPORT_START_TIME

Filter parameter for setting PNG auto generation start time. The value specified **MUST** be in milliseconds. This parameter affects the following changes on time calculations explained in the [FE2_PNGEX_AUTO_EXPORT_COUNT](#) section:

```
time_slice_duration =
    (video_stream_duration - FE2_PNGEX_AUTO_EXPORT_START_TIME) /
    FE2_PNGEX_AUTO_EXPORT_COUNT
first_png_time = FE2_PNGEX_AUTO_EXPORT_START_TIME
```

Subsequent PNG times will be calculated the final equation outlined in the [FE2_PNGEX_AUTO_EXPORT_COUNT](#) section.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flx, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // set the first auto export time to 1 second
    sc = Flix2_FilterSetParam(png_exporter_plgn,
        FE2_PNGEX_AUTO_EXPORT_START_TIME, 1000);
```

Definition at line 289 of file png_export.h.

12.32.2.6 #define FE2_PNGEX_COMPRESSION_LEVEL

Filter parameter for setting compression level used by libpng. Notes from png.h:

Set the library compression level. Currently, valid values range from 0 - 9, corresponding directly to the zlib compression levels 0 - 9 (0 - no compression, 9 - "maximal" compression). Note that tests have shown that zlib compression levels 3-6 usually perform as well as level 9 for PNG images, and do considerably fewer calculations. In the future, these values may not correspond directly to the zlib compression levels.

In addition to the above range, -1 corresponds to zlib's default value.

Note:

Default: -1 (Z_DEFAULT_COMPRESSION)

Definition at line 370 of file png_export.h.

12.32.2.7 #define FE2_PNGEX_ENABLE_ALPHA

Filter parameter for preserving source video alpha channel data in exported PNG images. If enabled, the PNG export filter preserves the source video alpha when exporting PNG images. This setting will only produce PNG images with alpha channels if encoding Flash 8 Video with Alpha.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flx, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // enable alpha preservation
    sc = Flix2_FilterSetParam(png_exporter_plgn,
        FE2_PNGEX_ENABLE_ALPHA, on2true);
```

Definition at line 172 of file png_export.h.

12.32.2.8 #define FE2_PNGEX_EXPORT_CUE_POINTS

Filter parameter for enabling export of PNG images at cue points. If enabled, PNG images will be exported at event, navigation, or all cue points. See [video_options_AddFLVCuePoint\(\)](#), for information with regard to adding cue points to your output FLV file.

Use [FE2_PNGEX_CP_EVENT](#), [FE2_PNGEX_CP_NAV](#), or [FE2_PNGEX_CP_ALL](#) to specify the types of cue points for which the PNG exporter should export PNG images.

```

sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // generate PNGs for all cue points
    sc = Flix2_FilterSetParam(png_exporter_plgn,
        FE2_PNGEX_EXPORT_CUE_POINTS, FE2_PNGEX_CP_ALL);

```

Definition at line 229 of file png_export.h.

12.32.2.9 #define FE2_PNGEX_EXPORT_FIRST_FRAME_PNG

Filter parameter for enabling PNG image creation using the first video frame. This parameter will force the PNG exporter to generate a PNG image from the first video frame in the input file.

```

sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // create a PNG image of the first video frame
    sc = Flix2_FilterSetParam(png_exporter_plgn,
        FE2_PNGEX_EXPORT_FIRST_FRAME_PNG, on2true);

```

Definition at line 353 of file png_export.h.

12.32.2.10 #define FE2_PNGEX_EXPORT_INTERVAL

Filter parameter for enabling PNG image export at a millisecond interval. The PNG image exporter will export a PNG once every [FE2_PNGEX_EXPORT_INTERVAL](#) milliseconds if this parameter is specified in a call to [Flix2_FilterSetParam\(\)](#)

```

sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // generate a PNG every 10 seconds
    sc = Flix2_FilterSetParam(png_exporter_plgn,
        FE2_PNGEX_EXPORT_INTERVAL, 10000);

```

Definition at line 190 of file png_export.h.

12.32.2.11 #define FE2_PNGEX_EXPORT_TIME_STRING

Filter parameter for setting a string of PNG export times in milliseconds. The PNG exporter will create PNG images from the source video stream at the times specified in the string. The string **MUST** be in the format: t0,t1,t2,tn

```

sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // produce PNG images from the source video
    // at times of 5 seconds, 10 seconds, and 15 seconds
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
        FE2_PNGEX_EXPORT_TIME_STRING, "5000,10000,15000");

```

Definition at line 209 of file png_export.h.

12.32.2.12 #define FE2_PNGEX_FILENAME_PREFIX

Filter parameter for setting PNG image file name prefix. If a prefix is specified, PNG image files will be named in the format *FE2_PNGEX_FILENAME_PREFIX*png_number.png. If a prefix is not specified, the

file name passed to [Flix2_SetOutputFile\(\)](#) will be used. If no prefix is desired, the prefix **MUST** be set to an empty string.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flx, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // use prefix for PNG output file names
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
        FE2_PNGEX_FILENAME_PREFIX, "pngprefix");
```

Definition at line 136 of file png_export.h.

12.32.2.13 #define FE2_PNGEX_FILENAME_SUFFIX

Filter parameter for setting PNG image file name suffix. If a suffix is specified, PNG image files will be named in the format png_number**FE2_PNGEX_FILENAME_SUFFIX**.png.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flx, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // use suffix for PNG output file names
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
        FE2_PNGEX_FILENAME_SUFFIX, "pngsuffix");
```

Note:

Default: empty

Definition at line 154 of file png_export.h.

12.32.2.14 #define FE2_PNGEX_HEIGHT

Filter parameter for setting PNG image height.

Note:

Default: input height

From **8.0.16.0**, use of the [reserved values](#) defined for [FE2_SCALE_HEIGHT](#) is valid

Definition at line 388 of file png_export.h.

12.32.2.15 #define FE2_PNGEX_OUTPUT_DIRECTORY

Filter parameter for setting PNG image output directory. If a directory is not specified, the PNG exporter will write PNG images to the directory specified in the path passed to [Flix2_SetOutputFile\(\)](#).

For example, the following code will set the PNG export directory to /png/output/directory:

```
sc = Flix2_AddFilter(&png_exporter_plgn, flx, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // output PNGs to directory
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
        FE2_PNGEX_OUTPUT_DIRECTORY, "/png/output/directory");
```

Definition at line 118 of file png_export.h.

12.32.2.16 `#define FE2_PNGEX_WIDTH`

Filter parameter for setting PNG image width.

Note:

Default: input width

From **8.0.16.0**, use of the [reserved values](#) defined for [FE2_SCALE_WIDTH](#) is valid

Definition at line 379 of file `png_export.h`.

12.32.3 Enumeration Type Documentation

12.32.3.1 `enum FE2_PNGExCuePtMode`

Constants for use with [FE2_PNGEX_EXPORT_CUE_POINTS](#).

Enumerator:

FE2_PNGEX_CP_EVENT PNG image exporter will export PNGs for only event cue points

FE2_PNGEX_CP_NAV PNG image exporter will export PNGs for only navigation cue points

FE2_PNGEX_CP_ALL PNG image exporter will export PNGs for all cue points

Definition at line 391 of file `png_export.h`.

12.33 Resample

Defines

- `#define FE2_FILTER_RESAMPLE`
Filter name for use with [Flix2_AddFilter\(\)](#).
- `#define FE2_RESAMPLE_RATE`
Desired sample rate in Hertz (Hz).
- `#define FE2_RESAMPLE_CHANNELS`
Filter parameter for channels.

12.33.1 Detailed Description

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_RESAMPLE_RATE	Numeric	Optional	[8000,192000]Hz
FE2_RESAMPLE_CHANNELS	Numeric	Optional	[1,2]

Usage notes:

- FLV/SWF files are limited to sample rates of [Hertz11025](#), [Hertz22050](#) and [Hertz44100](#).

Example Usage:

```
sc = Flix2_AddFilter(&filter, flx, FE2_FILTER_RESAMPLE);
// Resample to 22050Hz
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_RESAMPLE_RATE, Hertz22050);
```

12.33.2 Define Documentation

12.33.2.1 `#define FE2_FILTER_RESAMPLE`

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 50 of file resample.h.

12.33.2.2 `#define FE2_RESAMPLE_CHANNELS`

Filter parameter for channels.

Definition at line 58 of file resample.h.

12.33.2.3 `#define FE2_RESAMPLE_RATE`

Desired sample rate in Hertz (Hz).

Definition at line 54 of file resample.h.

12.34 Rotate

Defines

- `#define FE2_FILTER_ROTATE`
Filter name for use with [Flix2_AddFilter\(\)](#).
- `#define FE2_ROTATE_ANGLE`
Specifies the image rotation angle in degrees.

12.34.1 Detailed Description

The rotate filter rotates the source clockwise by the angle specified by [FE2_ROTATE_ANGLE](#).

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_ROTATE_ANGLE	Numeric	Optional	{0,90,180,270}

Example Usage:

```
sc = Flix2_AddFilter(&filter, flx, FE2_FILTER_ROTATE);
// rotate source 90 degrees clockwise
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_ROTATE_ANGLE, 90.0);
```

12.34.2 Define Documentation

12.34.2.1 `#define FE2_FILTER_ROTATE`

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 45 of file rotate.h.

12.34.2.2 `#define FE2_ROTATE_ANGLE`

Specifies the image rotation angle in degrees. The source is rotated clockwise.

Note:

Default: 0.0 (no rotation)
Valid range: {0,90,180,270}

Definition at line 52 of file rotate.h.

12.35 Scale

Defines

- #define `FE2_FILTER_SCALE`
Filter name for use with `Flix2_AddFilter()`.
- #define `FE2_SCALE_WIDTH`
Filter parameter name for scaled width.
- #define `FE2_SCALE_HEIGHT`
Filter parameter name for scaled height.

Deprecated functions

- `on2sc video_options_GetImageHeight` (const `FLIX2HANDLE` flix, `int32_t` *lpImageHeight)
Get the current scaled image height.
- `on2sc video_options_SetImageHeight` (`FLIX2HANDLE` flix, const `int32_t` lImageHeight)
Set the scaled image height.
- `on2sc video_options_GetImageWidth` (const `FLIX2HANDLE` flix, `int32_t` *lpImageWidth)
Get the current scaled image width.
- `on2sc video_options_SetImageWidth` (`FLIX2HANDLE` flix, const `int32_t` lImageWidth)
Set the scaled image width.
- `on2sc video_options_GetUseSourceDimensions` (const `FLIX2HANDLE` flix, `on2bool` *bpUseSourceDimensions)
Determine if the source's dimensions will be used instead of the scaled dimensions.
- `on2sc video_options_SetUseSourceDimensions` (`FLIX2HANDLE` flix, const `on2bool` bUseSourceDimensions)
Switch between source and scaled dimensions.
- `on2sc video_options_GetDeinterlace` (const `FLIX2HANDLE` flix, `on2bool` *lpDeinterlace)
Determine if the deinterlace filter is enabled.
- `on2sc video_options_SetDeinterlace` (`FLIX2HANDLE` flix, const `on2bool` lDeinterlace)
Enable/disable the deinterlace filter.
- `on2sc video_options_GetVideoFramerate` (const `FLIX2HANDLE` flix, `int32_t` *lpVideoFramerate)
Get the current video framerate.
- `on2sc video_options_SetVideoFramerate` (`FLIX2HANDLE` flix, const `int32_t` lVideoFramerate)
Set the video framerate.

- `on2sc video_options_GetVideoFramerateAsDouble` (const `FLIX2HANDLE` flix, double *p_fps)
Get the current video framerate.
- `on2sc video_options_SetVideoFramerateAsDouble` (`FLIX2HANDLE` flix, const double fps)
Set the video framerate.
- `on2sc video_options_GetUseSourceFramerate` (`FLIX2HANDLE` flix, `on2bool` *bpUseSourceFramerate)
Determine if the source's framerate will be used instead of the modified framerate.
- `on2sc video_options_SetUseSourceFramerate` (`FLIX2HANDLE` flix, const `on2bool` bUseSourceFramerate)
Switch between source and scaled framerate.
- `on2sc video_options_GetDecimateValue` (const `FLIX2HANDLE` flix, `uint32_t` *pValue)
Get the current decimation of the video framerate.
- `on2sc video_options_SetDecimateValue` (`FLIX2HANDLE` flix, const `uint32_t` value)
Set the decimation of the video framerate.

12.35.1 Detailed Description

The video scale (resize) filter implements a high quality bicubic scaler to change the dimensions of the video.

Filter Parameters:

Name	Type	Opt/Reqd	Range
=====			
FE2_SCALE_WIDTH	Numeric	Optional	[-19,)
FE2_SCALE_HEIGHT	Numeric	Optional	[-19,)

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_SCALE);
// scale to 320xXXX maintaining the display aspect ratio
// e.g., 640x480 -> 320x240 (4:3)
//        640x360 -> 320x180 (16:9)
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_SCALE_WIDTH, 320);
if (sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_SCALE_HEIGHT, -2);
```

12.35.2 Reserved values

The following values have special meaning and are applicable to both `FE2_SCALE_WIDTH` and `FE2_SCALE_HEIGHT`:

- 0: display dimension
- -1: original dimension (default)

- -2: calculate dimension based on its complement and the display aspect ratio
- -3: calculate dimension based on its complement and the original aspect ratio
- $n-4*x$: where n is one of the values above with the resulting dimension being an even multiple of 2^x

Display aspect ratio is calculated based on the desired resolution, as set by the container, e.g., the input is encoded at 4:3, but displays at 16:9. Needless to say, this is clip dependent and requires the correct values be set within the container and forwarded from the input source. If unavailable this value will be set to the encoded resolution.

Attention:

Only one dimension may be set to $\{-2-4*[0, 4], -3-4*[0, 4]\}$. The other MUST be $\{0-4*[0, 4], -1-4*[0, 4]\}$ or > 0 .

12.35.3 Define Documentation

12.35.3.1 #define FE2_FILTER_SCALE

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 73 of file scale.h.

12.35.3.2 #define FE2_SCALE_HEIGHT

Filter parameter name for scaled height. Use FE2_SCALE_HEIGHT with [Flix2_FilterSetParam\(\)](#) and [Flix2_FilterGetParam\(\)](#) to set and/or get the scaled height of the video.

Note:

The value returned by a call to [Flix2_FilterGetParam\(\)](#) when this name is specified should only be considered valid if it follows a call to [Flix2_FilterSetParam\(\)](#).

Attention:

This function can NOT be used to obtain information about the unscaled input video, use [video_options_GetSourceHeight\(\)](#).

See also:

[Reserved values](#)

Definition at line 103 of file scale.h.

12.35.3.3 #define FE2_SCALE_WIDTH

Filter parameter name for scaled width. Use FE2_SCALE_WIDTH with [Flix2_FilterSetParam\(\)](#) and [Flix2_FilterGetParam\(\)](#) to set and/or get the scaled width of the video.

Note:

The value returned by a call to [Flix2_FilterGetParam\(\)](#) when this name is specified should only be considered valid if it follows a call to [Flix2_FilterSetParam\(\)](#).

Attention:

This function can NOT be used to obtain information about the unscaled input video, use [video_options_GetSourceWidth\(\)](#).

See also:

[Reserved values](#)

Definition at line 88 of file scale.h.

12.35.4 Function Documentation**12.35.4.1 on2sc video_options_GetDecimateValue (const FLIX2HANDLE *flix*, uint32_t * *pValue*)**

Get the current decimation of the video framerate.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *pValue* Decimate value

Return values:

[ON2_OK](#) Success.
[ON2_INVALID_PARAMS](#) Should one or more of the preconditions fail.

Precondition:

flix is not NULL
pValue is not NULL

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_DECIMATE](#) parameter. This function will be removed in a future release.

12.35.4.2 on2sc video_options_GetDeinterlace (const FLIX2HANDLE *flix*, on2bool * *lpDeinterlace*)

Determine if the deinterlace filter is enabled.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *lpDeinterlace* Variable to update with the current deinterlace enable status

Return values:

[ON2_OK](#) The deinterlace filter enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_ADAPTIVE_DEINTERLACE](#). This function will be removed in a future release.

12.35.4.3 on2sc video_options_GetImageHeight (const FLIX2HANDLE *flix*, int32_t * *lpImageHeight*)

Get the current scaled image height.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *lpImageHeight* Image height (in pixels)

Return values:

[ON2_OK](#) The height value was successfully retrieved from the engine.

Note:

This value should only be considered valid if it follows a call to [video_options_SetImageHeight\(\)](#). It can not be used to get information about the unscaled input video.

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flix engine video scaling filter.

12.35.4.4 on2sc video_options_GetImageWidth (const FLIX2HANDLE *flix*, int32_t * *lpImageWidth*)

Get the current scaled image width.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *lpImageWidth* Image width (in pixels)

Return values:

[*ON2_OK*](#) The width value was successfully retrieved from the engine.

Note:

This value should only be considered valid if it follows a call to [video_options_SetImageWidth\(\)](#). It can not be used to get information about the unscaled input video.

If a value other than *ON2_OK* is returned, the output variables should not be considered valid.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flx engine video scaling filter.

12.35.4.5 **on2sc video_options_GetUseSourceDimensions** (const FLIX2HANDLE *flx*, on2bool * *bpUseSourceDimensions*)

Determine if the source's dimensions will be used instead of the scaled dimensions.

Parameters:

- ← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *bpUseSourceDimensions* Variable to update with the current dimension status

Return values:

[*ON2_OK*](#) The dimension status was successfully retrieved from the engine.

Note:

If a value other than *ON2_OK* is returned, the output variables should not be considered valid.

Deprecated

The flx engine encodes video using the video source dimensions by default. If [FE2_FILTER_SCALE](#) has not been added to the filter chain, video will be encoded using source dimensions.

12.35.4.6 **on2sc video_options_GetUseSourceFramerate** (FLIX2HANDLE *flx*, on2bool * *bpUseSourceFramerate*)

Determine if the source's framerate will be used instead of the modified framerate.

Parameters:

- ← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *bpUseSourceFramerate* Variable to update with the current framerate selection

Return values:

ON2_OK The framerate status was successfully retrieved from the engine.

Note:

If a value other than *ON2_OK* is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#). Not adding [FE2_FILTER_FRAMERATE](#) with [Flix2_AddFilter\(\)](#) implies use of the source frame rate. This function will be removed in a future release.

12.35.4.7 on2sc video_options_GetVideoFramerate (const FLIX2HANDLE *flix*, int32_t * *lpVideoFramerate*)

Get the current video framerate.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *lpVideoFramerate* Frames per second

Return values:

ON2_OK The framerate was successfully retrieved from the engine.

Deprecated

Please use [video_options_GetVideoFramerateAsDouble\(\)](#) to allow for non integer framerates.

Note:

If a value other than *ON2_OK* is returned, the output variables should not be considered valid.

12.35.4.8 on2sc video_options_GetVideoFramerateAsDouble (const FLIX2HANDLE *flix*, double * *p_fps*)

Get the current video framerate.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *p_fps* Frames per second

Return values:

ON2_OK The framerate was successfully retrieved from the engine.

Note:

If a value other than *ON2_OK* is returned, the output variables should not be considered valid.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_FPS](#) parameter. This function will be removed in a future release.

12.35.4.9 on2sc video_options_SetDecimateValue (FLIX2HANDLE *flix*, const uint32_t *value*)

Set the decimation of the video framerate. This value will be used to drop frames according to the video framerate. For example if the framerate is 30fps and the decimate value is 3 then the encoded framerate will be 10fps.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *value* Decimate value

Return values:

- [ON2_OK](#) The target framerate was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) Should one or more of the preconditions fail.

Precondition:

flix is not NULL

Note:

- Default value is 0.
- A value of ≤ 1 means to not drop any frames.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_DECIMATE](#) parameter. This function will be removed in a future release.

12.35.4.10 on2sc video_options_SetDeinterlace (FLIX2HANDLE *flix*, const on2bool *lDeinterlace*)

Enable/disable the deinterlace filter.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *lDeinterlace* New deinterlace enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

- [ON2_OK](#) The deinterlace enable status was successfully set in the engine.

Note:

- By default, the deinterlace filter is disabled.
- Using this function will force the use of 1:2:1 deinterlace.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_ADAPTIVE_DEINTERLACE](#). This function will be removed in a future release.

12.35.4.11 on2sc video_options_SetImageHeight (FLIX2HANDLE *flix*, const int32_t *lImageHeight*)

Set the scaled image height. Determines the scaled height of the output video.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *lImageHeight* Image height (in pixels)

Return values:

- [ON2_OK](#) The image height value was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

The default is to use the source's height.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flix engine video scaling filter.

12.35.4.12 on2sc video_options_SetImageWidth (FLIX2HANDLE *flix*, const int32_t *lImageWidth*)

Set the scaled image width. Determines the scaled width of the output video.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *lImageWidth* Image width (in pixels)

Return values:

- [ON2_OK](#) The image width value was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

The default is to use the source's width.

Deprecated

Please use:

- [Flix2_AddFilter\(\)](#)
- [Flix2_FilterGetParam\(\)](#)
- [Flix2_FilterSetParam\(\)](#)

With the constants:

- [FE2_FILTER_SCALE](#)
- [FE2_SCALE_WIDTH](#)
- [FE2_SCALE_HEIGHT](#)

for access to the flx engine video scaling filter.

12.35.4.13 [on2sc video_options_SetUseSourceDimensions](#) (FLIX2HANDLE *flx*, const on2bool *bUseSourceDimensions*)

Switch between source and scaled dimensions.

Parameters:

- ← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseSourceDimensions* New dimension status. [on2true](#) to use source dimensions, [on2false](#) to use scaled dimensions.

Return values:

[ON2_OK](#) The dimension status was successfully set in the engine.

Note:

By default, the source dimensions will be used, unless a call to [video_options_SetImageHeight\(\)](#) or [video_options_SetImageWidth\(\)](#) has been made.

Deprecated

The flx engine encodes video using the video source dimensions by default. If [FE2_FILTER_SCALE](#) has not been added to the filter chain, video will be encoded using source dimensions.

Attention:

Use of this function will cause [FE2_FILTER_SCALE](#) settings to be ignored. If [FE2_FILTER_SCALE](#) has been added to the filter chain via the [Filter Interface](#) this function is unnecessary. Scaling can be disabled by calling [Flix2_RemoveFilter\(\)](#) with the handle returned from [Flix2_AddFilter\(\)](#).

12.35.4.14 [on2sc video_options_SetUseSourceFramerate](#) (FLIX2HANDLE *flx*, const on2bool *bUseSourceFramerate*)

Switch between source and scaled framerate.

Parameters:

- ← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseSourceFramerate* New framerate status. [on2true](#) to use source framerate, [on2false](#) to use scaled framerate.

Return values:

ON2_OK The framerate status was successfully set in the engine.

Note:

By default, the source framerate will be used, unless a call to [video_options_SetVideoFramerate\(\)](#), [video_options_SetVideoFramerateAsDouble\(\)](#) or [video_options_SetDecimateValue\(\)](#) has been made.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#). Not adding [FE2_FILTER_FRAMERATE](#) with [Flix2_AddFilter\(\)](#) implies use of the source frame rate. This function will be removed in a future release.

12.35.4.15 on2sc video_options_SetVideoFramerate (FLIX2HANDLE *flix*, const int32_t *IVideoFramerate*)

Set the video framerate. Sets the framerate to be used when encoding the video. Frames will be duplicated or dropped as necessary to achieve the desired framerate.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *IVideoFramerate* Framerate target (in frames per second)

Return values:

ON2_OK The target framerate was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Deprecated

Please use [video_options_SetVideoFramerateAsDouble\(\)](#) to allow for non integer framerates.

Note:

The default is to use the source framerate.

12.35.4.16 on2sc video_options_SetVideoFramerateAsDouble (FLIX2HANDLE *flix*, const double *fps*)

Set the video framerate. Sets the framerate to be used when encoding the video. Frames will be duplicated or dropped as necessary to achieve the desired framerate.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *fps* Framerate target (in frames per second)

Return values:

ON2_OK The target framerate was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

Note:

The default is to use the source framerate.

Deprecated

Use the [Filter Interface](#) along with [FE2_FILTER_FRAMERATE](#) and the [FE2_FRAMERATE_FPS](#) parameter. This function will be removed in a future release.

12.36 Sharpen

Defines

- `#define FE2_FILTER_SHARPEN`
Filter name for use with [Flix2_AddFilter\(\)](#).

12.36.1 Detailed Description

Sharpens the source image using a 3x3 kernel with the values:

```
-1/8 -1/8 -1/8  
-1/8 16/8 -1/8  
-1/8 -1/8 -1/8
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_SHARPEN);  
if(sc == ON2_OK)  
    ; //a sharpening filter will be applied to the source image
```

12.36.2 Define Documentation

12.36.2.1 `#define FE2_FILTER_SHARPEN`

Filter name for use with [Flix2_AddFilter\(\)](#).

Definition at line 41 of file sharpen.h.

12.37 Flix Engine API

Modules

- [Audio Encoding Options](#)
- [Codecs](#)
- [Encoding Statistics](#)
- [Filters](#)
- [Main Engine Interface](#)
- [Filter Manipulation](#)
- [Codec Manipulation](#)
- [Muxer Manipulation](#)
- [Muxers](#)
- [Video Encoding Options](#)

12.37.1 Detailed Description

The Flix Engine for Windows provides a [COM](#) interface, supplied by *flixengine_com.dll*, which wraps the Flix Engine API. Within this documentation each member of the COM interface will redirect the user to the API member it wraps, where the behavior is fully described.

See also:

[Support Options & FAQ](#)

Version:

8.0.17.2

12.38 Main Engine Interface

Enumerations

- enum `FE2_ExportedVideoType` {
 `ExportSWF3To6Video`,
 `ExportSWF7PlusVideo`,
 `ExportSWFVectorVideo`,
 `ExportFLVVideo` }
 Output file types for use in calls to `Flix2_SetExportVideoType()` and `Flix2_GetExportVideoType()`.
- enum `FE2_EncState` {
 `EncStateIdle`,
 `EncStateRunning`,
 `EncStateQueued` }
 Encoder state returned by `Flix2_GetEncoderState()`.
- enum `FE2_errno` {
 `ErrNone`,
 `ErrSys`,
 `ErrFileIO`,
 `ErrFileOpen`,
 `ErrFileDecode`,
 `ErrFileDecodeA`,
 `ErrFileDecodeV`,
 `ErrEncodeA`,
 `ErrEncodeV` }
 Flix engine error state returned by `Flix2_Errno()`.

Functions

- `on2sc Flix2_Create (FLIX2HANDLE *pFlix)`
 Create a handle to the flix engine.
- `on2sc Flix2_Destroy (FLIX2HANDLE flix)`
 Frees resources associated with a `FLIX2HANDLE`.
- `const char * Flix2_Version ()`
 Returns the library version as a string.
- `const on2tc * Flix2_Copyright ()`
 Returns copyright information for this library as a string.
- `on2sc Flix2_SetOutputFile (FLIX2HANDLE flix, const on2tc *outputFile)`
 Set the destination file for the encode session.

- `on2sc Flix2_GetOutputFile` (const `FLIX2HANDLE` flix, `on2tc` *pOutputFile, `int32_t` *len)
Get the destination file for the encode session.
- `on2sc Flix2_SetOverwriteExistingFiles` (`FLIX2HANDLE` flix, const `on2bool` bOverwriteExistingFiles)
Enable/disable overwriting of existing output files.
- `on2sc Flix2_GetOverwriteExistingFiles` (const `FLIX2HANDLE` flix, `on2bool` *bpOverwriteExistingFiles)
Retrieve the engine's current behavior regarding existing output files.
- `on2sc Flix2_SetExportAudio` (`FLIX2HANDLE` flix, const `on2bool` bExportAudio)
Enable/disable audio output in the current encoding session.
- `on2sc Flix2_GetExportAudio` (const `FLIX2HANDLE` flix, `on2bool` *bpExportAudio)
Retrieve the engine's current behavior regarding audio output.
- `on2sc Flix2_SetExportVideo` (`FLIX2HANDLE` flix, const `on2bool` bExportVideo)
Enable/disable video output in the current encoding session.
- `on2sc Flix2_GetExportVideo` (const `FLIX2HANDLE` flix, `on2bool` *bpExportVideo)
Retrieve the engine's current behavior regarding video output.
- `on2sc Flix2_SetExportVideoType` (`FLIX2HANDLE` flix, const `FE2_ExportedVideoType` exportVideoType)
Set the output file type.
- `on2sc Flix2_GetExportVideoType` (const `FLIX2HANDLE` flix, `FE2_ExportedVideoType` *pExportVideoType)
Retrieve the current output file type.
- `on2sc Flix2_SetInputFile` (`FLIX2HANDLE` flix, const `on2tc` *inputFile)
Set the source file for the encode session.
- `on2sc Flix2_GetInputFile` (const `FLIX2HANDLE` flix, `on2tc` *pInputFile, `int32_t` *len)
Get the source file for the encode session.
- `on2sc Flix2_GetSourceDuration` (const `FLIX2HANDLE` flix, `int32_t` *duration)
Get the duration, in milliseconds, of the source file.
- `on2sc Flix2_Encode` (`FLIX2HANDLE` flix)
Start encoding an output file.
- `on2sc Flix2_StopEncoding` (`FLIX2HANDLE` flix)
Cancel a running encoding session.
- `on2sc Flix2_Reset` (`FLIX2HANDLE` flix)
Reset the engine to its defaults.

- [on2sc Flix2_Validate](#) (const [FLIX2HANDLE](#) flix)
Validate the current encoder settings.
- [on2sc Flix2_IsEncoderRunning](#) (const [FLIX2HANDLE](#) flix, [on2bool](#) *bpIsEncoderRunning)
Check the status of an encode.
- [on2sc Flix2_GetEncoderState](#) (const [FLIX2HANDLE](#) flix, [FE2_EncState](#) *pEncState)
Retrieve the current state of the encoder.
- [on2sc Flix2_Errno](#) (const [FLIX2HANDLE](#) flix, [FE2_errno](#) *flixereno, [int32_t](#) *syserrno)
Retrieve the current error state of the engine.

12.38.1 Detailed Description

LINUX

12.38.2 Enumeration Type Documentation

12.38.2.1 enum FE2_EncState

Encoder state returned by [Flix2_GetEncoderState\(\)](#).

Enumerator:

EncStateIdle the encoder is not running
EncStateRunning the encoder is running
EncStateQueued the session has been queued until space is available

Definition at line 101 of file flixengine2.h.

12.38.2.2 enum FE2_errno

Flix engine error state returned by [Flix2_Errno\(\)](#).

Enumerator:

ErrNone internal error was not set by the library
ErrSys generic system error
ErrFileIO generic file io error
ErrFileOpen error opening file
ErrFileDecode
ErrFileDecodeA
ErrFileDecodeV
ErrEncodeA
ErrEncodeV

Definition at line 108 of file flixengine2.h.

12.38.2.3 enum FE2_ExportedVideoType

Output file types for use in calls to [Flix2_SetExportVideoType\(\)](#) and [Flix2_GetExportVideoType\(\)](#).

Deprecated

Use the [Muxer Interface](#). This enumeration will be removed in a future release. See also: [Muxers](#).

Enumerator:

ExportSWF3To6Video

ExportSWF7PlusVideo

ExportSWFVectorVideo

ExportFLVVideo

Definition at line 93 of file flxengine2.h.

12.38.3 Function Documentation

12.38.3.1 const on2tc* Flix2_Copyright ()

Returns copyright information for this library as a string.

12.38.3.2 on2sc Flix2_Create (FLIX2HANDLE * *pFlix*)

Create a handle to the flx engine. Initializes a FLIX2HANDLE for use in subsequent FLIXENGINE_API calls.

Parameters:

→ *pFlix* storage location to receive the value of the created FLIX2HANDLE

Return values:

[ON2_OK](#) the engine successfully created and initialized a FLIX2HANDLE for use in FLIXENGINE_API functions.

[ON2_INVALID_PARAMS](#) should the precondition fail

[ON2_NO_MEM](#) memory could not be allocated for the FLIX2HANDLE

Precondition:

pFlix is not NULL

12.38.3.3 on2sc Flix2_Destroy (FLIX2HANDLE *flx*)

Frees resources associated with a FLIX2HANDLE.

Parameters:

← *flx* FLIXHANDLE to be destroyed

Return values:

[*ON2_OK*](#) the engine successfully destroyed the handle

[*ON2_INVALID_PARAMS*](#) should the precondition fail

Precondition:

flix is not NULL

Remarks:

If an encoding session started with [*Flix2_Encode\(\)*](#) is still running when this function is called. It will be stopped with [*Flix2_StopEncoding\(\)*](#) before destroying the handle.

12.38.3.4 on2sc Flix2_Encode (FLIX2HANDLE *flix*)

Start encoding an output file. Encode the input file set in [*Flix2_SetInputFile\(\)*](#) to the output file set in [*Flix2_SetOutputFile\(\)*](#), using the options set in the accessor functions or the engine defaults where applicable. Before starting the encode session, calls [*Flix2_Validate\(\)*](#) to ensure the engine is ready to encode the input.

Parameters:

← *flix* Handle to the flix engine returned from [*Flix2_Create\(\)*](#) or [*Flix2_CreateEx\(\)*](#)

Return values:

[*ON2_OK*](#) the engine successfully started the encode

[*ON2_INVALID_PARAMS*](#) should the precondition fail

<[*ON2_OK*](#) [*Flix2_Validate\(\)*](#) failed / there was an error starting the encode session. Call [*Flix2_Errno\(\)*](#) for more detail.

Precondition:

flix is not NULL

Note:

Detail about the state of the encoder can be obtained by calling [*Flix2_IsEncoderRunning\(\)*](#), [*Flix2_GetEncoderState\(\)*](#) or any of the [Encoding Statistics](#) functions

12.38.3.5 on2sc Flix2_Errno (const FLIX2HANDLE *flix*, FE2_errno * *flixerrno*, int32_t * *syserrno*)

Retrieve the current error state of the engine. If an API call has returned an error, this function can be used to retrieve a more specific reason.

Parameters:

← *flix* Handle to the flix engine returned from [*Flix2_Create\(\)*](#) or [*Flix2_CreateEx\(\)*](#)

→ *flixerrno* Storage location to receive the engine error

→ *syserrno* Storage location to receive the system wide error

Return values:*ON2_OK* on success*ON2_INVALID_PARAMS* should the precondition fail**Precondition:***flix* is not NULL**Remarks:***syserrno* is the value of `GetLastError()`**12.38.3.6 on2sc Flix2_GetEncoderState (const FLIX2HANDLE *flix*, FE2_EncState * *pEncState*)**

Retrieve the current state of the encoder. If using the function to check for encode completion: when *EncStateIdle* is returned, the completion status can be obtained by calling *Flix2_Errno()*

Parameters:← *flix* Handle to the flix engine returned from *Flix2_Create()* or *Flix2_CreateEx()*→ *pEncState* Storage location to receive the result**Return values:***ON2_OK* on success*ON2_INVALID_PARAMS* should one or more of the preconditions fail**Precondition:***flix* is not NULL*pEncState* is not NULL**12.38.3.7 on2sc Flix2_GetExportAudio (const FLIX2HANDLE *flix*, on2bool * *bpExportAudio*)**

Retrieve the engine's current behavior regarding audio output.

Parameters:← *flix* Handle to the flix engine returned from *Flix2_Create()* or *Flix2_CreateEx()*→ *bpExportAudio* Storage location to receive the result**Return values:***ON2_OK* on success*ON2_INVALID_PARAMS* should one or more of the preconditions fail**Precondition:***flix* is not NULL*bpExportAudio* is not NULL**Note:**Default: *on2true*

12.38.3.8 on2sc Flix2_GetExportVideo (const FLIX2HANDLE *flix*, on2bool * *bpExportVideo*)

Retrieve the engine's current behavior regarding video output.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *bpExportVideo* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- bpExportVideo* is not NULL

Note:

Default: [on2true](#)

12.38.3.9 on2sc Flix2_GetExportVideoType (const FLIX2HANDLE *flix*, FE2_ExportedVideoType * *pExportVideoType*)

Retrieve the current output file type.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pExportVideoType* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pExportVideoType* is not NULL

Note:

Default: [ExportFLVVideo](#)

Deprecated

Use the [Muxer Interface](#). This function will be removed in a future release. See also: [Muxers](#).

12.38.3.10 on2sc Flix2_GetInputFile (const FLIX2HANDLE *flx*, on2tc * *pInputFile*, int32_t * *len*)

Get the source file for the encode session. Returns the input file name.

If *pInputFile* is NULL this function will return the size in bytes required to store the current input file in *len*, not including the null terminator.

Parameters:

- ← *flx* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pInputFile* Storage location for the input file path
- ↔ *len* on input, the max number of bytes available in *pInputFile*. On return, the number of bytes used.

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flx* is not NULL
- len* is not NULL

Attention:

Currently only supported in C/C++

12.38.3.11 on2sc Flix2_GetOutputFile (const FLIX2HANDLE *flx*, on2tc * *pOutputFile*, int32_t * *len*)

Get the destination file for the encode session. Returns the (possibly modified see [Flix2_SetOutputFile\(\)](#)) output file name.

If *pOutputFile* is NULL this function will return the size in bytes required to store the current output file in *len*, not including the null terminator.

Parameters:

- ← *flx* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pOutputFile* Storage location for the output file path
- ↔ *len* on input, the max number of bytes available in *pOutputFile*. On return, the number of bytes used.

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flx* is not NULL
- len* is not NULL

Attention:

Currently only supported in C/C++

12.38.3.12 on2sc Flix2_GetOverwriteExistingFiles (const FLIX2HANDLE *flix*, on2bool * *bpOverwriteExistingFiles*)

Retrieve the engine's current behavior regarding existing output files.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *bpOverwriteExistingFiles* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- bpOverwriteExistingFiles* is not NULL

Note:

Default: [on2false](#)

12.38.3.13 on2sc Flix2_GetSourceDuration (const FLIX2HANDLE *flix*, int32_t * *duration*)

Get the duration, in milliseconds, of the source file.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *duration* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- duration* is not NULL

Note:

This value is only valid after a call to [Flix2_SetInputFile\(\)](#). If the duration for the file could not be determined *duration* will be set to -1.

12.38.3.14 on2sc Flix2_IsEncoderRunning (const FLIX2HANDLE *flix*, on2bool * *bpIsEncoderRunning*)

Check the status of an encode. Returns whether an encode session started with [Flix2_Encode\(\)](#) is still running. If the function returns [on2false](#) the completion status can be obtained by calling [Flix2_Errno\(\)](#)

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or `Flix2_CreateEx()`
- *bpIsEncoderRunning* Storage location to receive the result

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

12.38.3.15 on2sc Flix2_Reset (FLIX2HANDLE *flix*)

Reset the engine to its defaults. Resets all encoding related options to their default values. Input and output files are retained.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or `Flix2_CreateEx()`

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

12.38.3.16 on2sc Flix2_SetExportAudio (FLIX2HANDLE *flix*, const on2bool *bExportAudio*)

Enable/disable audio output in the current encoding session.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or `Flix2_CreateEx()`
- ← *bExportAudio* Enable/Disable

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

Note:

Default: [on2true](#)

12.38.3.17 on2sc Flix2_SetExportVideo (FLIX2HANDLE *flix*, const on2bool *bExportVideo*)

Enable/disable video output in the current encoding session.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bExportVideo* Enable/Disable

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

Note:

Default: [on2true](#)

12.38.3.18 on2sc Flix2_SetExportVideoType (FLIX2HANDLE *flix*, const FE2_ExportedVideoType *exportVideoType*)

Set the output file type.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *exportVideoType* File type to output

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NOT_SUPP](#) if the video type is unsupported by the engine

Precondition:

flix is not NULL
exportVideoType is a valid member of [FE2_ExportedVideoType](#)

Attention:

[ExportSWFVectorVideo](#) is unsupported by this version of the engine

Note:

Default: [ExportFLVVideo](#)
When using [ExportSWF7PlusVideo](#) the video codec should be set to [CODEC_H263](#) with [video_options_SetVideoCodec\(\)](#)

Deprecated

Use the [Muxer Interface](#). This function will be removed in a future release. See also: [Muxers](#).

12.38.3.19 on2sc Flix2_SetInputFile (FLIX2HANDLE *flix*, const on2tc * *inputFile*)

Set the source file for the encode session. Attempts to open the input file, extracting the available source information, e.g. video width/height, duration. Additionally configures the encoder to match the source attributes by calling [video_options_Reset\(\)](#) and [audio_options_Reset\(\)](#). For this reason this function should be called before setting any encoder options.

Parameters:

- ← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *inputFile* Path to the source file.

Return values:

- [ON2_OK](#) the engine successfully opened the input file for decoding
- [ON2_INVALID_PARAMS](#) a parameter is invalid.
- <[ON2_OK](#) An error occurred opening the file. Call [Flix2_Errno\(\)](#) for more detail.

Precondition:

- flix* is not NULL
- inputFile* is not NULL

12.38.3.20 on2sc Flix2_SetOutputFile (FLIX2HANDLE *flix*, const on2tc * *outputFile*)

Set the destination file for the encode session. When [Flix2_Encode\(\)](#) is called an attempt is made to open *outputFile*.

If the file exists the behavior is determined by the value of [Flix2_GetOverwriteExistingFiles\(\)](#):

- [on2true](#) the file will be overwritten
- [on2false](#) a new file will be created of the form: *outputFilennn.ext*, e.g. out000.flv

Parameters:

- ← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *outputFile* Path to the output file.

Return values:

- [ON2_OK](#) the engine successfully set the output file
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NO_MEM](#) an error occurred allocating memory for *outputFile*

Precondition:

- flix* is not NULL
- outputFile* is not NULL

12.38.3.21 on2sc Flix2_SetOverwriteExistingFiles (FLIX2HANDLE *flix*, const on2bool *bOverwriteExistingFiles*)

Enable/disable overwriting of existing output files. The engine's behavior based on this setting is described by [Flix2_SetOutputFile\(\)](#)

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bOverwriteExistingFiles* Enable/Disable

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

Note:

Default: [on2false](#)

12.38.3.22 on2sc Flix2_StopEncoding (FLIX2HANDLE *flix*)

Cancel a running encoding session. Stops an encoding session started by [Flix2_Encode\(\)](#). The output file will be removed.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

- [ON2_OK](#) the engine successfully stopped the encode
- [ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

flix is not NULL

12.38.3.23 on2sc Flix2_Validate (const FLIX2HANDLE *flix*)

Validate the current encoder settings. Ensures that the current options set in the engine will allow for an encoding session to be started.

The validation steps performed are:

- input file has been set
- at least one of [Flix2_GetExportVideo\(\)](#) and [Flix2_GetExportAudio\(\)](#) is set to [on2true](#)
- the export video type is [ExportFLVVideo](#) or [ExportSWF7PlusVideo](#)

- if `Flix2_GetExportVideo()` is `on2true`, calls `video_options_Validate()`
- if `Flix2_GetExportAudio()` is `on2true`, calls `audio_options_Validate()`

Parameters:

← *flx* Handle to the flx engine returned from `Flix2_Create()` or `Flix2_CreateEx()`

Return values:

ON2_OK on success

ON2_ERROR on failure

ON2_INVALID_PARAMS should the precondition fail

Precondition:

flx is not NULL

12.38.3.24 const char* Flix2_Version ()

Returns the library version as a string.

12.39 Filter Manipulation

Modules

- [Deprecated](#)

Functions

- [on2sc Flix2_AddFilter](#) ([FLIX2PLGNHANDLE](#) *pPlgn, const [FLIX2HANDLE](#) flx, const char *plgn_name)
Add a filter to the encoder's filter chain.
- [on2sc Flix2_RemoveFilter](#) ([FLIX2PLGNHANDLE](#) plgn)
Remove a filter to the encoder's filter chain.
- [on2sc Flix2_FilterSetParamAsStr](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, const [on2tc](#) *value)
Set the value of a parameter in a filter instance using a string representation.
- [on2sc Flix2_FilterGetParamAsStr](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, [on2tc](#) *value, [int32_t](#) *len)
Retrieve the value of a parameter in a filter instance represented as a string.
- [on2sc Flix2_FilterSetParam](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, double inDbfVal)
Set the value of a parameter in a filter instance using a double representation.
- [on2sc Flix2_FilterGetParam](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, double *outDbfVal)
Retrieve the value of a parameter in a filter instance represented as a double.

12.39.1 Function Documentation

12.39.1.1 [on2sc Flix2_AddFilter](#) ([FLIX2PLGNHANDLE](#) *pPlgn, const [FLIX2HANDLE](#) flx, const char *plgn_name)

Add a filter to the encoder's filter chain. Initializes a [FLIX2PLGNHANDLE](#) mapped to plgn_name for use in subsequent filter calls.

Parameters:

- *pPlgn* Storage location to receive the value of the created [FLIX2PLGNHANDLE](#)
- ← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *plgn_name* Name of the filter to add to the chain

Return values:

- [ON2_OK](#) the engine successfully added the filter and initialized a [FLIX2PLGNHANDLE](#) for use in filter related functions.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NO_MEM](#) memory could not be allocated for the [FLIX2PLGNHANDLE](#)

Precondition:

pPlgn is not NULL
flix is not NULL
plgn_name is not NULL

See also:

[Filters](#)

12.39.1.2 on2sc Flix2_FilterGetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double * *outDblVal*)

Retrieve the value of a parameter in a filter instance represented as a double.

Parameters:

← *plgn* Handle to the filter returned from [Flix2_AddFilter\(\)](#)
 ← *name* Name of the parameter to retrieve
 → *outDblVal* Storage location to receive the value

Return values:

[ON2_OK](#) on success
[ON2_INVALID_PARAMS](#) should the precondition fail
[ON2_NOT_SUPP](#) the filter does not support this parameter/representation

Precondition:

plgn is not NULL
outDblVal is not NULL

See also:

[Filters](#)

12.39.1.3 on2sc Flix2_FilterGetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, on2tc * *value*, int32_t * *len*)

Retrieve the value of a parameter in a filter instance represented as a string.

Parameters:

← *plgn* Handle to the filter returned from [Flix2_AddFilter\(\)](#)
 ← *name* Name of the parameter to retrieve
 → *value* Storage location to receive the value
 ↔ *len* Pointer to buffer length variable. Should contain the maximum size of the buffer. This call updates this variable with the size of the returned string.

Return values:

[ON2_OK](#) on success

[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail

[*ON2_NOT_SUPP*](#) the filter does not support this parameter

Precondition:

plgn is not NULL
value is not NULL
len is not NULL

Attention:

This function is currently unsupported and will return [*ON2_NOT_SUPP*](#).

See also:

[Filters](#)

12.39.1.4 on2sc Flix2_FilterSetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double *inDblVal*)

Set the value of a parameter in a filter instance using a double representation.

Parameters:

← *plgn* Handle to the filter returned from [Flix2_AddFilter\(\)](#)
← *name* Name of the parameter to set
← *inDblVal* New value for the parameter

Return values:

[*ON2_OK*](#) on success
[*ON2_INVALID_PARAMS*](#) should the precondition fail
[*ON2_NOT_SUPP*](#) the filter does not support this parameter

Precondition:

plgn is not NULL

See also:

[Filters](#)

12.39.1.5 on2sc Flix2_FilterSetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, const on2tc * *value*)

Set the value of a parameter in a filter instance using a string representation.

Parameters:

← *plgn* Handle to the filter returned from [Flix2_AddFilter\(\)](#)
← *name* Name of the parameter to set
← *value* New value for the parameter

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should the precondition fail

[*ON2_NOT_SUPP*](#) the filter does not support this parameter

Precondition:

plgn is not NULL

See also:

[Filters](#)

12.39.1.6 on2sc Flix2_RemoveFilter (FLIX2PLGNHANDLE *plgn*)

Remove a filter to the encoder's filter chain.

Parameters:

← *plgn* Handle to the filter returned from [Flix2_AddFilter\(\)](#)

Return values:

[*ON2_OK*](#) on success

[*ON2_INVALID_PARAMS*](#) should the precondition fail

Precondition:

plgn is not NULL

12.40 Codec Manipulation

Functions

- **on2sc Flix2_AddCodec** (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)
Add a codec to the encoder's codec chain.
- **on2sc Flix2_RemoveCodec** (FLIX2PLGNHANDLE plgn)
Remove a codec from the encoder's codec chain.
- **on2sc Flix2_CodecSetParamAsStr** (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)
Set the value of a parameter in a codec instance using a string representation.
- **on2sc Flix2_CodecGetParamAsStr** (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)
Retrieve the value of a parameter in a codec instance represented as a string.
- **on2sc Flix2_CodecSetParam** (FLIX2PLGNHANDLE plgn, const char *name, double inDbfVal)
Set the value of a parameter in a codec instance using a double representation.
- **on2sc Flix2_CodecGetParam** (FLIX2PLGNHANDLE plgn, const char *name, double *outDbfVal)
Retrieve the value of a parameter in a codec instance represented as a double.

12.40.1 Function Documentation

12.40.1.1 on2sc Flix2_AddCodec (FLIX2PLGNHANDLE * pPlgn, const FLIX2HANDLE flix, const char * plgn_name)

Add a codec to the encoder's codec chain. Initializes a FLIX2PLGNHANDLE mapped to plgn_name for use in subsequent codec calls.

Parameters:

- **pPlgn** Storage location to receive the value of the created FLIX2PLGNHANDLE
- ← **flix** Handle to the flix engine returned from **Flix2_Create()** or **Flix2_CreateEx()**
- ← **plgn_name** Name of the codec to add to the chain

Return values:

- ON2_OK** the engine successfully added the codec and initialized a FLIX2PLGNHANDLE for use in codec related functions.
- ON2_INVALID_PARAMS** should one or more of the preconditions fail
- ON2_NO_MEM** memory could not be allocated for the FLIX2PLGNHANDLE

Precondition:

- pPlgn* is not NULL
- flix* is not NULL
- plgn_name* is not NULL

See also:

[Codecs](#)

12.40.1.2 on2sc Flix2_CodecGetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double * *outDbfVal*)

Retrieve the value of a parameter in a codec instance represented as a double.

Parameters:

- ← *plgn* Handle to the codec returned from [Flix2_AddCodec\(\)](#)
- ← *name* Name of the parameter to retrieve
- *outDbfVal* Storage location to receive the value

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail
- [ON2_NOT_SUPP](#) the codec does not support this parameter/representation

Precondition:

- plgn* is not NULL
- outDbfVal* is not NULL

See also:

[Codecs](#)

12.40.1.3 on2sc Flix2_CodecGetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, on2tc * *value*, int32_t * *len*)

Retrieve the value of a parameter in a codec instance represented as a string.

Parameters:

- ← *plgn* Handle to the codec returned from [Flix2_AddCodec\(\)](#)
- ← *name* Name of the parameter to retrieve
- *value* Storage location to receive the value
- ↔ *len* Pointer to buffer length variable. Should contain the maximum size of the buffer. This call updates this variable with the size of the returned string.

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NOT_SUPP](#) the codec does not support this parameter

Precondition:

- plgn* is not NULL
- value* is not NULL
- len* is not NULL

Attention:

This function is currently unsupported and will return [ON2_NOT_SUPP](#).

See also:

[Codecs](#)

12.40.1.4 on2sc Flix2_CodecSetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double *inDbfVal*)

Set the value of a parameter in a codec instance using a double representation.

Parameters:

- ← *plgn* Handle to the codec returned from [Flix2_AddCodec\(\)](#)
- ← *name* Name of the parameter to set
- ← *inDbfVal* New value for the parameter

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail
- [ON2_NOT_SUPP](#) the codec does not support this parameter

Precondition:

plgn is not NULL

See also:

[Codecs](#)

12.40.1.5 on2sc Flix2_CodecSetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, const on2tc * *value*)

Set the value of a parameter in a codec instance using a string representation.

Parameters:

- ← *plgn* Handle to the codec returned from [Flix2_AddCodec\(\)](#)
- ← *name* Name of the parameter to set
- ← *value* New value for the parameter

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail
- [ON2_NOT_SUPP](#) the codec does not support this parameter

Precondition:

plgn is not NULL

See also:

[Codecs](#)

12.40.1.6 on2sc Flix2_RemoveCodec (FLIX2PLGNHANDLE *plgn*)

Remove a codec from the encoder's codec chain.

Parameters:

← *plgn* Handle to the codec returned from [Flix2_AddCodec\(\)](#)

Return values:

[ON2_OK](#) on success

[ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

plgn is not NULL

12.41 Muxer Manipulation

Functions

- **on2sc Flix2_AddMuxer** (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)
Set the desired muxer.
- **on2sc Flix2_RemoveMuxer** (FLIX2PLGNHANDLE plgn)
Remove the specified muxer.
- **on2sc Flix2_MuxerSetParamAsStr** (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)
Set the value of a parameter in a muxer instance using a string representation.
- **on2sc Flix2_MuxerGetParamAsStr** (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)
Retrieve the value of a parameter in a muxer instance represented as a string.
- **on2sc Flix2_MuxerSetParam** (FLIX2PLGNHANDLE plgn, const char *name, double inDbfVal)
Set the value of a parameter in a muxer instance using a double representation.
- **on2sc Flix2_MuxerGetParam** (FLIX2PLGNHANDLE plgn, const char *name, double *outDbfVal)
Retrieve the value of a parameter in a muxer instance represented as a double.

12.41.1 Function Documentation

12.41.1.1 on2sc Flix2_AddMuxer (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)

Set the desired muxer. Initializes a FLIX2PLGNHANDLE mapped to plgn_name for use in subsequent muxer calls.

Parameters:

- **pPlgn** Storage location to receive the value of the created FLIX2PLGNHANDLE
- ← **flix** Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← **plgn_name** Name of the desired muxer

Return values:

- ON2_OK** the engine successfully added the muxer and initialized a FLIX2PLGNHANDLE for use in muxer related functions.
- ON2_INVALID_PARAMS** should one or more of the preconditions fail
- ON2_NO_MEM** memory could not be allocated for the FLIX2PLGNHANDLE

Precondition:

- pPlgn* is not NULL
- flix* is not NULL
- plgn_name* is not NULL

See also:

[Muxers](#)

12.41.1.2 on2sc Flix2_MuxerGetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double * *outDbfVal*)

Retrieve the value of a parameter in a muxer instance represented as a double.

Parameters:

- ← *plgn* Handle to the muxer returned from [Flix2_AddMuxer\(\)](#)
- ← *name* Name of the parameter to retrieve
- *outDbfVal* Storage location to receive the value

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail
- [ON2_NOT_SUPP](#) the muxer does not support this parameter/representation

Precondition:

- plgn* is not NULL
- outDbfVal* is not NULL

See also:

[Codecs](#)

12.41.1.3 on2sc Flix2_MuxerGetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, on2tc * *value*, int32_t * *len*)

Retrieve the value of a parameter in a muxer instance represented as a string.

Parameters:

- ← *plgn* Handle to the muxer returned from [Flix2_AddMuxer\(\)](#)
- ← *name* Name of the parameter to retrieve
- *value* Storage location to receive the value
- ↔ *len* Pointer to buffer length variable. Should contain the maximum size of the buffer. This call updates this variable with the size of the returned string.

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NOT_SUPP](#) the muxer does not support this parameter

Precondition:

- plgn* is not NULL
- value* is not NULL
- len* is not NULL

Attention:

This function is currently unsupported and will return [ON2_NOT_SUPP](#).

See also:

[muxer](#)

12.41.1.4 on2sc Flix2_MuxerSetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double *inDbIVal*)

Set the value of a parameter in a muxer instance using a double representation.

Parameters:

- ← *plgn* Handle to the muxer returned from [Flix2_AddMuxer\(\)](#)
- ← *name* Name of the parameter to set
- ← *inDbIVal* New value for the parameter

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail
- [ON2_NOT_SUPP](#) the muxer does not support this parameter

Precondition:

plgn is not NULL

See also:

[Muxers](#)

12.41.1.5 on2sc Flix2_MuxerSetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, const on2tc * *value*)

Set the value of a parameter in a muxer instance using a string representation.

Parameters:

- ← *plgn* Handle to the muxer returned from [Flix2_AddMuxer\(\)](#)
- ← *name* Name of the parameter to set
- ← *value* New value for the parameter

Return values:

- [ON2_OK](#) on success
- [ON2_INVALID_PARAMS](#) should the precondition fail
- [ON2_NOT_SUPP](#) the muxer does not support this parameter

Precondition:

plgn is not NULL

See also:

[Muxers](#)

12.41.1.6 on2sc Flix2_RemoveMuxer (FLIX2PLGNHANDLE *plgn*)

Remove the specified muxer.

Parameters:

← *plgn* Handle to the muxer returned from [Flix2_AddMuxer\(\)](#)

Return values:

[ON2_OK](#) on success

[ON2_INVALID_PARAMS](#) should the precondition fail

Precondition:

plgn is not NULL

12.42 Deprecated

Functions

- [on2sc editor_options_Reset](#) (FLIX2HANDLE *flix*)
Reset the media editor options to their defaults.
- [on2sc editor_options_Validate](#) (const FLIX2HANDLE *flix*)
Ensure the current media editor settings are valid.

12.42.1 Function Documentation

12.42.1.1 on2sc editor_options_Reset (FLIX2HANDLE *flix*)

Reset the media editor options to their defaults.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

[ON2_OK](#) The editor options were reset successfully.

Deprecated

Use the [Filter Interface](#)

12.42.1.2 on2sc editor_options_Validate (const FLIX2HANDLE *flix*)

Ensure the current media editor settings are valid.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

[ON2_OK](#) The editor options are valid and encoding may proceed.

[ON2_INVALID_PARAMS](#) One or more options are invalid. Proceeding with encoding may cause encoding to fail or a file that does not match the specified options will be created.

Deprecated

Use the [Filter Interface](#)

12.43 Muxers

Modules

- [FLV](#)
- [FXM](#)
- [3GPP - FFmpeg](#)
- [3GPP2 - FFmpeg](#)
- [MOV - FFmpeg](#)
- [MP4 - FFmpeg](#)
- [SWF](#)
- [WebM - FFmpeg](#)

12.43.1 Detailed Description

Muxers are configurable both by the [muxer interface](#) and individual functions (e.g. `(swf|video)_options_-SetXXX`). The latter, however, are deprecated and will be removed in a future release. Therefore, any new code should use the [muxer interface](#).

12.44 FLV

Defines

- #define [FE2_MUXER_FLV](#)
FLV muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_FLV_CUEPT_EVENT](#)
Set an event cue point.
- #define [FE2_FLV_CUEPT_NAV](#)
Set a navigation cue point.
- #define [FE2_FLV_CUEPT_PARAM](#)
Add a name/value pair to an existing cue point.
- #define [FE2_FLV_METADATA_ENABLE](#)
Enable output of meta data element.
- #define [FE2_FLV_METADATA_DISABLE](#)
Disable output of meta data element.

Typedefs

- typedef enum [flv_metadata](#) [flvmetadata_t](#)
Supported FLV onMetaData elements.

Enumerations

- enum [flv_metadata](#) {
 [MD_DURATION](#),
 [MD_DATASIZE](#),
 [MD_AUDIO_SIZE](#),
 [MD_VIDEO_SIZE](#),
 [MD_AUDIO_DATARATE](#),
 [MD_VIDEO_DATARATE](#),
 [MD_AUDIO_CODECID](#),
 [MD_VIDEO_CODECID](#),
 [MD_WIDTH](#),
 [MD_HEIGHT](#),
 [MD_FRAMERATE](#),
 [MD_CANSEEKTOEND](#),
 [MD_LASTTIMESTAMP](#),
 [MD_LASTKEYFRAMETIMESTAMP](#),
};

[MD_LASTKEYFRAMELOCATION](#),
[MD_KEYFRAMES](#) }

Supported FLV onMetaData elements.

12.44.1 Detailed Description

Muxer Parameters:

Name	Type	Opt/Reqd	Range
FE2_FLV_CUEPT_EVENT	String	Optional	[0.0,)
FE2_FLV_CUEPT_NAV	String	Optional	[0.0,)
FE2_FLV_CUEPT_PARAM	String	Optional	N/A
FE2_FLV_METADATA_ENABLE	Numeric	Optional	flvmetadata_t
FE2_FLV_METADATA_DISABLE	Numeric	Optional	flvmetadata_t

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flx, FE2_MUXER_FLV);
// set a navigation cue point at 10s
if(sc == ON2_OK)
    sc = Flix2_MuxerSetParamAsStr(muxer, FE2_FLV_CUEPT_NAV, "nav0=10.0");
```

12.44.2 Supported Codecs

- Video:
 - [FE2_CODEC_H263](#)
 - [FE2_CODEC_VP6](#)
 - [FE2_CODEC_VP6ALPHA](#)
- Audio:
 - [FE2_CODEC_AAC](#)
 - [FE2_CODEC_AACPLUS](#)
 - [FE2_CODEC_LAME](#)
 - [FE2_CODEC_PCM](#)

12.44.3 Format Restrictions

- Audio:
 - [FE2_CODEC_AAC/FE2_CODEC_AACPLUS](#): 44.1kHz & 2ch
 - Others: 44.1/22.05/11.025kHz & 1/2ch

If the input does not meet the above conditions [FE2_FILTER_RESAMPLE](#) will be automatically added, overriding user settings if necessary.

12.44.4 Define Documentation

12.44.4.1 #define FE2_FLV_CUEPT_EVENT

Set an event cue point.

Parameter format:

Format = cueptNAME '=' cueptTIME_SECONDS
e.g., "evtpt0=343.0".

Precondition:

time must be ≥ 0.0

Note:

Multiple cue points may be added.

See also:

<http://livedocs.macromedia.com/flash/8/main/wwhelp/wwhimpl/common/html/wwhelp.htm?Parts&file=00001574.html>

Definition at line 133 of file flv.h.

12.44.4.2 #define FE2_FLV_CUEPT_NAV

Set a navigation cue point. This parameter allows seeking to the specified time by generating an I-Frame (keyframe) in the video stream and adding the entry to the metadata.

Parameter format:

Format = cueptNAME '=' cueptTIME_SECONDS
e.g., "navpt0=343.0".

Precondition:

time must be ≥ 0.0

Note:

Multiple cue points may be added.

See also:

<http://livedocs.macromedia.com/flash/8/main/wwhelp/wwhimpl/common/html/wwhelp.htm?Parts&file=00001574.html>

Definition at line 145 of file flv.h.

12.44.4.3 #define FE2_FLV_CUEPT_PARAM

Add a name/value pair to an existing cue point.

Parameter format:

```
Format = cueptNAME *['&' Pair ]
Pair = Name '=' Value
e.g., "cuept0&name0=value0&name1=value1"
```

Precondition:

The cue point has already been added

Definition at line 154 of file flv.h.

12.44.4.4 #define FE2_FLV_METADATA_DISABLE

Disable output of meta data element.

Note:

Valid elements are defined by [flvmetadata_t](#)

Definition at line 164 of file flv.h.

12.44.4.5 #define FE2_FLV_METADATA_ENABLE

Enable output of meta data element.

Note:

Valid elements are defined by [flvmetadata_t](#)

Definition at line 159 of file flv.h.

12.44.4.6 #define FE2_MUXER_FLV

FLV muxer. For use with [Flix2_AddMuxer\(\)](#).

Definition at line 124 of file flv.h.

12.44.5 Typedef Documentation**12.44.5.1 typedef enum flv_metadata flvmetadata_t**

Supported FLV onMetaData elements. Descriptions indicate the default in bold, followed by the element name in italics, its type and a description of the units if applicable.

Note:

Audio/Video specific entries are disabled should there be no stream of the type

12.44.6 Enumeration Type Documentation**12.44.6.1 enum flv_metadata**

Supported FLV onMetaData elements. Descriptions indicate the default in bold, followed by the element name in italics, its type and a description of the units if applicable.

Note:

Audio/Video specific entries are disabled should there be no stream of the type

Enumerator:

- MD_DURATION Enabled.** *duration* <Number> seconds
- MD_DATASIZE Enabled.** *datasize* <Number> bytes
- MD_AUDIO_SIZE Enabled.** *audiosize* <Number> bytes
- MD_VIDEO_SIZE Enabled.** *videosize* <Number> bytes
- MD_AUDIO_DATARATE Enabled.** *audiodatarate* <Number> kbps
- MD_VIDEO_DATARATE Enabled.** *videodatarate* <Number> kbps
- MD_AUDIO_CODECID Enabled.** *audiocodecid* <Number>
- MD_VIDEO_CODECID Enabled.** *videocodecid* <Number>
- MD_WIDTH Enabled.** *width* <Number>
- MD_HEIGHT Enabled.** *height* <Number>
- MD_FRAMERATE Enabled.** *framerate* <Number> frames/sec
- MD_CANSEEKTOEND Enabled.** *canSeekToEnd* <Boolean> Indicates last video tag is a key frame.
- MD_LASTTIMESTAMP Enabled.** *lasttimestamp* <Number> seconds
- MD_LASTKEYFRAME_TIMESTAMP Disabled.** *lastkeyframetimestamp* <Number> seconds
- MD_LASTKEYFRAMELOCATION Disabled.** *lastkeyframeoffset* <Number> byte offset
- MD_KEYFRAMES Disabled.** *keyframes* <Object> Contains 2 arrays:
- *filepositions*: <Number> byte offset
 - *times*: <Number> timestamp in seconds

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the 'keyframes' object is added, with the result then replacing the original output file.

Definition at line 71 of file flv.h.

12.45 FXM

Defines

- #define [FE2_MUXER_FXM](#)
FXM muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_FXM_CUEPT_EVENT](#)
Set an event cue point.
- #define [FE2_FXM_CUEPT_NAV](#)
Set a navigation cue point.
- #define [FE2_FXM_CUEPT_PARAM](#)
Add a name/value pair to an existing cue point.
- #define [FE2_FXM_METADATA_ENABLE](#)
Enable output of meta data element.
- #define [FE2_FXM_METADATA_DISABLE](#)
Disable output of meta data element.

Typedefs

- typedef enum [flv_metadata](#) [fxmmetadata_t](#)

12.45.1 Detailed Description

The FXM muxer produces output compatible with the Sun Java VM.

Muxer Parameters:

Name	Type	Opt/Reqd	Range
FE2_FXM_CUEPT_EVENT	String	Optional	[0.0,)
FE2_FXM_CUEPT_NAV	String	Optional	[0.0,)
FE2_FXM_CUEPT_PARAM	String	Optional	N/A
FE2_FXM_METADATA_ENABLE	Numeric	Optional	fxmmetadata_t
FE2_FXM_METADATA_DISABLE	Numeric	Optional	fxmmetadata_t

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flx, FE2_MUXER_FXM);
if(sc == ON2_OK)
    ; //now using the FXM muxer for output
```

12.45.2 Supported Codecs

- Video:
 - [FE2_CODEC_VP6](#)

- Audio:
 - [FE2_CODEC_LAME](#)

12.45.3 Format Restrictions

- Audio:
 - 44.1/22.05/11.025kHz & 1/2ch

If the input does not meet the above conditions [FE2_FILTER_RESAMPLE](#) will be automatically added, overriding user settings if necessary.

12.45.4 Define Documentation

12.45.4.1 #define FE2_FXM_CUEPT_EVENT

Set an event cue point.

Parameter format:

Format = cueptNAME '=' cueptTIME_SECONDS
e.g., "evtpt0=343.0".

Precondition:

time must be ≥ 0.0

Note:

Multiple cue points may be added.

Definition at line 70 of file fxm.h.

12.45.4.2 #define FE2_FXM_CUEPT_NAV

Set a navigation cue point. This parameter allows seeking to the specified time by generating an I-Frame (keyframe) in the video stream and adding the entry to the metadata.

Parameter format:

Format = cueptNAME '=' cueptTIME_SECONDS
e.g., "navpt0=343.0".

Precondition:

time must be ≥ 0.0

Note:

Multiple cue points may be added.

Definition at line 81 of file fxm.h.

12.45.4.3 **#define FE2_FXM_CUEPT_PARAM**

Add a name/value pair to an existing cue point.

Parameter format:

```
Format = cueptNAME *['&' Pair ]  
Pair = Name '=' Value  
e.g., "cuept0&name0=value0&name1=value1"
```

Precondition:

The cue point has already been added

Definition at line 90 of file fxm.h.

12.45.4.4 **#define FE2_FXM_METADATA_DISABLE**

Disable output of meta data element.

Note:

Valid elements are defined by [fxmmetadata_t](#)

Definition at line 100 of file fxm.h.

12.45.4.5 **#define FE2_FXM_METADATA_ENABLE**

Enable output of meta data element.

Note:

Valid elements are defined by [fxmmetadata_t](#)

Definition at line 95 of file fxm.h.

12.45.4.6 **#define FE2_MUXER_FXM**

FXM muxer. For use with [Flix2_AddMuxer\(\)](#).

Definition at line 62 of file fxm.h.

12.45.5 Typedef Documentation

12.45.5.1 **typedef enum flv_metadata fxmmetadata_t**

Definition at line 59 of file fxm.h.

12.46 3GPP - FFmpeg

Defines

- `#define FE2_MUXER_3GP`
3GPP muxer. For use with `Flix2_AddMuxer()`
- `#define FE2_3GP_FASTSTART`
`FE2_ISOMEDIA_FASTSTART` alias

12.46.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flx, FE2_MUXER_3GP);  
if(sc == ON2_OK)  
    ; //Now using the 3GPP muxer for output
```

12.46.2 Supported Codecs

- Video:
 - `FE2_CODEC_H263_BASELINE`
 - `FE2_CODEC_H264`
- Audio:
 - `FE2_CODEC_AAC`
 - `FE2_CODEC_AACPLUS`
 - `FE2_CODEC_AMR_NB`

Additional References:

- [3GPP homepage](#)

12.46.3 Define Documentation

12.46.3.1 `#define FE2_3GP_FASTSTART`

`FE2_ISOMEDIA_FASTSTART` alias Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 67 of file isomedia.h.

12.46.3.2 `#define FE2_MUXER_3GP`

3GPP muxer. For use with [Flix2_AddMuxer\(\)](#)

Definition at line 62 of file isomedia.h.

12.47 3GPP2 - FFmpeg

Defines

- `#define FE2_MUXER_3G2`
3GPP2 muxer. For use with [Flix2_AddMuxer\(\)](#)
- `#define FE2_3G2_FASTSTART`
[FE2_ISOMEDIA_FASTSTART](#) alias

12.47.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_3G2);  
if(sc == ON2_OK)  
    ; //Now using the 3GPP2 muxer for output
```

12.47.2 Supported Codecs

- Video:
 - [FE2_CODEC_H263_BASELINE](#)
 - [FE2_CODEC_H264](#)
- Audio:
 - [FE2_CODEC_AAC](#)
 - [FE2_CODEC_AACPLUS](#)
 - [FE2_CODEC_AMR_NB](#)

Additional References:

- [3GPP2 homepage](#)

12.47.3 Define Documentation

12.47.3.1 `#define FE2_3G2_FASTSTART`

[FE2_ISOMEDIA_FASTSTART](#) alias Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 102 of file isomedia.h.

12.47.3.2 `#define FE2_MUXER_3G2`

3GPP2 muxer. For use with [Flix2_AddMuxer\(\)](#)

Definition at line 97 of file isomedia.h.

12.48 MOV - FFmpeg

Defines

- `#define FE2_MUXER_MOV`
MOV muxer. For use with `Flix2_AddMuxer()`.
- `#define FE2_MOV_FASTSTART`
FE2_ISOMEDIA_FASTSTART alias

12.48.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_MOV);  
if(sc == ON2_OK)  
    ; //Now using the MOV muxer for output
```

12.48.2 Supported Codecs

- Video:
 - `FE2_CODEC_H263_BASELINE`
 - `FE2_CODEC_H264`
- Audio:
 - `FE2_CODEC_AAC`
 - `FE2_CODEC_AACPLUS`
 - `FE2_CODEC_AMR_NB`

Additional References:

- [ISO - International Organization for Standardization](#)
- [ISO/IEC 14496-12](#):ISO base media file format
- [QuickTime File Format](#)

Attention:

The current implementation does not write a 'ctts' table (cf., ISO/IEC 14496-12) to the output. This will cause playback under QuickTime to stutter should `FE2_H264_B_FRAME_RATE` be used. Note playback under the Flash Player is unaffected. This will be addressed in a future release.

12.48.3 Define Documentation

12.48.3.1 `#define FE2_MOV_FASTSTART`

`FE2_ISOMEDIA_FASTSTART` alias Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 146 of file isomedia.h.

12.48.3.2 #define FE2_MUXER_MOV

MOV muxer. For use with [Flix2_AddMuxer\(\)](#).

Definition at line 141 of file isomedia.h.

12.49 MP4 - FFmpeg

Defines

- `#define FE2_MUXER_MP4`
MP4 muxer. For use with [Flix2_AddMuxer\(\)](#).
- `#define FE2_MP4_FASTSTART`
[FE2_ISOMEDIA_FASTSTART](#) alias

12.49.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_MP4);  
if(sc == ON2_OK)  
    ; //Now using the MP4 muxer for output
```

12.49.2 Supported Codecs

- Video:
 - [FE2_CODEC_H264](#)
- Audio:
 - [FE2_CODEC_AAC](#)
 - [FE2_CODEC_AACPLUS](#)

Additional References:

- [ISO - International Organization for Standardization](#)
- [ISO/IEC 14496-12](#):ISO base media file format

Attention:

The current implementation does not write a 'ctts' table (cf., ISO/IEC 14496-12) to the output. This will cause playback under QuickTime to stutter should [FE2_H264_B_FRAME_RATE](#) be used. Note playback under the Flash Player is unaffected. This will be addressed in a future release.

See also:

H.264 notes for [Apple device support](#)

12.49.3 Define Documentation

12.49.3.1 `#define FE2_MP4_FASTSTART`

[FE2_ISOMEDIA_FASTSTART](#) alias Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 188 of file isomedia.h.

12.49.3.2 #define FE2_MUXER_MP4

MP4 muxer. For use with [Flix2_AddMuxer\(\)](#).

Definition at line 183 of file isomedia.h.

12.50 SWF

Defines

- #define [FE2_MUXER_SWF](#)
SWF muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_SWF_HEIGHT](#)
Set the SWF height.
- #define [FE2_SWF_WIDTH](#)
Set the SWF width.
- #define [FE2_SWF_FRAMERATE](#)
Set the SWF framerate.
- #define [FE2_SWF_EMBEDDED_URL](#)
Set the SWF's target URL.
- #define [FE2_SWF_EMBEDDED_URL_TARGET](#)
Set the target of [FE2_SWF_EMBEDDED_URL](#).
- #define [FE2_SWF_EMBEDDED_URL_TYPE](#)
Set how [FE2_SWF_EMBEDDED_URL](#) is interpreted.
- #define [FE2_SWF_LOOP_COUNT](#)
Sets the number of times the SWF should loop.
- #define [FE2_SWF_PRELOAD_TYPE](#)
Sets the type of preloader.
- #define [FE2_SWF_FIXED_PRELOAD_PCT](#)
Sets the percent of the SWF movie to preload before playback begins.
- #define [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#)
Sets the adaptive preload buffer factor.
- #define [FE2_SWF_ON_END_OPTION](#)
Sets the options for the end of the SWF.
- #define [FE2_SWF_ON_END_URL](#)
Sets the URL a SWF movie will load after the current movie ends.
- #define [FE2_SWF_ON_START_OPTION](#)
Sets the options for the start of the SWF.
- #define [FE2_SWF_START_BLANK_FRAME](#)
Control the insertion of a blank first frame in the SWF.
- #define [FE2_SWF_START_WAIT_SEC](#)

Sets the number of seconds to wait before playback begins.

- `#define FE2_SWF_ADD_VARIABLE`
Add or update a variable in the SWF.
- `#define FE2_SWF_DELETE_VARIABLE`
Delete an existing variable in the SWF.

Enumerations

- `enum FE2_EmbeddedUrlType {`
 `EmbeddedUrlIsGetUrl,`
 `EmbeddedUrlIsLoadMovie }`
Differentiates between the type of file (HTML or SWF) set through `FE2_SWF_EMBEDDED_URL`.
- `enum FE2_SwfOnEndOptions {`
 `SwfOnMovieEndNothing,`
 `SwfOnMovieEndSTOP,`
 `SwfOnMovieEndLoop,`
 `SwfOnMovieEndUnload,`
 `SwfOnMovieEndLoadMovie }`
Actions that can be added to the last frame of a SWF file.
- `enum FE2_SwfOnStartOptions {`
 `SwfOnMovieStartAutomatically,`
 `SwfOnMovieStartOnClick,`
 `SwfOnMovieStartWait,`
 `SwfOnMovieStartEmbedSTOP }`
Actions that can be added to the start frame of the SWF file.
- `enum FE2_SwfPreloaderOptions {`
 `SwfPreloaderNone,`
 `SwfFixedPreloader,`
 `SwfAdaptivePreloader }`
Determines the type of preloader added to the SWF file.

Deprecated functions

- `on2sc swf_options_Reset (FLIX2HANDLE flx)`
Resets the swf options.
- `on2sc swf_options_GetEmbeddedUrl (FLIX2HANDLE flx, char *embeddedUrl, int32_t *len)`
Gets the embedded URL.

- `on2sc swf_options_GetEmbeddedUrlTarget` (FLIX2HANDLE flix, char *embeddedUrlTarget, int32_t *len)
Gets the target of embedded URL.
- `on2sc swf_options_SetEmbeddedUrl` (FLIX2HANDLE flix, const char *embeddedUrl)
Sets the embedded URL of the SWF or HTML that will be loaded when the user clicks on the video.
- `on2sc swf_options_SetEmbeddedUrlTarget` (FLIX2HANDLE flix, const char *embeddedUrlTarget)
Sets the target of the embedded URL.
- `on2sc swf_options_GetEmbeddedUrlType` (FLIX2HANDLE flix, FE2_EmbeddedUrlType *embeddedUrlType)
Gets the type of embedded URL.
- `on2sc swf_options_SetEmbeddedUrlType` (FLIX2HANDLE flix, const FE2_EmbeddedUrlType embeddedUrlType)
Sets the type of embedded URL.
- `on2sc swf_options_GetSwfFramerate` (FLIX2HANDLE flix, int32_t *pSwfFramerate)
Gets the SWF framerate.
- `on2sc swf_options_SetSwfFramerate` (FLIX2HANDLE flix, const int32_t swfFramerate)
Sets the SWF framerate.
- `on2sc swf_options_GetInsertBlankFrameOnStart` (FLIX2HANDLE flix, on2bool *pInsertBlankFrameOnStart)
Gets if the engine is to insert a blank frame as the first frame of the SWF or not.
- `on2sc swf_options_SetInsertBlankFrameOnStart` (FLIX2HANDLE flix, const on2bool insertBlankFrameOnStart)
Tells the engine to insert a blank frame as the first frame of the SWF or not.
- `on2sc swf_options_GetSwfFramerateAsDouble` (FLIX2HANDLE flix, double *pSwfFramerate)
Gets the SWF framerate as a double.
- `on2sc swf_options_SetSwfFramerateAsDouble` (FLIX2HANDLE flix, const double swfFramerate)
Sets the SWF framerate as a double.
- `on2sc swf_options_GetEnablePreloader` (FLIX2HANDLE flix, on2bool *pEnablePreloader)
Gets if a video preloader is enabled or disabled.
- `on2sc swf_options_SetEnablePreloader` (FLIX2HANDLE flix, const on2bool enablePreloader)
Enables or disables the video preloader.
- `on2sc swf_options_GetPercentToPreload` (FLIX2HANDLE flix, int32_t *pPercentToPreload)
Gets the percent of the SWF movie to preload before playback begins.
- `on2sc swf_options_SetPercentToPreload` (FLIX2HANDLE flix, const int32_t percentToPreload)

Sets the percent of the SWF movie to preload before playback begins.

- `on2sc swf_options_GetPreloaderType (FLIX2HANDLE flx, FE2_SwfPreloaderOptions *pPreloaderType)`

Gets the type of preloader.

- `on2sc swf_options_SetPreloaderType (FLIX2HANDLE flx, const FE2_SwfPreloaderOptions preloaderType)`

Sets the type of preloader.

- `on2sc swf_options_GetAdaptivePreloaderBufferFactor (FLIX2HANDLE flx, double *pPreloaderBufferFactor)`

Gets the adaptive preload buffer factor.

- `on2sc swf_options_SetAdaptivePreloaderBufferFactor (FLIX2HANDLE flx, const double preloaderBufferFactor)`

Sets the adaptive preload buffer factor.

- `on2sc swf_options_GetMovieOnEndOptions (FLIX2HANDLE flx, FE2_SwfOnEndOptions *pOnEndOptions)`

Gets the options for the end of the SWF.

- `on2sc swf_options_SetMovieOnEndOptions (FLIX2HANDLE flx, const FE2_SwfOnEndOptions onEndOptions)`

Sets the options for the end of the SWF.

- `on2sc swf_options_GetLoopCount (FLIX2HANDLE flx, int32_t *pLoopCount)`

Gets the number of times the SWF should loop.

- `on2sc swf_options_SetLoopCount (FLIX2HANDLE flx, const int32_t loopCount)`

Sets the number of times the SWF should loop.

- `on2sc swf_options_GetLoadMovieOnEndUrl (FLIX2HANDLE flx, char *pLoadMovieOnEndUrl, int32_t *len)`

Gets the URL of a SWF movie to load after the current movie ends.

- `on2sc swf_options_SetLoadMovieOnEndUrl (FLIX2HANDLE flx, const char *loadMovieOnEndUrl)`

Sets the URL of a SWF movie to load after the current movie ends.

- `on2sc swf_options_GetMovieOnStartOptions (FLIX2HANDLE flx, FE2_SwfOnStartOptions *pOnStartOptions)`

Gets the options for the start of the SWF.

- `on2sc swf_options_SetMovieOnStartOptions (FLIX2HANDLE flx, const FE2_SwfOnStartOptions onStartOptions)`

Sets the options for the start of the SWF.

- `on2sc swf_options_GetWaitTimeToStart (FLIX2HANDLE flx, int32_t *pWaitTimeToStart)`

Gets the number of seconds to wait before playback begins.

- `on2sc swf_options_SetWaitTimeToStart` (FLIX2HANDLE flix, const `int32_t` waitTimeToStart)
Sets the number of seconds to wait before playback begins.
- `on2sc swf_options_AddVariable` (FLIX2HANDLE flix, const char *name, const char *value)
Adds a custom SWF variable as a name/value pair.
- `on2sc swf_options_DeleteVariable` (FLIX2HANDLE flix, const `int32_t` index)
Deletes a custom SWF variable.
- `on2sc swf_options_GetVariableCount` (FLIX2HANDLE flix, `int32_t` *pVariableCount)
Gets the number of custom SWF variables already added.
- `on2sc swf_options_UpdateVariable` (FLIX2HANDLE flix, const `int32_t` index, const char *name, const char *value)
Updates an already existing SWF variable.

12.50.1 Detailed Description

Muxer Parameters:

Name	Type	Opt/Reqd	Range
=====			
FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR	Number	Optional	[0.0,2.0] step: 0.1
FE2_SWF_ADD_VARIABLE	String	Optional	N/A
FE2_SWF_DELETE_VARIABLE	String	Optional	N/A
FE2_SWF_EMBEDDED_URL	String	Optional	N/A
FE2_SWF_EMBEDDED_URL_TARGET	String	Optional	N/A
FE2_SWF_EMBEDDED_URL_TYPE	String	Optional	
FE2_SwfEmbeddedUrlType			
FE2_SWF_FIXED_PRELOAD_PCT	Number	Optional	[0,100]
FE2_SWF_FRAMERATE	Number	Optional	N/A
FE2_SWF_HEIGHT	Number	Optional	N/A
FE2_SWF_LOOP_COUNT	Number	Optional	[0,)
FE2_SWF_ON_END_OPTION	String	Optional	
FE2_SwfOnEndOptions			
FE2_SWF_ON_END_URL	String	Optional	N/A
FE2_SWF_ON_START_OPTION	String	Optional	
FE2_SwfOnStartOptions			
FE2_SWF_PRELOAD_TYPE	Number	Optional	
FE2_SwfPreloaderOptions			
FE2_SWF_START_BLANK_FRAME	Number	Optional	[0,1]
FE2_SWF_START_WAIT_SEC	Number	Optional	[0.0,)
FE2_SWF_WIDTH	Number	Optional	N/A

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_SWF);
// enable the adaptive preloader
if(sc == ON2_OK)
    sc = Flix2_MuxerSetParam(muxer, FE2_SWF_PRELOAD_TYPE,
        SwfAdaptivePreloader);
```

12.50.2 Supported Codecs

- Video:
 - [FE2_CODEC_H263](#)
 - [FE2_CODEC_VP6](#)
 - [FE2_CODEC_VP6ALPHA](#)
- Audio:
 - [FE2_CODEC_LAME](#)
 - [FE2_CODEC_PCM](#)

12.50.3 Define Documentation

12.50.3.1 #define FE2_MUXER_SWF

SWF muxer. For use with [Flix2_AddMuxer\(\)](#).

Definition at line 105 of file swf.h.

12.50.3.2 #define FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR

Sets the adaptive preload buffer factor. An adaptive preloader is an "intelligent" preloader that will preload a variable amount of video based on the size of the video and the viewer's internet connection speed so that the viewer can watch the video without stops and starts - no matter what speed they are connected to the internet at. The adaptive preloader determines the user's connection speed and based on that figure and the size of the video file preloading, it preloads precisely the amount of the video required to enable the video to begin playing as soon as it possibly can and still not stop at any point until it is done. In this way, the preload process is optimized so the amount of time taken for a video to preload is minimized as much as possible. If the adaptive preloader is selected, a buffer factor must also be selected. The buffer factor will cause the preloader to preload more (or less) of the video than the initial calculations determine is the minimum amount of time required. This is useful because a viewer's connection speed often varies, even during the course of watching a single video. The time required to load the file up to the end is multiplied by this factor. If a buffer is not needed, set the factor to 1.0 (a factor of 1.0 means no buffer - $\text{loadingTime} * 1.0 = \text{loadingTime}$). To be on the safe side, the recommendation is to use a factor larger than 1.0. If the viewer's connection should get faster during the visit (e.g., perhaps a download ends) then the buffer factor can be changed to a number smaller than 1.0. If the viewer's connection will likely get slower during the visit (e.g., because the viewer starts a new download) the buffer factor should be changed to a number greater than 1.0. As a general rule, it is a good idea to make the buffer factor larger than 1.0 to be on the safe side.

Note:

Default: 1.1

Valid range: [0.0,2.0] step: 0.1

This implicitly sets [FE2_SWF_PRELOAD_TYPE](#) to [SwfAdaptivePreloader](#)

Works with any Flash player version 5 and higher.

Definition at line 222 of file swf.h.

12.50.3.3 #define FE2_SWF_ADD_VARIABLE

Add or update a variable in the SWF. Multiple variables may be specified.

Parameter format:

```
Format = Variable *['&' Variable ]  
Variable = varName '=' varValue  
e.g. "n0=v0&n1=v1"
```

Definition at line 262 of file swf.h.

12.50.3.4 #define FE2_SWF_DELETE_VARIABLE

Delete an existing variable in the SWF. Multiple variables may be specified.

Parameter format:

```
Format = varName *['&' varName ]  
e.g. "var0&var1"
```

Definition at line 270 of file swf.h.

12.50.3.5 #define FE2_SWF_EMBEDDED_URL

Set the SWF's target URL. When a user clicks on the video of the encoded SWF file, the SWF file will immediately try and load the embedded URL.

See also:

[FE2_SWF_EMBEDDED_URL_TARGET](#), [FE2_SWF_EMBEDDED_URL_TYPE](#)

Definition at line 138 of file swf.h.

12.50.3.6 #define FE2_SWF_EMBEDDED_URL_TARGET

Set the target of [FE2_SWF_EMBEDDED_URL](#). Valid values:

- "_self"
- "_blank"
- "_parent"
- "_top"

Note:

Default: "_self"

See also:

http://www.w3.org/TR/REC-WebCGM/REC-03-CGM-IC.html#webcgm_3_1_2_2

Definition at line 151 of file swf.h.

12.50.3.7 `#define FE2_SWF_EMBEDDED_URL_TYPE`

Set how [FE2_SWF_EMBEDDED_URL](#) is interpreted. Valid values are defined by [FE2_EMBEDDED_URL_TYPE](#)

Note:

Default: [EmbeddedUrlIsLoadMovie](#)

Definition at line 158 of file swf.h.

12.50.3.8 `#define FE2_SWF_FIXED_PRELOAD_PCT`

Sets the percent of the SWF movie to preload before playback begins.

Note:

Default: 20%

Valid range: [0,100]

This implicitly sets [FE2_SWF_PRELOAD_TYPE](#) to [SwfFixedPreloader](#)

Works with any Flash player version 4 and higher.

Definition at line 184 of file swf.h.

12.50.3.9 `#define FE2_SWF_FRAMERATE`

Set the SWF framerate. A few things should be considered before setting the SWF framerate. First, SWF framerate should always be equal to or a multiple of the video framerate to prevent problems with encoding such as loss of audio sync and audio distortion. Second, the fractional part of the output SWF framerate has to be rounded to the nearest 1/256th, i.e. 29.97 becomes 29.96875.

Note:

Changing the SWF framerate will NOT change the duration or the audio/video synchronization except as mentioned above.

Default: video framerate

See also:

[Frame Rate](#)

Definition at line 130 of file swf.h.

12.50.3.10 `#define FE2_SWF_HEIGHT`

Set the SWF height.

Note:

Default: video height

Definition at line 110 of file swf.h.

12.50.3.11 #define FE2_SWF_LOOP_COUNT

Sets the number of times the SWF should loop.

Note:

Default: 0

Setting this to a value >0 will see the SWF ignore the loop command of the Flash player.

This will implicitly set [FE2_SWF_ON_END_OPTION](#) to [SwfOnMovieEndLoop](#)

Definition at line 168 of file swf.h.

12.50.3.12 #define FE2_SWF_ON_END_OPTION

Sets the options for the end of the SWF. Valid values are defined by [FE2_SwfOnEndOptions](#)

Note:

Default: [SwfOnMovieEndNothing](#)

Definition at line 228 of file swf.h.

12.50.3.13 #define FE2_SWF_ON_END_URL

Sets the URL a SWF movie will load after the current movie ends.

Note:

This will implicitly set [FE2_SWF_ON_END_OPTION](#) to [SwfOnMovieEndLoadMovie](#)

Definition at line 234 of file swf.h.

12.50.3.14 #define FE2_SWF_ON_START_OPTION

Sets the options for the start of the SWF. Valid values are defined by [FE2_SwfOnStartOptions](#).

Note:

Default: [SwfOnMovieStartAutomatically](#)

Definition at line 241 of file swf.h.

12.50.3.15 #define FE2_SWF_PRELOAD_TYPE

Sets the type of preloader. Valid types are defined by [FE2_SwfPreloaderOptions](#).

Note:

Default: [SwfPreloaderNone](#)

Definition at line 175 of file swf.h.

12.50.3.16 #define FE2_SWF_START_BLANK_FRAME

Control the insertion of a blank first frame in the SWF.

Note:

Default: 0

Definition at line 247 of file swf.h.

12.50.3.17 #define FE2_SWF_START_WAIT_SEC

Sets the number of seconds to wait before playback begins.

Note:

Default: 0

Definition at line 252 of file swf.h.

12.50.3.18 #define FE2_SWF_WIDTH

Set the SWF width.

Note:

Default: video width

Definition at line 115 of file swf.h.

12.50.4 Enumeration Type Documentation

12.50.4.1 enum FE2_EmbeddedUrlType

Differentiates between the type of file (HTML or SWF) set through [FE2_SWF_EMBEDDED_URL](#).

Enumerator:

EmbeddedUrlsGetUrl

EmbeddedUrlsLoadMovie

Definition at line 68 of file swf.h.

12.50.4.2 enum FE2_SwfOnEndOptions

Actions that can be added to the last frame of a SWF file. For use with [FE2_SWF_ON_END_OPTION](#)

Enumerator:

SwfOnMovieEndNothing

SwfOnMovieEndSTOP

SwfOnMovieEndLoop

SwfOnMovieEndUnload
SwfOnMovieEndLoadMovie

Definition at line 76 of file swf.h.

12.50.4.3 enum FE2_SwfOnStartOptions

Actions that can be added to the start frame of the SWF file. For use with [FE2_SWF_ON_START_OPTION](#)

Enumerator:

SwfOnMovieStartAutomatically
SwfOnMovieStartOnClick
SwfOnMovieStartWait
SwfOnMovieStartEmbedSTOP

Definition at line 87 of file swf.h.

12.50.4.4 enum FE2_SwfPreloaderOptions

Determines the type of preloader added to the SWF file. For use with [FE2_SWF_PRELOAD_TYPE](#)

Enumerator:

SwfPreloaderNone
SwfFixedPreloader
SwfAdaptivePreloader

Definition at line 97 of file swf.h.

12.50.5 Function Documentation

12.50.5.1 on2sc swf_options_AddVariable (FLIX2HANDLE *flix*, const char * *name*, const char * *value*)

Adds a custom SWF variable as a name/value pair.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *name* The name of the variable.
- ← *value* The value of the variable.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NO_MEM](#) an error occurred allocating memory for *name* or *value*

Precondition:

flix is not NULL
name is not NULL
value is not NULL

See also:

[swf_options_DeleteVariable\(\)](#), [swf_options_GetVariableCount\(\)](#), [swf_options_UpdateVariable\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADD_VARIABLE](#) parameter. This function will be removed in a future release.

12.50.5.2 on2sc swf_options_DeleteVariable (FLIX2HANDLE *flix*, const int32_t *index*)

Deletes a custom SWF variable.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 ← *index* The zero based index of the variable to delete.

Return values:

[ON2_OK](#) Success
[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
[ON2_NO_MEM](#) an error occurred reallocating memory for the variable list.

Precondition:

flix is not NULL
index is inside the range of current variables.

See also:

[swf_options_AddVariable\(\)](#), [swf_options_GetVariableCount\(\)](#), [swf_options_UpdateVariable\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_DELETE_VARIABLE](#) parameter. This function will be removed in a future release.

12.50.5.3 on2sc swf_options_GetAdaptivePreloaderBufferFactor (FLIX2HANDLE *flix*, double * *pPreloaderBufferFactor*)

Gets the adaptive preload buffer factor.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *pPreloaderBufferFactor* The adaptive preloader buffer factor.

Return values:[*ON2_OK*](#) Success[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail**Precondition:***flix* is not NULL*pPreloaderBufferFactor* is not NULL**See also:**[swf_options_SetAdaptivePreloaderBufferFactor\(\)](#)**Deprecated**

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter. This function will be removed in a future release.

12.50.5.4 on2sc swf_options_GetEmbeddedUrl (FLIX2HANDLE *flix*, char * *embeddedUrl*, int32_t * *len*)

Gets the embedded URL.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *embeddedUrl* The embedded URL.

↔ *len* • IN: The length of *embeddedUrl*.
 • OUT: The length of the URL (minus the null terminator).

Return values:[*ON2_OK*](#) Success[*ON2_INVALID_PARAMS*](#) should one or more of the preconditions fail[*ON2_NO_MEM*](#) *embeddedUrl* does not have enough allocated space to return the URL. The size in bytes needed (minus null terminator) will be returned in *len*.**Precondition:***flix* is not NULL*len* is not NULL**Note:**

If *embeddedUrl* is NULL this function will return the size in bytes required to store the current embedded URL in *len*, not including the null terminator.

See also:[swf_options_SetEmbeddedUrl\(\)](#),
[SetEmbeddedUrlType\(\)](#)[swf_options_GetEmbeddedUrlType\(\)](#)[swf_options_](#)**Attention:**

Currently only supported in C/C++

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL](#) parameter. This function will be removed in a future release.

12.50.5.5 on2sc swf_options_GetEmbeddedUrlTarget (FLIX2HANDLE *flx*, char * *embeddedUrlTarget*, int32_t * *len*)

Gets the target of embedded URL.

Parameters:

- ← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *embeddedUrlTarget* The target of embedded URL.
- ↔ *len*
 - IN: The length of *embeddedUrlTarget*.
 - OUT: The length of the URL (minus the null terminator).

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NO_MEM](#) *embeddedUrlTarget* does not have enough allocated space to return the target. The size in bytes needed (minus null terminator) will be returned in *len*.

Precondition:

flx is not NULL
len is not NULL

Note:

If *embeddedUrlTarget* is NULL this function will return the size in bytes required to store the current target in *len*, not including the null terminator.

See also:

[swf_options_GetEmbeddedUrl\(\)](#), [swf_options_SetEmbeddedUrl\(\)](#), [swf_options_SetEmbeddedUrlType\(\)](#)

Attention:

Currently only supported in C/C++

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TARGET](#) parameter. This function will be removed in a future release.

12.50.5.6 on2sc swf_options_GetEmbeddedUrlType (FLIX2HANDLE *flx*, FE2_EmbeddedUrlType * *embeddedUrlType*)

Gets the type of embedded URL.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *embeddedUrlType* The type of embedded URL.

Return values:

- [ON2_OK](#) success.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- embeddedUrlType* is not NULL

See also:

[swf_options_SetEmbeddedUrl\(\)](#), [swf_options_SetEmbeddedUrlType\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TYPE](#) parameter. This function will be removed in a future release.

12.50.5.7 on2sc swf_options_GetEnablePreloader (FLIX2HANDLE *flix*, on2bool * *pEnablePreloader*)

Gets if a video preloader is enabled or disabled.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pEnablePreloader*

Return values:

- [ON2_OK](#) Success.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pEnablePreloader* is not NULL

See also:

[swf_options_SetEnablePreloader\(\)](#)

12.50.5.8 on2sc swf_options_GetInsertBlankFrameOnStart (FLIX2HANDLE *flix*, on2bool * *pInsertBlankFrameOnStart*)

Gets if the engine is to insert a blank frame as the first frame of the SWF or not.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *pInsertBlankFrameOnStart*

Return values:

[ON2_OK](#) Successfully returned if the engine will insert a blank frame or not.
[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL
pInsertBlankFrameOnStart is not NULL

See also:

[swf_options_SetInsertBlankFrameOnStart\(\)](#)

12.50.5.9 on2sc swf_options_GetLoadMovieOnEndUrl (FLIX2HANDLE *flix*, char * *pLoadMovieOnEndUrl*, int32_t * *len*)

Gets the URL of a SWF movie to load after the current movie ends.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *pLoadMovieOnEndUrl* The URL of a SWF movie.
 ↔ *len*

- IN: The length of *pLoadMovieOnEndUrl*.
- OUT: The length of the URL (minus the null terminator).

Return values:

[ON2_OK](#) Success
[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
[ON2_NO_MEM](#) *pLoadMovieOnEndUrl* does not have enough allocated space to return the URL.
 The size in bytes needed (minus null terminator) will be returned in *len*.

Precondition:

flix is not NULL
len is not NULL

Note:

If *pLoadMovieOnEndUrl* is NULL this function will return the size in bytes required to store the current URL in *len*, not including the null terminator.

See also:

[swf_options_SetLoadMovieOnEndUrl\(\)](#)

Attention:

Currently only supported in C/C++

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_URL](#) parameter. This function will be removed in a future release.

12.50.5.10 on2sc swf_options_GetLoopCount (FLIX2HANDLE *flix*, int32_t * *pLoopCount*)

Gets the number of times the SWF should loop.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *pLoopCount* The number of times the SWF should loop.

Return values:

- [ON2_OK](#) Success
[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
pLoopCount is not NULL

See also:

[swf_options_SetLoopCount\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_LOOP_COUNT](#) parameter. This function will be removed in a future release.

12.50.5.11 on2sc swf_options_GetMovieOnEndOptions (FLIX2HANDLE *flix*, FE2_SwfOnEndOptions * *pOnEndOptions*)

Gets the options for the end of the SWF.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
→ *pOnEndOptions* The options for the end of the SWF.

Return values:

- [ON2_OK](#) Success
[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
pOnEndOptions is not NULL

See also:

[swf_options_SetMovieOnEndOptions\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_OPTION](#) parameter. This function will be removed in a future release.

12.50.5.12 on2sc swf_options_GetMovieOnStartOptions (FLIX2HANDLE *flix*, FE2_SwfOnStartOptions * *pOnStartOptions*)

Gets the options for the start of the SWF.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pOnStartOptions* The options for the start of the SWF.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pOnStartOptions* is not NULL

See also:

[swf_options_SetMovieOnStartOptions\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_START_OPTION](#) parameter. This function will be removed in a future release.

12.50.5.13 on2sc swf_options_GetPercentToPreload (FLIX2HANDLE *flix*, int32_t * *pPercentToPreload*)

Gets the percent of the SWF movie to preload before playback begins.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pPercentToPreload* Percent of the SWF movie to preload before playback begins.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pPercentToPreload* is not NULL

See also:

[swf_options_SetPercentToPreload\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FIXED_PRELOAD_PCT](#) parameter. This function will be removed in a future release.

12.50.5.14 on2sc swf_options_GetPreloaderType (FLIX2HANDLE *flix*, FE2_SwfPreloaderOptions * *pPreloaderType*)

Gets the type of preloader.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pPreloaderType* The preloader type.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pPreloaderType* is not NULL

See also:

[swf_options_SetPreloaderType\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_PRELOAD_TYPE](#) parameter. This function will be removed in a future release.

12.50.5.15 on2sc swf_options_GetSwfFramerate (FLIX2HANDLE *flix*, int32_t * *pSwfFramerate*)

Gets the SWF framerate.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *pSwfFramerate* The SWF framerate.

Return values:

- [ON2_OK](#) The engine successfully returned SWF framerate.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pSwfFramerate* is not NULL

Deprecated

This function has been deprecated in favor of [swf_options_GetSwfFramerateAsDouble\(\)](#) because this function can only handle integer framerates.

See also:

[swf_options_GetSwfFramerateAsDouble\(\)](#)

12.50.5.16 on2sc swf_options_GetSwfFramerateAsDouble (FLIX2HANDLE *flix*, double * *pSwfFramerate*)

Gets the SWF framerate as a double.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pSwfFramerate* The SWF framerate.

Return values:

- [ON2_OK](#) The engine successfully returned SWF framerate.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pSwfFramerate* is not NULL

See also:

[swf_options_SetSwfFramerateAsDouble\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FRAMERATE](#) parameter. This function will be removed in a future release.

12.50.5.17 on2sc swf_options_GetVariableCount (FLIX2HANDLE *flix*, int32_t * *pVariableCount*)

Gets the number of custom SWF variables already added.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pVariableCount* The number of custom SWF variables already added

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pVariableCount* is not NULL

See also:

[swf_options_AddVariable\(\)](#), [swf_options_DeleteVariable\(\)](#), [swf_options_UpdateVariable\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#). This function will be removed in a future release.

12.50.5.18 on2sc swf_options_GetWaitTimeToStart (FLIX2HANDLE *flix*, int32_t * *pWaitTimeToStart*)

Gets the number of seconds to wait before playback begins.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pWaitTimeToStart* Number of seconds to wait.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- pWaitTimeToStart* is not NULL

See also:

[swf_options_SetWaitTimeToStart\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_WAIT_SEC](#) parameter. This function will be removed in a future release.

12.50.5.19 on2sc swf_options_Reset (FLIX2HANDLE *flix*)

Resets the swf options.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

- [ON2_OK](#) reset of swf options was successful
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL

Note:

Call this function if you wish to reset all swf options to their default values. This will free any memory that was allocated to the swf object.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#). This function will be removed in a future release.

12.50.5.20 `on2sc swf_options_SetAdaptivePreloaderBufferFactor (FLIX2HANDLE flix, const double preloaderBufferFactor)`

Sets the adaptive preload buffer factor.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *preloaderBufferFactor* The adaptive preloader buffer factor.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

Note:

- The preloader has to be enabled.
- The default value is 1.1.
- The range is 0-2 in increments of 0.1.
- The preloader type must be [SwfAdaptivePreloader](#).
- Works with any Flash player version 5 and higher.

See also:

[swf_options_GetAdaptivePreloaderBufferFactor\(\)](#)

Remarks:

An adaptive preloader is an "intelligent" preloader that will preload a variable amount of video based on the size of the video and the viewer's internet connection speed so that the viewer can watch the video without stops and starts - no matter what speed they are connected to the internet at. The adaptive preloader determines the user's connection speed and based on that figure and the size of the video file preloading, it preloads precisely the amount of the video required to enable the video to begin playing as soon as it possibly can and still not stop at any point until it is done. In this way, the preload process is optimized so the amount of time taken for a video to preload is minimized as much as possible. If the adaptive preloader is selected, a buffer factor must also be selected. The buffer factor will cause the preloader to preload more (or less) of the video than the initial calculations determine is the minimum amount of time required. This is useful because a viewer's connection speed often varies, even during the course of watching a single video. The time required to load the file up to the end is multiplied by this factor. If a buffer is not needed, set the factor to 1.0 (a factor of 1.0 means no buffer - $\text{loadingTime} * 1.0 = \text{loadingTime}$). To be on the safe side, the recommendation is to use a factor larger than 1.0. If the viewer's connection should get faster during the visit (e.g., perhaps a download ends) then the buffer factor can be changed to a number smaller than 1.0. If the viewer's connection will likely get slower during the visit (e.g., because the viewer starts a new download) the buffer factor should be changed to a number greater than 1.0. As a general rule, it is a good idea to make the buffer factor larger than 1.0 to be on the safe side.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter. This function will be removed in a future release.

12.50.5.21 on2sc swf_options_SetEmbeddedUrl (FLIX2HANDLE *flix*, const char * *embeddedUrl*)

Sets the embedded URL of the SWF or HTML that will be loaded when the user clicks on the video.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *embeddedUrl* The embedded URL.

Return values:

- [ON2_OK](#) success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NO_MEM](#) an error occurred allocating memory for *embeddedUrl*

Precondition:

flix is not NULL

Note:

When a user clicks on the video of the encoded SWF file, the SWF file will immediately try and load the embedded URL.
If *embeddedUrl* is NULL then the engine will delete the embedded URL.

See also:

[swf_options_GetEmbeddedUrl\(\)](#), [swf_options_GetEmbeddedUrlType\(\)](#), [swf_options_SetEmbeddedUrlType\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL](#) parameter. This function will be removed in a future release.

12.50.5.22 on2sc swf_options_SetEmbeddedUrlTarget (FLIX2HANDLE *flix*, const char * *embeddedUrlTarget*)

Sets the target of the embedded URL.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *embeddedUrlTarget* The target of the embedded URL.

Return values:

- [ON2_OK](#) success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NO_MEM](#) an error occurred allocating memory for *embeddedUrlTarget*

Precondition:

flix is not NULL
If *embeddedUrlTarget* is not NULL then it must be one of the valid values.

Note:

The *embeddedUrlTarget* will be ignored if the embedded URL is not set.
Valid values for *embeddedUrlTarget* are:

- `"_self"`
- `"_blank"`
- `"_parent"`
- `"_top"`
- see http://www.w3.org/TR/REC-WebCGM/REC-03-CGM-IC.html#webcgm_3_1_2_2 for more information.

The default value is `"_self"`.

If *embeddedUrlTarget* is `NULL` then the engine will delete the target of the embedded URL.

See also:

[swf_options_GetEmbeddedUrl\(\)](#), [swf_options_SetEmbeddedUrl\(\)](#), [swf_options_GetEmbeddedUrlType\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TARGET](#) parameter. This function will be removed in a future release.

12.50.5.23 on2sc swf_options_SetEmbeddedUrlType (FLIX2HANDLE *flix*, const FE2_EmbeddedUrlType *embeddedUrlType*)

Sets the type of embedded URL.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *embeddedUrlType* The type of embedded URL.

Return values:

[ON2_OK](#) success.

[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not `NULL`

embeddedUrlType is a valid member of [FE2_EmbeddedUrlType](#)

Note:

The *embeddedUrlType* will be ignored if the embedded URL is not set.

Remarks:

The default value is [EmbeddedUrlIsLoadMovie](#).

See also:

[swf_options_SetEmbeddedUrl\(\)](#), [swf_options_GetEmbeddedUrlType\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TYPE](#) parameter. This function will be removed in a future release.

12.50.5.24 on2sc swf_options_SetEnablePreloader (FLIX2HANDLE *flix*, const on2bool *enablePreloader*)

Enables or disables the video preloader.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *enablePreloader*

Return values:

[ON2_OK](#) Success

[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

Note:

[on2true](#) will enable a preloader.
[on2false](#) will disable a preloader.

See also:

[swf_options_GetEnablePreloader\(\)](#)

12.50.5.25 on2sc swf_options_SetInsertBlankFrameOnStart (FLIX2HANDLE *flix*, const on2bool *insertBlankFrameOnStart*)

Tells the engine to insert a blank frame as the first frame of the SWF or not.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *insertBlankFrameOnStart* [on2true](#) will insert a blank frame. [on2false](#) will not.

Return values:

[ON2_OK](#) The engine set the variable of whether or not to insert a blank frame.

[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

Note:

[on2true](#) will insert a blank frame.
[on2false](#) will not.

See also:

[swf_options_GetInsertBlankFrameOnStart\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_BLANK_FRAME](#) parameter. This function will be removed in a future release.

12.50.5.26 on2sc swf_options_SetLoadMovieOnEndUrl (FLIX2HANDLE *flix*, const char * *loadMovieOnEndUrl*)

Sets the URL of a SWF movie to load after the current movie ends.

Parameters:

← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *loadMovieOnEndUrl* The URL of a SWF movie.

Return values:

[ON2_OK](#) success

[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

[ON2_NO_MEM](#) an error occurred allocating memory for *loadMovieOnEndUrl*

Precondition:

flix is not NULL

Note:

If *loadMovieOnEndUrl* is NULL then the engine will delete the URL.

See also:

[swf_options_GetLoadMovieOnEndUrl\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_URL](#) parameter. This function will be removed in a future release.

12.50.5.27 on2sc swf_options_SetLoopCount (FLIX2HANDLE *flix*, const int32_t *loopCount*)

Sets the number of times the SWF should loop.

Parameters:

← *flix* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *loopCount* The number of times the SWF should loop.

Return values:

[ON2_OK](#) Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not NULL

Note:

The default value is 0.

Setting this to a value >0 will have the SWF ignore the loop command of the Flash player.

See also:

[swf_options_GetLoopCount\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_LOOP_COUNT](#) parameter. This function will be removed in a future release.

12.50.5.28 on2sc swf_options_SetMovieOnEndOptions (FLIX2HANDLE *flix*, const FE2_SwfOnEndOptions *onEndOptions*)

Sets the options for the end of the SWF.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *onEndOptions* The options for the end of the SWF.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not NULL

onEndOptions is a valid member of [FE2_SwfOnEndOptions](#)

Note:

The default value is [SwfOnMovieEndNothing](#).

See also:

[swf_options_GetMovieOnEndOptions\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_OPTION](#) parameter. This function will be removed in a future release.

12.50.5.29 `on2sc swf_options_SetMovieOnStartOptions (FLIX2HANDLE flix, const FE2_SwfOnStartOptions onStartOptions)`

Sets the options for the start of the SWF.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *onStartOptions* The options for the start of the SWF.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- onStartOptions* is a valid member of [FE2_SwfOnStartOptions](#)

Note:

The default value is [SwfOnMovieStartAutomatically](#).

See also:

[swf_options_GetMovieOnStartOptions\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_START_OPTION](#) parameter. This function will be removed in a future release.

12.50.5.30 `on2sc swf_options_SetPercentToPreload (FLIX2HANDLE flix, const int32_t percentToPreload)`

Sets the percent of the SWF movie to preload before playback begins.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *percentToPreload* Percent of the SWF movie to preload before playback begins.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL

Note:

The preloader has to be enabled.

The default value is 20%.
 The range is 1-100.
 The preloader type must be [SwfFixedPreloader](#).
 Works with any Flash player version 4 and higher.

See also:

[swf_options_GetPercentToPreload\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FIXED_PRELOAD_PCT](#) parameter. This function will be removed in a future release.

12.50.5.31 on2sc swf_options_SetPreloaderType (FLIX2HANDLE *flix*, const FE2_SwfPreloaderOptions *preloaderType*)

Sets the type of preloader.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *preloaderType* The preloader type.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flix* is not NULL
- preloaderType* is a valid member of [FE2_SwfPreloaderOptions](#)

Note:

- The preloader has to be enabled.
- The default value is [SwfFixedPreloader](#).

See also:

[swf_options_GetPreloaderType\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_PRELOAD_TYPE](#) parameter. This function will be removed in a future release.

12.50.5.32 on2sc swf_options_SetSwfFramerate (FLIX2HANDLE *flix*, const int32_t *swfFramerate*)

Sets the SWF framerate.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *swfFramerate* The SWF framerate.

Return values:

- [ON2_OK](#) The engine successfully set the SWF framerate.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

Deprecated

This function has been deprecated in favor of [swf_options_SetSwfFramerateAsDouble\(\)](#) because this function can only handle integer framerates.

See also:

[swf_options_SetSwfFramerateAsDouble\(\)](#)

12.50.5.33 on2sc swf_options_SetSwfFramerateAsDouble (FLIX2HANDLE *flix*, const double *swfFramerate*)

Sets the SWF framerate as a double.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *swfFramerate* The SWF framerate.

Return values:

- [ON2_OK](#) The engine successfully set SWF framerate.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

Note:

The default value will be the video framerate.
 The SWF framerate may be changed when the encoding process starts because of two conditions. Changing the SWF framerate will NOT change the duration or the audio/video synchronization. The first condition is the SWF framerate should always be equal to or a multiple of the video framerate to prevent problems with encoding such as loss of audio sync and audio distortion. The second condition is the fractional part of the output SWF framerate has to be rounded to the nearest 1/256th. I.E. If the SWF framerate is set to 29.97 then the output SWF framerate will be converted to 29.96875.

See also:

[swf_options_GetSwfFramerateAsDouble\(\)](#), [video_options_SetVideoFramerateAsDouble\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FRAMERATE](#) parameter. This function will be removed in a future release.

12.50.5.34 on2sc swf_options_SetWaitTimeToStart (FLIX2HANDLE *flix*, const int32_t *waitTimeToStart*)

Sets the number of seconds to wait before playback begins.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *waitTimeToStart* Number of seconds to wait.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

Note:

The default value is 0.

See also:

[swf_options_GetWaitTimeToStart\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_WAIT_SEC](#) parameter. This function will be removed in a future release.

12.50.5.35 on2sc swf_options_UpdateVariable (FLIX2HANDLE *flix*, const int32_t *index*, const char * *name*, const char * *value*)

Updates an already existing SWF variable.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *index* The zero based index of the variable to update.
- ← *name* The name of the variable.
- ← *value* The value of the variable.

Return values:

- [ON2_OK](#) Success
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail
- [ON2_NO_MEM](#) an error occurred reallocating memory for the variable list.

Precondition:

flix is not NULL
name is not NULL
value is not NULL
index is inside the range of current variables

See also:

[swf_options_AddVariable\(\)](#), [swf_options_DeleteVariable\(\)](#), [swf_options_GetVariableCount\(\)](#)

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADD_VARIABLE](#) parameter. This function will be removed in a future release.

12.51 WebM - FFmpeg

Defines

- `#define` [FE2_MUXER_WEBM](#)
WebM muxer. For use with [Flix2_AddMuxer\(\)](#).

12.51.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_WEBM);  
if(sc != ON2_OK)  
    ; //Now using the WebM muxer for output
```

12.51.2 Supported Codecs

- Video:
 - [FE2_CODEC_VP8](#)
- Audio:
 - [FE2_CODEC_VORBIS](#)

Additional References:

- [WebM Project](#)

12.51.3 Define Documentation

12.51.3.1 `#define` FE2_MUXER_WEBM

WebM muxer. For use with [Flix2_AddMuxer\(\)](#).

Definition at line 45 of file webm.h.

12.52 Base Types

Defines

- `#define OTC(str)`
a macro suitable for declaring a constant `on2tc`
- `#define ON2TC`
printf format string suitable for printing an `on2tc`
- `#define ON264`
printf format string suitable for printing an `on2s64`
- `#define INT64_MIN`
- `#define INT64_MAX`
- `#define UINT64_MAX`
- `#define PRId64`
- `#define PRIu64`

Typedefs

- `typedef char int8_t`
- `typedef short int16_t`
- `typedef int int32_t`
- `typedef unsigned char uint8_t`
- `typedef unsigned short uint16_t`
- `typedef unsigned int uint32_t`
- `typedef int8_t on2s8`
- `typedef uint8_t on2u8`
- `typedef int16_t on2s16`
- `typedef uint16_t on2u16`
- `typedef int32_t on2s32`
- `typedef uint32_t on2u32`
- `typedef int32_t on2bool`
- `typedef char on2tc`
- `typedef __int64 on2s64`
- `typedef unsigned __int64 on2u64`
- `typedef on2s64 int64_t`
- `typedef on2u64 uint64_t`

Enumerations

- `enum _on2bool {`
 `on2false,`
 `on2true }`

- enum [on2sc](#) {
 [ON2_NOT_FOUND](#),
 [ON2_BUFFER_EMPTY](#),
 [ON2_BUFFER_FULL](#),
 [ON2_CONNREFUSED](#),
 [ON2_TIMEDOUT](#),
 [ON2_WOULDBLOCK](#),
 [ON2_NET_ERROR](#),
 [ON2_INVALID_VERSION](#),
 [ON2_INPROGRESS](#),
 [ON2_NOT_SUPP](#),
 [ON2_NO_MEM](#),
 [ON2_INVALID_PARAMS](#),
 [ON2_ERROR](#),
 [ON2_OK](#),
 [ON2_DONE](#) }

Common return type.

12.52.1 Define Documentation

12.52.1.1 #define INT64_MAX

Definition at line 104 of file [on2types.h](#).

12.52.1.2 #define INT64_MIN

Definition at line 102 of file [on2types.h](#).

12.52.1.3 #define ON264

printf format string suitable for printing an [on2s64](#)

Definition at line 110 of file [on2types.h](#).

12.52.1.4 #define ON2TC

printf format string suitable for printing an [on2tc](#)

Definition at line 81 of file [on2types.h](#).

12.52.1.5 #define OTC(str)

a macro suitable for declaring a constant [on2tc](#)

Definition at line 80 of file [on2types.h](#).

12.52.1.6 #define PRId64

Definition at line 108 of file on2types.h.

12.52.1.7 #define PRIu64

Definition at line 109 of file on2types.h.

12.52.1.8 #define UINT64_MAX

Definition at line 106 of file on2types.h.

12.52.2 Typedef Documentation**12.52.2.1 typedef short int16_t**

Definition at line 43 of file on2types.h.

12.52.2.2 typedef int int32_t

Definition at line 44 of file on2types.h.

12.52.2.3 typedef on2s64 int64_t

Definition at line 130 of file on2types.h.

12.52.2.4 typedef char int8_t

Definition at line 42 of file on2types.h.

12.52.2.5 typedef int32_t on2bool

Definition at line 57 of file on2types.h.

12.52.2.6 typedef int16_t on2s16

Definition at line 53 of file on2types.h.

12.52.2.7 typedef int32_t on2s32

Definition at line 55 of file on2types.h.

12.52.2.8 typedef __int64 on2s64

Definition at line 111 of file on2types.h.

12.52.2.9 typedef int8_t on2s8

Definition at line 51 of file on2types.h.

12.52.2.10 typedef char on2tc

Definition at line 79 of file on2types.h.

12.52.2.11 typedef uint16_t on2u16

Definition at line 54 of file on2types.h.

12.52.2.12 typedef uint32_t on2u32

Definition at line 56 of file on2types.h.

12.52.2.13 typedef unsigned __int64 on2u64

Definition at line 112 of file on2types.h.

12.52.2.14 typedef uint8_t on2u8

Definition at line 52 of file on2types.h.

12.52.2.15 typedef unsigned short uint16_t

Definition at line 47 of file on2types.h.

12.52.2.16 typedef unsigned int uint32_t

Definition at line 48 of file on2types.h.

12.52.2.17 typedef on2u64 uint64_t

Definition at line 131 of file on2types.h.

12.52.2.18 typedef unsigned char uint8_t

Definition at line 46 of file on2types.h.

12.52.3 Enumeration Type Documentation**12.52.3.1 enum _on2bool**

Enumerator:

on2false

on2true

Definition at line 59 of file on2types.h.

12.52.3.2 enum on2sc

Common return type.

Enumerator:

ON2_NOT_FOUND
ON2_BUFFER_EMPTY
ON2_BUFFER_FULL
ON2_CONNREFUSED
ON2_TIMEDOUT
ON2_WOULDBLOCK
ON2_NET_ERROR
ON2_INVALID_VERSION
ON2_INPROGRESS
ON2_NOT_SUPP
ON2_NO_MEM
ON2_INVALID_PARAMS
ON2_ERROR
ON2_OK
ON2_DONE

Definition at line 137 of file on2types.h.

12.53 Video Encoding Options

Modules

- [Deprecated](#)

Functions

- [on2sc video_options_Reset](#) ([FLIX2HANDLE](#) flx)
Reset the video options to their defaults.
- [on2sc video_options_Validate](#) (const [FLIX2HANDLE](#) flx)
Ensure the current video settings are valid.
- [on2sc video_options_GetSwfHeight](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpSwfHeight)
Gets the SWF height.
- [on2sc video_options_SetSwfHeight](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lSwfHeight)
Sets the SWF height.
- [on2sc video_options_GetSwfWidth](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpSwfWidth)
Gets the SWF width.
- [on2sc video_options_SetSwfWidth](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lSwfWidth)
Sets the SWF width.
- [on2sc video_options_GetUseCustomSwfDimensions](#) (const [FLIX2HANDLE](#) flx, [on2bool](#) *bpUseCustomSwfDimensions)
Determine if the engine is using the SWF width and height for the SWF.
- [on2sc video_options_SetUseCustomSwfDimensions](#) ([FLIX2HANDLE](#) flx, const [on2bool](#) bUseCustomSwfDimensions)
Tells the engine to use the SWF width and SWF height for the SWF.
- [on2sc video_options_GetSourceHeight](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *height)
Get the height of the source video.
- [on2sc video_options_GetSourceWidth](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *width)
Get the width of the source video.

12.53.1 Function Documentation

12.53.1.1 on2sc video_options_GetSourceHeight (const [FLIX2HANDLE](#) flx, [int32_t](#) * height)

Get the height of the source video.

Parameters:

← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *height* Source video height, in pixels

Return values:

[*ON2_OK*](#) The source video size was successfully retrieved from the engine.

12.53.1.2 on2sc video_options_GetSourceWidth (const FLIX2HANDLE *flix*, int32_t * *width*)

Get the width of the source video.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *width* Source video width, in pixels

Return values:

[*ON2_OK*](#) The source video size was successfully retrieved from the engine.

12.53.1.3 on2sc video_options_GetSwfHeight (const FLIX2HANDLE *flix*, int32_t * *lpSwfHeight*)

Gets the SWF height.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *lpSwfHeight* SWF height.

Return values:

[*ON2_OK*](#) Success.

Note:

If a value other than [ON2_OK](#) is returned, the output variables should not be considered valid.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_HEIGHT](#) parameter. This function will be removed in a future release.

12.53.1.4 on2sc video_options_GetSwfWidth (const FLIX2HANDLE *flix*, int32_t * *lpSwfWidth*)

Gets the SWF width.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *lpSwfWidth* SWF width.

Return values:

[*ON2_OK*](#) Success.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH](#) parameter. This function will be removed in a future release.

12.53.1.5 on2sc video_options_GetUseCustomSwfDimensions (const FLIX2HANDLE *flix*, on2bool * *bpUseCustomSwfDimensions*)

Determine if the engine is using the SWF width and height for the SWF.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *bpUseCustomSwfDimensions* Variable to update with the current custom SWF dimensions enable status

Return values:

[ON2_OK](#) Success.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH/FE2_SWF_HEIGHT](#) parameters. This function will be removed in a future release.

12.53.1.6 on2sc video_options_Reset (FLIX2HANDLE *flix*)

Reset the video options to their defaults.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

[ON2_OK](#) The video options were reset successfully.

12.53.1.7 on2sc video_options_SetSwfHeight (FLIX2HANDLE *flix*, const int32_t *lSwfHeight*)

Sets the SWF height.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *lSwfHeight* SWF height.

Return values:

[*ON2_OK*](#) Success.

Note:

By default, the SWF height will be the video height.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_HEIGHT](#) parameter. This function will be removed in a future release.

12.53.1.8 on2sc video_options_SetSwfWidth (FLIX2HANDLE *flix*, const int32_t *lSwfWidth*)

Sets the SWF width.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *lSwfWidth* SWF width.

Return values:

[*ON2_OK*](#) Success.

Note:

By default, the SWF width will be the video width.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH](#) parameter. This function will be removed in a future release.

12.53.1.9 on2sc video_options_SetUseCustomSwfDimensions (FLIX2HANDLE *flix*, const on2bool *bUseCustomSwfDimensions*)

Tells the engine to use the SWF width and SWF height for the SWF.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseCustomSwfDimensions* [on2true](#) to enable, [on2false](#) to disable.

Return values:

[*ON2_OK*](#) Success.

Note:

By default, this is disabled.

This will automatically be set to enabled if [video_options_SetSwfWidth\(\)](#) or [video_options_SetSwfHeight\(\)](#) is called.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_SWF](#) and the [FE2_SWF_WIDTH](#)/[FE2_SWF_HEIGHT](#) parameters. This function will be removed in a future release.

12.53.1.10 on2sc video_options_Validate (const FLIX2HANDLE *flix*)

Ensure the current video settings are valid.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

Return values:

[ON2_OK](#) The video options are valid and encoding may proceed.

[ON2_INVALID_PARAMS](#) One or more options are invalid. Proceeding with encoding may cause encoding to fail or a file that does not match the specified options will be created.

12.54 Deprecated

Enumerations

- enum [FE2_VideoCodec](#) {
[CODEC_NULL](#),
[CODEC_H263](#),
[CODEC_SCREENVIDEO](#),
[CODEC_VP6](#),
[CODEC_VP6ALPHA](#) }

Output video codec types, influences quality/compatibility.

- enum [FE2_CuePointType](#) {
[CUE_EVENT](#),
[CUE_NAVIGATION](#) }

Cue point type for use with [video_options_AddFLVCuePoint\(\)](#).

Functions

- [on2sc video_options_GetImageQuality](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpImageQuality)
Get the current image quality factor.
- [on2sc video_options_SetImageQuality](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lImageQuality)
Set the image quality factor.
- [on2sc video_options_GetKeyframeInterval](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpKeyframeInterval)
Get the current interval between keyframes.
- [on2sc video_options_SetKeyframeInterval](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lKeyframeInterval)
Set the interval between keyframes.
- [on2sc video_options_GetKeyframeIntervalType](#) (const [FLIX2HANDLE](#) flx, [FE2_VideoKeyframeTypes](#) *pKeyframeIntervalType)
Get the current keyframe interval type.
- [on2sc video_options_SetKeyframeIntervalType](#) ([FLIX2HANDLE](#) flx, const [FE2_VideoKeyframeTypes](#) keyframeIntervalType)
Set the keyframe interval type.
- [on2sc video_options_GetMaximumBitrate](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpMaximumBitrate)
Get the current maximum bitrate target.
- [on2sc video_options_SetMaximumBitrate](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lMaximumBitrate)
Set the maximum bitrate target.

- `on2sc video_options_GetRateControlType` (const `FLIX2HANDLE` flix, `FE2_VideoBitrateControls` *pRateControlType)
Get the current rate control type.
- `on2sc video_options_SetRateControlType` (`FLIX2HANDLE` flix, const `FE2_VideoBitrateControls` rateControlType)
Set the rate control type.
- `on2sc video_options_GetUseMaximumBitrate` (const `FLIX2HANDLE` flix, `on2bool` *bpUseMaximumBitrate)
Determine if the maximum bitrate target will be used.
- `on2sc video_options_SetUseMaximumBitrate` (`FLIX2HANDLE` flix, const `on2bool` bUseMaximumBitrate)
Enable/disable the maximum bitrate target value.
- `on2sc video_options_GetSwfFramerate` (const `FLIX2HANDLE` flix, `int32_t` *pSwfFramerate)
Gets the SWF framerate.
- `on2sc video_options_SetSwfFramerate` (`FLIX2HANDLE` flix, const `int32_t` swfFramerate)
Sets the SWF framerate.
- `on2sc video_options_GetVideoCodec` (const `FLIX2HANDLE` flix, `FE2_VideoCodec` *pVideoCodec)
Determine which video codec will be used.
- `on2sc video_options_SetVideoCodec` (`FLIX2HANDLE` flix, const `FE2_VideoCodec` videoCodec)
Select the video codec to be used.
- `on2sc video_options_GetAlphaPercentage` (const `FLIX2HANDLE` flix, `int32_t` *percentage)
Gets the percentage of the video bitrate to be used to encode the alpha channel.
- `on2sc video_options_SetAlphaPercentage` (`FLIX2HANDLE` flix, const `int32_t` percentage)
Sets the percentage of the video bitrate to be used to encode the alpha channel.
- `on2sc video_options_AddFLVCuePoint` (`FLIX2HANDLE` flix, const char *pName, const double time, const `FE2_CuePointType` type)
Adds a cue point to the FLV.
- `on2sc video_options_AddFLVCuePointParameter` (`FLIX2HANDLE` flix, const char *pCuePointName, const char *pName, const char *pValue)
Adds a name/value parameter to an already added cue point.
- `on2sc video_options_GetCompressMode` (const `FLIX2HANDLE` flix, `FE2_CompressMode` *mode)
Get the current VP6 compression mode.
- `on2sc video_options_SetCompressMode` (`FLIX2HANDLE` flix, `FE2_CompressMode` mode)
Set the VP6 compression mode.

12.54.1 Enumeration Type Documentation

12.54.1.1 enum FE2_CuePointType

Cue point type for use with [video_options_AddFLVCuePoint\(\)](#).

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_FLV](#) and the [FE2_FLV_CUEPT_EVENT/FE2_FLV_CUEPT_NAV](#) parameter. This enumeration will be removed in a future release.

Enumerator:

CUE_EVENT Trigger an event at a specified point

CUE_NAVIGATION Allows the user to seek to a specified point

Definition at line 52 of file video_options.h.

12.54.1.2 enum FE2_VideoCodec

Output video codec types, influences quality/compatibility. For use in calls to [video_options_SetVideoCodec\(\)](#) and [video_options_GetVideoCodec\(\)](#)

Deprecated

Use the [Codec Interface](#). This enumeration will be removed in a future release. See also: [Video Codecs](#).

Enumerator:

CODEC_NULL place holder, not for external use

CODEC_H263 H263 codec

CODEC_SCREENVIDEO Screen Video codec, NOT supported

CODEC_VP6 VP6 (Flash8) codec

CODEC_VP6ALPHA VP6 + encoded alpha channel

Definition at line 38 of file video_options.h.

12.54.2 Function Documentation

12.54.2.1 on2sc video_options_AddFLVCuePoint (FLIX2HANDLE *flx*, const char * *pName*, const double *time*, const FE2_CuePointType *type*)

Adds a cue point to the FLV.

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *pName* Name of the cue point.

← *time* Time of the cue point in seconds.

← *type* Cue point type.

Return values:*ON2_OK* on success*ON2_INVALID_PARAMS* should one or more of the preconditions fail*ON2_NO_MEM* an error occurred allocating memory for the new cue point.**Precondition:***flix* is not NULL*pName* is not NULL*time* >= 0.0*type* is a valid member of *FE2_CuePointType***Note:**

Cue points can only be added to FLV files.

May add multiple cue points.

For more information on cue points please see <http://livedocs.macromedia.com/flash/8/main/wwhelp/Parts&file=00001574.html>**Deprecated**Use the [Muxer Interface](#) along with *FE2_MUXER_FLV* and the *FE2_FLV_CUEPT_EVENT/FE2_FLV_CUEPT_NAV* parameter. This function will be removed in a future release.**12.54.2.2 on2sc video_options_AddFLVCuePointParameter (FLIX2HANDLE *flix*, const char * *pCuePointName*, const char * *pName*, const char * *pValue*)**

Adds a name/value parameter to an already added cue point.

Parameters:← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)← *pCuePointName* Name of the cue point to add the parameter to.← *pName* Name of the parameter.← *pValue* Value of the parameter.**Return values:***ON2_OK* on success*ON2_INVALID_PARAMS* should one or more of the preconditions fail*ON2_NO_MEM* an error occurred allocating memory for the new cue point.**Precondition:***flix* is not NULL*pCuePointName* is not NULL*pName* is not NULL*pValue* is not NULL*pCuePointName* has already been added to the cue point list**Note:**

Cue points can only be added to FLV files.

May add multiple parameters to a cue point.

Deprecated

Use the [Muxer Interface](#) along with [FE2_MUXER_FLV](#) and the [FE2_FLV_CUEPT_PARAM](#) parameter. This function will be removed in a future release.

12.54.2.3 on2sc video_options_GetAlphaPercentage (const FLIX2HANDLE *flix*, int32_t * *percentage*)

Gets the percentage of the video bitrate to be used to encode the alpha channel.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *percentage* Alpha percentage

Return values:

[ON2_OK](#) Success.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#) along with [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6A_ALPHA_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.4 on2sc video_options_GetCompressMode (const FLIX2HANDLE *flix*, FE2_CompressMode * *mode*)

Get the current VP6 compression mode.

Parameters:

← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
 → *mode* Compression mode (one of [FE2_CompressMode](#))

Return values:

[ON2_OK](#) The rate control type was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#) along with [FE2_CODEC_VP6](#) or [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6_CXMODE](#) or [FE2_VP6A_CXMODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.5 on2sc video_options_GetImageQuality (const FLIX2HANDLE *flix*, int32_t * *lpImageQuality*)

Get the current image quality factor.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *lpImageQuality* Image quality (0..100)

Return values:

[ON2_OK](#) The quality value was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

The image quality functions have been deprecated in favor of the maximum bitrate functions. Use [video_options_GetMaximumBitrate\(\)](#) instead.

12.54.2.6 on2sc video_options_GetKeyframeInterval (const FLIX2HANDLE *flix*, int32_t * *lpKeyframeInterval*)

Get the current interval between keyframes.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *lpKeyframeInterval* Keyframe interval (in frames)

Return values:

[ON2_OK](#) The keyframe interval was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFFREQ](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.7 on2sc video_options_GetKeyframeIntervalType (const FLIX2HANDLE *flix*, FE2_VideoKeyframeTypes * *pKeyframeIntervalType*)

Get the current keyframe interval type.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pKeyframeIntervalType* Keyframe interval type (one of [FE2_VideoKeyframeTypes](#))

Return values:

[ON2_OK](#) The keyframe interval type was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFINTTYPE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.8 on2sc video_options_GetMaximumBitrate (const FLIX2HANDLE *flx*, int32_t * *lpMaximumBitrate*)

Get the current maximum bitrate target.

Parameters:

← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *lpMaximumBitrate* Target bitrate (in Kbps)

Return values:

[ON2_OK](#) The keyframe interval type was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.9 on2sc video_options_GetRateControlType (const FLIX2HANDLE *flx*, FE2_VideoBitrateControls * *pRateControlType*)

Get the current rate control type.

Parameters:

← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

→ *pRateControlType* Rate control type (one of [FE2_VideoBitrateControls](#))

Return values:

[ON2_OK](#) The rate control type was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_RC_MODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.10 on2sc video_options_GetSwfFramerate (const FLIX2HANDLE *flx*, int32_t * *pSwfFramerate*)

Gets the SWF framerate.

Parameters:

- ← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *pSwfFramerate* The SWF framerate.

Return values:

- [ON2_OK](#) The engine successfully returned SWF framerate.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

- flx* is not NULL
- pSwfFramerate* is not NULL

Deprecated

Please use [swf_options_GetSwfFramerateAsDouble\(\)](#) to allow for non integer framerates.

12.54.2.11 on2sc video_options_GetUseMaximumBitrate (const FLIX2HANDLE *flx*, on2bool * *bpUseMaximumBitrate*)

Determine if the maximum bitrate target will be used.

Parameters:

- ← *flx* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *bpUseMaximumBitrate* Variable to update with the current bitrate enable status

Return values:

- [ON2_OK](#) The bitrate enable status was successfully retrieved from the engine.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.12 `on2sc video_options_GetVideoCodec (const FLIX2HANDLE flix, FE2_VideoCodec * pVideoCodec)`

Determine which video codec will be used.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- *pVideoCodec* Current video codec (one of [FE2_VideoCodec](#))

Return values:

[ON2_OK](#) The selected video codec was successfully retrieved from the engine.

Note:

If a value other than [ON2_OK](#) is returned, the output variables should not be considered valid.

Deprecated

Use the [Codec Interface](#). This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.13 `on2sc video_options_SetAlphaPercentage (FLIX2HANDLE flix, const int32_t percentage)`

Sets the percentage of the video bitrate to be used to encode the alpha channel.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *percentage* Alpha percentage

Return values:

[ON2_OK](#) Success.

[ON2_INVALID_PARAMS](#) The value is out of range.

Note:

By default, the percentage is 15%.

The range is 0-100%.

The codec type must be [CODEC_VP6ALPHA](#).

For example if the maximum video bitrate was set to 500 through [video_options_SetMaximumBitrate\(\)](#), and the alpha percentage was set to 15% then the bitrate for the VP6 video would be 425 and the alpha bitrate would be 75.

Deprecated

Use the [Codec Interface](#) along with [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6A_ALPHA_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.14 on2sc video_options_SetCompressMode (FLIX2HANDLE *flix*, FE2_CompressMode *mode*)

Set the VP6 compression mode.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *mode* Compression mode (one of [FE2_CompressMode](#))

Return values:

[ON2_OK](#) The compression mode was successfully set in the engine.

[ON2_INVALID_PARAMS](#) The value is out of range.

Note:

This setting affects VP6 only. [COMPRESSMODE_BEST](#) provides a slightly better quality image, but takes approximately twice as long to encode as VP6 set to [COMPRESSMODE_GOOD](#). The default value is [COMPRESSMODE_GOOD](#).

Deprecated

Use the [Codec Interface](#) along with [FE2_CODEC_VP6](#) or [FE2_CODEC_VP6ALPHA](#) and the [FE2_VP6_CXMODE](#) or [FE2_VP6A_CXMODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.15 on2sc video_options_SetImageQuality (FLIX2HANDLE *flix*, const int32_t *ImageQuality*)

Set the image quality factor. Determines the image "quality" level. Higher numbers will generally result in higher quality video, at the cost of higher bitrates and file sizes.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *ImageQuality* Quality Factor (0 to 100)

Return values:

[ON2_OK](#) The quality value was successfully set in the engine.

[ON2_INVALID_PARAMS](#) The value is out of range.

Note:

The default value is 75.

Deprecated

The image quality functions have been deprecated in favor of the maximum bitrate functions. Use [video_options_GetMaximumBitrate\(\)](#) instead.

12.54.2.16 on2sc video_options_SetKeyframeInterval (FLIX2HANDLE *flix*, const int32_t *lKeyframeInterval*)

Set the interval between keyframes. Determines the interval (in frames) between keyframes. Keyframes "refresh" the player with the best possible quality image, and subsequent images are derived from that image. In addition, the keyframe interval determines the granularity at which seeking can happen (i.e., the player can only seek to a keyframe). Reducing the keyframe interval increases the number of seekable points in the video. However, keyframes take significantly more bits to encode than non-keyframes, which can result in unnecessarily large files if the value is set too low. For most purposes, the default value is preferred.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *lKeyframeInterval* Keyframe interval (in frames)

Return values:

- [ON2_OK](#) The keyframe interval was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

The default value is $12 * \text{framerate}$ (12 seconds worth)

Deprecated

Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFFREQ](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.17 on2sc video_options_SetKeyframeIntervalType (FLIX2HANDLE *flix*, const FE2_VideoKeyframeTypes *keyframeIntervalType*)

Set the keyframe interval type. Two keyframe modes are supported, [MAX_KEYFRAMES](#) (the default) and [FIXED_KEYFRAMES](#). These specify that the keyframe interval set using [video_options_SetKeyframeInterval\(\)](#) is either the maximum interval between keyframes or a fixed interval, respectively. In general, the compression codec does the best job of deciding when keyframes should be used, so setting the maximum interval is usually more appropriate.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *keyframeIntervalType* Keyframe interval type (one of [FE2_VideoKeyframeTypes](#))

Return values:

- [ON2_OK](#) The keyframe interval was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

The default value is [MAX_KEYFRAMES](#)

Deprecated

Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_KFINTTYPE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.18 on2sc video_options_SetMaximumBitrate (FLIX2HANDLE *flix*, const int32_t *lMaximumBitrate*)

Set the maximum bitrate target. Sets the bitrate the compressor will target when encoding the video.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *lMaximumBitrate* Bitrate target (in Kbps)

Return values:

- [ON2_OK](#) The target bitrate was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

This bitrate should be a reasonable number with respect to the other video parameters. To determine if a particular value is reasonable or not, first calculate the number of bits per pixel it represents, given the bitrate in Kbps, framerate in frames per second, and width and height in pixels, using this equation:

$$bpp = \frac{bitrate * 1024}{width * height * framerate}$$

The H.263 codec is usable down to 0.05 bpp. For lower bitrates, VP6 must be used.

The default value is 448 Kbps.

Setting this value will have no effect unless enabled via [video_options_SetUseMaximumBitrate\(\)](#)

Deprecated

Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

See also:

[Video Codecs](#)

12.54.2.19 on2sc video_options_SetRateControlType (FLIX2HANDLE *flix*, const FE2_VideoBitrateControls *rateControlType*)

Set the rate control type.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *rateControlType* Rate control type (one of [FE2_VideoBitrateControls](#))

Return values:

- [ON2_OK](#) The rate control type was successfully set in the engine.
- [ON2_INVALID_PARAMS](#) The value is out of range.

Note:

The default value is [VBR_2PASSControl](#).

Deprecated

Use the [Codec Interface](#) along with the [FE2_VCODECPARAM_RC_MODE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.20 `on2sc video_options_SetSwfFramerate (FLIX2HANDLE flix, const int32_t swfFramerate)`

Sets the SWF framerate.

Parameters:

- ← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *swfFramerate* The SWF framerate.

Return values:

- [ON2_OK](#) The engine successfully set the SWF framerate.
- [ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

Deprecated

Please use [swf_options_SetSwfFramerateAsDouble\(\)](#) to allow for non integer framerates.

12.54.2.21 `on2sc video_options_SetUseMaximumBitrate (FLIX2HANDLE flix, const on2bool bUseMaximumBitrate)`

Enable/disable the maximum bitrate target value.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)
- ← *bUseMaximumBitrate* New bitrate enable status. [on2true](#) to enable, [on2false](#) to disable.

Return values:

- [ON2_OK](#) The bitrate enable status was successfully set in the engine.

Note:

By default, the deinterlace filter is disabled.

Deprecated

Use the [Codec Interface](#) along with the [FE2_CODECPARAM_BITRATE](#) parameter. This function will be removed in a future release. See also: [Video Codecs](#).

12.54.2.22 `on2sc video_options_SetVideoCodec (FLIX2HANDLE flix, const FE2_VideoCodec videoCodec)`

Select the video codec to be used.

Parameters:

- ← *flix* Handle to the Flix Engine returned from [Flix2_Create\(\)](#) or [Flix2_CreateEx\(\)](#)

← *videoCodec* New video codec (one of [FE2_VideoCodec](#))

Return values:

[ON2_OK](#) The video codec was successfully set in the engine.

[ON2_INVALID_PARAMS](#) The value is out of range.

Note:

By default, the [CODEC_VP6](#) will be used.

Deprecated

Use the [Codec Interface](#). This function will be removed in a future release. See also: [Video Codecs](#).

12.55 Flix Engine COM

Data Structures

- interface `flixfengine_com::IFlixPlgn`
Interface for accessing [Codec](#), [Filter](#) and [Muxer](#) functions.
- interface `flixfengine_com::IEncodingStatus`
Interface for accessing [encoding statistics](#) functions.
- interface `flixfengine_com::ISwfOptions`
Interface for accessing [SWF creation](#) functions.
- interface `flixfengine_com::IVideoOptions`
Interface for accessing non-deprecated [video_options_](#) functions.*
- interface `flixfengine_com::IFlix`
Main interface for accessing [engine](#) functions and obtaining instances of other COM interfaces.

Namespaces

- namespace `flixfengine_com`

12.55.1 Detailed Description

The COM interface provides a thin wrapper for access to non-deprecated [Flix Engine](#) functions.

Usage Notes:

- For consistency with COM, all Flix Engine COM functions return an HRESULT. The `on2sc` returned from the corresponding Flix Engine function is therefore mapped to an HRESULT:

```
HRESULT hr= MAKE_HRESULT(SEVERITY_ERROR, FACILITY_NULL, sc);
```

Should the language not provide access to the HRESULT returned from the function, the `sc` property within each interface stores the last return code.

- For consistency with COM, all strings are represented using a BSTR. Some Flix Engine functions, however, only support ASCII strings, i.e. those defined as taking a `char*` rather than an `on2tc*`. In this case the value passed to the interface will be converted before the call is made. Check the notes associated with the interface for further details.
- Enumeration values should be available as constants if the language has the ability to extract them from the type library, cf. [samples](#).
- String constants (e.g., filter/codec names/params) are available as properties within `flixfengine_com::IFlix`. The documentation for each property refers back to the corresponding filter/codec.

Attention:

[flixengine_com](#) is a dual interface, meaning it exposes methods and properties via `IDispatch` as well as directly through the VTBL.

If using a language that supports direct VTBL binding (e.g., [ASP.NET](#), [C#](#), [VB.NET](#)) the assembly file, *Interop.flixengine_com.dll*, provided in `%PROGRAMFILES%\On2 Flix Engine\lib` will need to be updated should the major (M) or minor (m) version (x.M.m.x) of the COM library increase. If the application is based on the [C++](#) sample it will need to be rebuilt in this case, *flixengine_com_i.c* being an analogue of the *Interop.flixengine_com.dll* assembly.

Chapter 13

Namespace Documentation

13.1 flxengine_com Namespace Reference

Data Structures

- interface [IFlixPlgn](#)
Interface for accessing [Codec](#), [Filter](#) and [Muxer](#) functions.
- interface [IEncodingStatus](#)
Interface for accessing [encoding statistics](#) functions.
- interface [ISwfOptions](#)
Interface for accessing [SWF creation](#) functions.
- interface [IVideoOptions](#)
Interface for accessing non-deprecated [video_options_](#) functions.*
- interface [IFlix](#)
Main interface for accessing [engine](#) functions and obtaining instances of other COM interfaces.

Chapter 14

Data Structure Documentation

14.1 flxengine_com::IEncodingStatus Interface Reference

Interface for accessing [encoding statistics](#) functions.

```
import "flxengine_com.idl";
```

Public Member Functions

- HRESULT [averageBitrate](#) ([out, retval] on2s32 *pBitrate)
Wrapper for [encoding_status_GetAverageBitrate\(\)](#).
- HRESULT [averageFramesize](#) ([out, retval] on2s32 *pFramesize)
Wrapper for [encoding_status_GetAverageFramesize\(\)](#).
- HRESULT [maximumFramesize](#) ([out, retval] on2s32 *pMaxFramesize)
Wrapper for [encoding_status_GetMaximumFramesize\(\)](#).
- HRESULT [minimumFramesize](#) ([out, retval] on2s32 *pMinFramesize)
Wrapper for [encoding_status_GetMinimumFramesize\(\)](#).
- HRESULT [totalFrames](#) ([out, retval] on2s32 *pTotalFrames)
Wrapper for [encoding_status_GetTotalFrames\(\)](#).
- HRESULT [elapsedTime](#) ([out, retval] on2s32 *pElapsedTime)
Wrapper for [encoding_status_GetElapsedTime\(\)](#).
- HRESULT [endTime](#) ([out, retval] on2s32 *pEndTime)
Wrapper for [encoding_status_GetEndTime\(\)](#).
- HRESULT [startTime](#) ([out, retval] on2s32 *pStartTime)
Wrapper for [encoding_status_GetStartTime\(\)](#).
- HRESULT [percentComplete](#) ([out, retval] on2s32 *percent)
Wrapper for [encoding_status_PercentComplete\(\)](#).

Properties

- [on2sc](#) `sc` [get]

[on2sc](#) from the last Flix Engine function called within this interface

14.1.1 Detailed Description

Interface for accessing [encoding statistics](#) functions.

Definition at line 155 of file `flxengine_com.idl`.

14.1.2 Member Function Documentation

14.1.2.1 HRESULT flxengine_com::IEncodingStatus::averageBitrate ([out, retval] on2s32 * *pBitrate*)

Wrapper for [encoding_status_GetAverageBitrate\(\)](#).

14.1.2.2 HRESULT flxengine_com::IEncodingStatus::averageFramesize ([out, retval] on2s32 * *pFramesize*)

Wrapper for [encoding_status_GetAverageFramesize\(\)](#).

14.1.2.3 HRESULT flxengine_com::IEncodingStatus::elapsedTime ([out, retval] on2s32 * *pElapsedTime*)

Wrapper for [encoding_status_GetElapsedTime\(\)](#).

14.1.2.4 HRESULT flxengine_com::IEncodingStatus::endTime ([out, retval] on2s32 * *pEndTime*)

Wrapper for [encoding_status_GetEndTime\(\)](#).

14.1.2.5 HRESULT flxengine_com::IEncodingStatus::maximumFramesize ([out, retval] on2s32 * *pMaxFramesize*)

Wrapper for [encoding_status_GetMaximumFramesize\(\)](#).

14.1.2.6 HRESULT flxengine_com::IEncodingStatus::minimumFramesize ([out, retval] on2s32 * *pMinFramesize*)

Wrapper for [encoding_status_GetMinimumFramesize\(\)](#).

14.1.2.7 HRESULT flxengine_com::IEncodingStatus::percentComplete ([out, retval] on2s32 * *percent*)

Wrapper for [encoding_status_PercentComplete\(\)](#).

14.1.2.8 HRESULT flxengine_com::IEncodingStatus::startTime ([out, retval] on2s32 * *pStartTime*)

Wrapper for [encoding_status_GetStartTime\(\)](#).

14.1.2.9 HRESULT flxengine_com::IEncodingStatus::totalFrames ([out, retval] on2s32 * *pTotalFrames*)

Wrapper for [encoding_status_GetTotalFrames\(\)](#).

14.1.3 Property Documentation

14.1.3.1 on2sc flxengine_com::IEncodingStatus::sc [get]

[on2sc](#) from the last Flix Engine function called within this interface

The documentation for this interface was generated from the following file:

- [flxengine_com.idl](#)

14.2 flixengine_com::IFlix Interface Reference

Main interface for accessing [engine](#) functions and obtaining instances of other COM interfaces.

```
import "flixengine_com.idl";
```

Public Member Functions

- HRESULT [com_version](#) ([out, retval] BSTR *com_v)
Returns the COM library version from the file version resource.
- HRESULT [version](#) ([out, retval] BSTR *v)
Wrapper for [Flix2_Version\(\)](#).
- HRESULT [copyright](#) ([out, retval] BSTR *c)
Wrapper for [Flix2_Copyright\(\)](#).
- HRESULT [setOutputFile](#) (const BSTR outputFile)
Wrapper for [Flix2_SetOutputFile\(\)](#).
- HRESULT [setOverwriteExistingFiles](#) (const [on2bool](#) bOverwriteExistingFiles)
Wrapper for [Flix2_SetOverwriteExistingFiles\(\)](#).
- HRESULT [getOverwriteExistingFiles](#) ([out, retval] [on2bool](#) *bpOverwriteExistingFiles)
Wrapper for [Flix2_GetOverwriteExistingFiles\(\)](#).
- HRESULT [setExportAudio](#) (const [on2bool](#) bExportAudio)
Wrapper for [Flix2_SetExportAudio\(\)](#).
- HRESULT [getExportAudio](#) ([out, retval] [on2bool](#) *bpExportAudio)
Wrapper for [Flix2_GetExportAudio\(\)](#).
- HRESULT [setExportVideo](#) (const [on2bool](#) bExportVideo)
Wrapper for [Flix2_SetExportVideo\(\)](#).
- HRESULT [getExportVideo](#) ([out, retval] [on2bool](#) *bpExportVideo)
Wrapper for [Flix2_GetExportVideo\(\)](#).
- HRESULT [setExportVideoType](#) (const [FE2_ExportedVideoType](#) exportVideoType)
Wrapper for [Flix2_SetExportVideoType\(\)](#).
- HRESULT [getExportVideoType](#) ([out, retval] [FE2_ExportedVideoType](#) *pExportVideoType)
Wrapper for [Flix2_GetExportVideoType\(\)](#).
- HRESULT [setInputFile](#) (const BSTR inputFile)
Wrapper for [Flix2_SetInputFile\(\)](#).
- HRESULT [getSourceDuration](#) ([out, retval] [on2s32](#) *duration)
Wrapper for [Flix2_GetSourceDuration\(\)](#).

- HRESULT [encode](#) ()
Wrapper for [Flix2_Encode\(\)](#).
- HRESULT [stopEncoding](#) ()
Wrapper for [Flix2_StopEncoding\(\)](#).
- HRESULT [reset](#) ()
Wrapper for [Flix2_Reset\(\)](#).
- HRESULT [validate](#) ()
Wrapper for [Flix2_Validate\(\)](#).
- HRESULT [isEncoderRunning](#) ([out, retval] [on2bool](#) *bpIsEncoderRunning)
Wrapper for [Flix2_IsEncoderRunning\(\)](#).
- HRESULT [getEncoderState](#) ([out, retval] [FE2_EncState](#) *pEncState)
Wrapper for [Flix2_GetEncoderState\(\)](#).
- HRESULT [setLogLevel](#) ([on2s32](#) level)
Wrapper for [Flix2_SetLogLevel\(\)](#).
- HRESULT [getLogLevel](#) ([out, retval] [on2s32](#) *level)
Wrapper for [Flix2_GetLogLevel\(\)](#).
- HRESULT [setLogPath](#) (BSTR logpath)
Wrapper for [Flix2_SetLogPath\(\)](#).
- HRESULT [addFilter](#) (const BSTR plgn_name,[out, retval] [IFlixPlgn](#) **pPlgn)
Wrapper for [Flix2_AddFilter\(\)](#).
- HRESULT [addCodec](#) (const BSTR plgn_name,[out, retval] [IFlixPlgn](#) **pPlgn)
Wrapper for [Flix2_AddCodec\(\)](#).
- HRESULT [addMuxer](#) (const BSTR plgn_name,[out, retval] [IFlixPlgn](#) **pPlgn)
Wrapper for [Flix2_AddMuxer\(\)](#).
- HRESULT [errno_](#) ([out] [FE2_errno](#) *flxerrno,[out] [on2s32](#) *syserrno)
Wrapper for [Flix2_Errno\(\)](#).
- HRESULT [flxerrno](#) ([out, retval] [FE2_errno](#) *flxerrno)
Wrapper for [Flix2_Errno\(\)](#).
- HRESULT [syserrno](#) ([out, retval] [on2s32](#) *syserrno)
Wrapper for [Flix2_Errno\(\)](#).
- HRESULT [encodingStatus](#) ([out, retval] [IEncodingStatus](#) **encstatus)
Returns an instance of [IEncodingStatus](#).
- HRESULT [swfOptions](#) ([out, retval] [ISwfOptions](#) **swfoptions)
Returns an instance of [ISwfOptions](#).

- HRESULT [videoOptions](#) ([out, retval] [IVideoOptions](#) **vidoptions)

Returns an instance of [IVideoOptions](#).

Properties

- [on2sc](#) [sc](#) [get]

on2sc from the last Flix Engine function called within this interface

Codec/Filter/Muxer constants

- BSTR [FE2_3G2_FASTSTART](#) [get]
[FE2_3G2_FASTSTART](#)
- BSTR [FE2_3GP_FASTSTART](#) [get]
[FE2_3GP_FASTSTART](#)
- BSTR [FE2_AACPLUS_BITRATE](#) [get]
[FE2_AACPLUS_BITRATE](#)
- BSTR [FE2_AACPLUS_PARAMETRIC_STEREO](#) [get]
[FE2_AACPLUS_PARAMETRIC_STEREO](#)
- BSTR [FE2_AAC_BITRATE](#) [get]
[FE2_AAC_BITRATE](#)
- BSTR [FE2_ADAPTIVE_DEINTERLACE_MODE](#) [get]
[FE2_ADAPTIVE_DEINTERLACE_MODE](#)
- BSTR [FE2_AMR_BITRATE](#) [get]
[FE2_AMR_BITRATE](#)
- BSTR [FE2_BCHS_BRIGHTNESS](#) [get]
[FE2_BCHS_BRIGHTNESS](#)
- BSTR [FE2_BCHS_CONTRAST](#) [get]
[FE2_BCHS_CONTRAST](#)
- BSTR [FE2_BCHS_HUE](#) [get]
[FE2_BCHS_HUE](#)
- BSTR [FE2_BCHS_SATURATION](#) [get]
[FE2_BCHS_SATURATION](#)
- BSTR [FE2_BLUR_FILTER](#) [get]
[FE2_BLUR_FILTER](#)
- BSTR [FE2_BLUR_MASKSIZE](#) [get]
[FE2_BLUR_MASKSIZE](#)
- BSTR [FE2_CODECPARAM_BITRATE](#) [get]
[FE2_CODECPARAM_BITRATE](#)

- BSTR [FE2_CODEC_AAC](#) [get]
[FE2_CODEC_AAC](#)
- BSTR [FE2_CODEC_AACPLUS](#) [get]
[FE2_CODEC_AACPLUS](#)
- BSTR [FE2_CODEC_AMR_NB](#) [get]
[FE2_CODEC_AMR_NB](#)
- BSTR [FE2_CODEC_H263](#) [get]
[FE2_CODEC_H263](#)
- BSTR [FE2_CODEC_H263_BASELINE](#) [get]
[FE2_CODEC_H263_BASELINE](#)
- BSTR [FE2_CODEC_H264](#) [get]
[FE2_CODEC_H264](#)
- BSTR [FE2_CODEC_LAME](#) [get]
[FE2_CODEC_LAME](#)
- BSTR [FE2_CODEC_PCM](#) [get]
[FE2_CODEC_PCM](#)
- BSTR [FE2_CODEC_VP6](#) [get]
[FE2_CODEC_VP6](#)
- BSTR [FE2_CODEC_VP6ALPHA](#) [get]
[FE2_CODEC_VP6ALPHA](#)
- BSTR [FE2_CROP_BOTTOM](#) [get]
[FE2_CROP_BOTTOM](#)
- BSTR [FE2_CROP_LEFT](#) [get]
[FE2_CROP_LEFT](#)
- BSTR [FE2_CROP_RIGHT](#) [get]
[FE2_CROP_RIGHT](#)
- BSTR [FE2_CROP_TOP](#) [get]
[FE2_CROP_TOP](#)
- BSTR [FE2_CUT_START_SEC](#) [get]
[FE2_CUT_START_SEC](#)
- BSTR [FE2_CUT_STOP_SEC](#) [get]
[FE2_CUT_STOP_SEC](#)
- BSTR [FE2_CUT_USE_SEEK](#) [get]
[FE2_CUT_USE_SEEK](#)
- BSTR [FE2_DENOISE_NOISE_LEVEL](#) [get]
[FE2_DENOISE_NOISE_LEVEL](#)
- BSTR [FE2_FILTER_ADAPTIVE_DEINTERLACE](#) [get]

FE2_FILTER_ADAPTIVE_DEINTERLACE

- BSTR *FE2_FILTER_BCHS* [get]
FE2_FILTER_BCHS
- BSTR *FE2_FILTER_BLUR* [get]
FE2_FILTER_BLUR
- BSTR *FE2_FILTER_CROP* [get]
FE2_FILTER_CROP
- BSTR *FE2_FILTER_CUT* [get]
FE2_FILTER_CUT
- BSTR *FE2_FILTER_DENOISE* [get]
FE2_FILTER_DENOISE
- BSTR *FE2_FILTER_FRAMERATE* [get]
FE2_FILTER_FRAMERATE
- BSTR *FE2_FILTER_HIGHPASS* [get]
FE2_FILTER_HIGHPASS
- BSTR *FE2_FILTER_LOWPASS* [get]
FE2_FILTER_LOWPASS
- BSTR *FE2_FILTER_MIRROR* [get]
FE2_FILTER_MIRROR
- BSTR *FE2_FILTER_OVERLAY* [get]
FE2_FILTER_OVERLAY
- BSTR *FE2_FILTER_PNGEX* [get]
FE2_FILTER_PNGEX
- BSTR *FE2_FILTER_RESAMPLE* [get]
FE2_FILTER_RESAMPLE
- BSTR *FE2_FILTER_ROTATE* [get]
FE2_FILTER_ROTATE
- BSTR *FE2_FILTER_SCALE* [get]
FE2_FILTER_SCALE
- BSTR *FE2_FILTER_SHARPEN* [get]
FE2_FILTER_SHARPEN
- BSTR *FE2_FLV_CUEPT_EVENT* [get]
FE2_FLV_CUEPT_EVENT
- BSTR *FE2_FLV_CUEPT_NAV* [get]
FE2_FLV_CUEPT_NAV
- BSTR *FE2_FLV_CUEPT_PARAM* [get]
FE2_FLV_CUEPT_PARAM

- BSTR [FE2_FRAMERATE_DECIMATE](#) [get]
[FE2_FRAMERATE_DECIMATE](#)
- BSTR [FE2_FRAMERATE_FPS](#) [get]
[FE2_FRAMERATE_FPS](#)
- BSTR [FE2_FXM_CUEPT_EVENT](#) [get]
[FE2_FXM_CUEPT_EVENT](#)
- BSTR [FE2_FXM_CUEPT_NAV](#) [get]
[FE2_FXM_CUEPT_NAV](#)
- BSTR [FE2_FXM_CUEPT_PARAM](#) [get]
[FE2_FXM_CUEPT_PARAM](#)
- BSTR [FE2_H263_BITRATE](#) [get]
[FE2_H263_BITRATE](#)
- BSTR [FE2_H263_KFFREQ](#) [get]
[FE2_H263_KFFREQ](#)
- BSTR [FE2_H263_KFINTTYPE](#) [get]
[FE2_H263_KFINTTYPE](#)
- BSTR [FE2_H263_MAX_Q](#) [get]
[FE2_H263_MAX_Q](#)
- BSTR [FE2_H263_MIN_Q](#) [get]
[FE2_H263_MIN_Q](#)
- BSTR [FE2_H263_RC_MODE](#) [get]
[FE2_H263_RC_MODE](#)
- BSTR [FE2_H264_BITRATE](#) [get]
[FE2_H264_BITRATE](#)
- BSTR [FE2_H264_B_FRAME_RATE](#) [get]
[FE2_H264_B_FRAME_RATE](#)
- BSTR [FE2_H264_KFFREQ](#) [get]
[FE2_H264_KFFREQ](#)
- BSTR [FE2_H264_KFINTTYPE](#) [get]
[FE2_H264_KFINTTYPE](#)
- BSTR [FE2_H264_PROFILE](#) [get]
[FE2_H264_PROFILE](#)
- BSTR [FE2_H264_RC_MODE](#) [get]
[FE2_H264_RC_MODE](#)
- BSTR [FE2_HIGHPASS_CUTOFF](#) [get]
[FE2_HIGHPASS_CUTOFF](#)
- BSTR [FE2_HIGHPASS_Q](#) [get]

FE2_HIGHPASS_Q

- BSTR *FE2_ISOMEDIA_FASTSTART* [get]
FE2_ISOMEDIA_FASTSTART
- BSTR *FE2_LAME_BITRATE* [get]
FE2_LAME_BITRATE
- BSTR *FE2_LAME_CHANNELS* [get]
FE2_LAME_CHANNELS
- BSTR *FE2_LAME_QUALITY* [get]
FE2_LAME_QUALITY
- BSTR *FE2_LAME_RC_MODE* [get]
FE2_LAME_RC_MODE
- BSTR *FE2_LOWPASS_CUTOFF* [get]
FE2_LOWPASS_CUTOFF
- BSTR *FE2_LOWPASS_Q* [get]
FE2_LOWPASS_Q
- BSTR *FE2_MIRROR_HORIZONTAL* [get]
FE2_MIRROR_HORIZONTAL
- BSTR *FE2_MIRROR_VERTICAL* [get]
FE2_MIRROR_VERTICAL
- BSTR *FE2_MOV_FASTSTART* [get]
FE2_MOV_FASTSTART
- BSTR *FE2_MP4_FASTSTART* [get]
FE2_MP4_FASTSTART
- BSTR *FE2_MUXER_3G2* [get]
FE2_MUXER_3G2
- BSTR *FE2_MUXER_3GP* [get]
FE2_MUXER_3GP
- BSTR *FE2_MUXER_FLV* [get]
FE2_MUXER_FLV
- BSTR *FE2_MUXER_FXM* [get]
FE2_MUXER_FXM
- BSTR *FE2_MUXER_MOV* [get]
FE2_MUXER_MOV
- BSTR *FE2_MUXER_MP4* [get]
FE2_MUXER_MP4
- BSTR *FE2_MUXER_SWF* [get]
FE2_MUXER_SWF

- BSTR [FE2_OVERLAY_FILE](#) [get]
FE2_OVERLAY_FILE
- BSTR [FE2_OVERLAY_MASK_B](#) [get]
FE2_OVERLAY_MASK_B
- BSTR [FE2_OVERLAY_MASK_G](#) [get]
FE2_OVERLAY_MASK_G
- BSTR [FE2_OVERLAY_MASK_R](#) [get]
FE2_OVERLAY_MASK_R
- BSTR [FE2_OVERLAY_MASK_RGB](#) [get]
FE2_OVERLAY_MASK_RGB
- BSTR [FE2_OVERLAY_MASK_X](#) [get]
FE2_OVERLAY_MASK_X
- BSTR [FE2_OVERLAY_MASK_XY](#) [get]
FE2_OVERLAY_MASK_XY
- BSTR [FE2_OVERLAY_MASK_Y](#) [get]
FE2_OVERLAY_MASK_Y
- BSTR [FE2_OVERLAY_POS](#) [get]
FE2_OVERLAY_POS
- BSTR [FE2_OVERLAY_POS_X](#) [get]
FE2_OVERLAY_POS_X
- BSTR [FE2_OVERLAY_POS_Y](#) [get]
FE2_OVERLAY_POS_Y
- BSTR [FE2_PNGEX_AUTO_EXPORT_COUNT](#) [get]
FE2_PNGEX_AUTO_EXPORT_COUNT
- BSTR [FE2_PNGEX_AUTO_EXPORT_END_TIME](#) [get]
FE2_PNGEX_AUTO_EXPORT_END_TIME
- BSTR [FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD](#) [get]
FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD
- BSTR [FE2_PNGEX_AUTO_EXPORT_START_TIME](#) [get]
FE2_PNGEX_AUTO_EXPORT_START_TIME
- BSTR [FE2_PNGEX_COMPRESSION_LEVEL](#) [get]
FE2_PNGEX_COMPRESSION_LEVEL
- BSTR [FE2_PNGEX_ENABLE_ALPHA](#) [get]
FE2_PNGEX_ENABLE_ALPHA
- BSTR [FE2_PNGEX_EXPORT_CUE_POINTS](#) [get]
FE2_PNGEX_EXPORT_CUE_POINTS
- BSTR [FE2_PNGEX_EXPORT_FIRST_FRAME_PNG](#) [get]

FE2_PNGEX_EXPORT_FIRST_FRAME_PNG

- BSTR *FE2_PNGEX_EXPORT_INTERVAL* [get]
FE2_PNGEX_EXPORT_INTERVAL
- BSTR *FE2_PNGEX_EXPORT_TIME_STRING* [get]
FE2_PNGEX_EXPORT_TIME_STRING
- BSTR *FE2_PNGEX_FILENAME_PREFIX* [get]
FE2_PNGEX_FILENAME_PREFIX
- BSTR *FE2_PNGEX_FILENAME_SUFFIX* [get]
FE2_PNGEX_FILENAME_SUFFIX
- BSTR *FE2_PNGEX_HEIGHT* [get]
FE2_PNGEX_HEIGHT
- BSTR *FE2_PNGEX_OUTPUT_DIRECTORY* [get]
FE2_PNGEX_OUTPUT_DIRECTORY
- BSTR *FE2_PNGEX_WIDTH* [get]
FE2_PNGEX_WIDTH
- BSTR *FE2_RESAMPLE_CHANNELS* [get]
FE2_RESAMPLE_CHANNELS
- BSTR *FE2_RESAMPLE_RATE* [get]
FE2_RESAMPLE_RATE
- BSTR *FE2_ROTATE_ANGLE* [get]
FE2_ROTATE_ANGLE
- BSTR *FE2_SCALE_HEIGHT* [get]
FE2_SCALE_HEIGHT
- BSTR *FE2_SCALE_WIDTH* [get]
FE2_SCALE_WIDTH
- BSTR *FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR* [get]
FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR
- BSTR *FE2_SWF_ADD_VARIABLE* [get]
FE2_SWF_ADD_VARIABLE
- BSTR *FE2_SWF_DELETE_VARIABLE* [get]
FE2_SWF_DELETE_VARIABLE
- BSTR *FE2_SWF_EMBEDDED_URL* [get]
FE2_SWF_EMBEDDED_URL
- BSTR *FE2_SWF_EMBEDDED_URL_TARGET* [get]
FE2_SWF_EMBEDDED_URL_TARGET
- BSTR *FE2_SWF_EMBEDDED_URL_TYPE* [get]
FE2_SWF_EMBEDDED_URL_TYPE

- BSTR [FE2_SWF_FIXED_PRELOAD_PCT](#) [get]
[FE2_SWF_FIXED_PRELOAD_PCT](#)
- BSTR [FE2_SWF_FRAMERATE](#) [get]
[FE2_SWF_FRAMERATE](#)
- BSTR [FE2_SWF_HEIGHT](#) [get]
[FE2_SWF_HEIGHT](#)
- BSTR [FE2_SWF_LOOP_COUNT](#) [get]
[FE2_SWF_LOOP_COUNT](#)
- BSTR [FE2_SWF_ON_END_OPTION](#) [get]
[FE2_SWF_ON_END_OPTION](#)
- BSTR [FE2_SWF_ON_END_URL](#) [get]
[FE2_SWF_ON_END_URL](#)
- BSTR [FE2_SWF_ON_START_OPTION](#) [get]
[FE2_SWF_ON_START_OPTION](#)
- BSTR [FE2_SWF_PRELOAD_TYPE](#) [get]
[FE2_SWF_PRELOAD_TYPE](#)
- BSTR [FE2_SWF_START_BLANK_FRAME](#) [get]
[FE2_SWF_START_BLANK_FRAME](#)
- BSTR [FE2_SWF_START_WAIT_SEC](#) [get]
[FE2_SWF_START_WAIT_SEC](#)
- BSTR [FE2_SWF_WIDTH](#) [get]
[FE2_SWF_WIDTH](#)
- BSTR [FE2_VCODECPARAM_KFFREQ](#) [get]
[FE2_VCODECPARAM_KFFREQ](#)
- BSTR [FE2_VCODECPARAM_KFINTTYPE](#) [get]
[FE2_VCODECPARAM_KFINTTYPE](#)
- BSTR [FE2_VCODECPARAM_RC_MODE](#) [get]
[FE2_VCODECPARAM_RC_MODE](#)
- BSTR [FE2_VP6A_2PASS_MAX_SECTION](#) [get]
[FE2_VP6A_2PASS_MAX_SECTION](#)
- BSTR [FE2_VP6A_2PASS_MIN_SECTION](#) [get]
[FE2_VP6A_2PASS_MIN_SECTION](#)
- BSTR [FE2_VP6A_ALPHA_BITRATE](#) [get]
[FE2_VP6A_ALPHA_BITRATE](#)
- BSTR [FE2_VP6A_ALPHA_MAX_Q](#) [get]
[FE2_VP6A_ALPHA_MAX_Q](#)
- BSTR [FE2_VP6A_ALPHA_MIN_Q](#) [get]

FE2_VP6A_ALPHA_MIN_Q

- BSTR *FE2_VP6A_ALPHA_NOISE_REDUCTION* [get]
FE2_VP6A_ALPHA_NOISE_REDUCTION
- BSTR *FE2_VP6A_ALPHA_SHARPNESS* [get]
FE2_VP6A_ALPHA_SHARPNESS
- BSTR *FE2_VP6A_BITRATE* [get]
FE2_VP6A_BITRATE
- BSTR *FE2_VP6A_CXMODE* [get]
FE2_VP6A_CXMODE
- BSTR *FE2_VP6A_KFFREQ* [get]
FE2_VP6A_KFFREQ
- BSTR *FE2_VP6A_KFINTTYPE* [get]
FE2_VP6A_KFINTTYPE
- BSTR *FE2_VP6A_MAX_Q* [get]
FE2_VP6A_MAX_Q
- BSTR *FE2_VP6A_MIN_Q* [get]
FE2_VP6A_MIN_Q
- BSTR *FE2_VP6A_NOISE_REDUCTION* [get]
FE2_VP6A_NOISE_REDUCTION
- BSTR *FE2_VP6A_RC_MODE* [get]
FE2_VP6A_RC_MODE
- BSTR *FE2_VP6A_SHARPNESS* [get]
FE2_VP6A_SHARPNESS
- BSTR *FE2_VP6A_STREAM_MAX_BUFFER* [get]
FE2_VP6A_STREAM_MAX_BUFFER
- BSTR *FE2_VP6A_STREAM_OPTIMAL_BUFFER* [get]
FE2_VP6A_STREAM_OPTIMAL_BUFFER
- BSTR *FE2_VP6A_STREAM_PEAK_BITRATE* [get]
FE2_VP6A_STREAM_PEAK_BITRATE
- BSTR *FE2_VP6A_STREAM_PREBUFFER* [get]
FE2_VP6A_STREAM_PREBUFFER
- BSTR *FE2_VP6A_TEMPORAL_DOWN_WATERMARK* [get]
FE2_VP6A_TEMPORAL_DOWN_WATERMARK
- BSTR *FE2_VP6A_TEMPORAL_RESAMPLING* [get]
FE2_VP6A_TEMPORAL_RESAMPLING
- BSTR *FE2_VP6A_UNDERSHOOT_PCT* [get]
FE2_VP6A_UNDERSHOOT_PCT

- BSTR [FE2_VP6_2PASS_MAX_SECTION](#) [get]
[FE2_VP6_2PASS_MAX_SECTION](#)
- BSTR [FE2_VP6_2PASS_MIN_SECTION](#) [get]
[FE2_VP6_2PASS_MIN_SECTION](#)
- BSTR [FE2_VP6_BITRATE](#) [get]
[FE2_VP6_BITRATE](#)
- BSTR [FE2_VP6_CONCURRENCY](#) [get]
[FE2_VP6_CONCURRENCY](#)
- BSTR [FE2_VP6_CXMODE](#) [get]
[FE2_VP6_CXMODE](#)
- BSTR [FE2_VP6_KFFREQ](#) [get]
[FE2_VP6_KFFREQ](#)
- BSTR [FE2_VP6_KFINTTYPE](#) [get]
[FE2_VP6_KFINTTYPE](#)
- BSTR [FE2_VP6_MAX_Q](#) [get]
[FE2_VP6_MAX_Q](#)
- BSTR [FE2_VP6_MIN_Q](#) [get]
[FE2_VP6_MIN_Q](#)
- BSTR [FE2_VP6_NOISE_REDUCTION](#) [get]
[FE2_VP6_NOISE_REDUCTION](#)
- BSTR [FE2_VP6_PROFILE](#) [get]
[FE2_VP6_PROFILE](#)
- BSTR [FE2_VP6_RC_MODE](#) [get]
[FE2_VP6_RC_MODE](#)
- BSTR [FE2_VP6_SHARPNESS](#) [get]
[FE2_VP6_SHARPNESS](#)
- BSTR [FE2_VP6_STREAM_MAX_BUFFER](#) [get]
[FE2_VP6_STREAM_MAX_BUFFER](#)
- BSTR [FE2_VP6_STREAM_OPTIMAL_BUFFER](#) [get]
[FE2_VP6_STREAM_OPTIMAL_BUFFER](#)
- BSTR [FE2_VP6_STREAM_PEAK_BITRATE](#) [get]
[FE2_VP6_STREAM_PEAK_BITRATE](#)
- BSTR [FE2_VP6_STREAM_PREBUFFER](#) [get]
[FE2_VP6_STREAM_PREBUFFER](#)
- BSTR [FE2_VP6_TEMPORAL_DOWN_WATERMARK](#) [get]
[FE2_VP6_TEMPORAL_DOWN_WATERMARK](#)
- BSTR [FE2_VP6_TEMPORAL_RESAMPLING](#) [get]

FE2_VP6_TEMPORAL_RESAMPLING

- BSTR *FE2_VP6_UNDERSHOOT_PCT* [get]
FE2_VP6_UNDERSHOOT_PCT
- BSTR *FE2_FLV_METADATA_DISABLE* [get]
FE2_FLV_METADATA_DISABLE
- BSTR *FE2_FLV_METADATA_ENABLE* [get]
FE2_FLV_METADATA_ENABLE
- BSTR *FE2_FXM_METADATA_DISABLE* [get]
FE2_FXM_METADATA_DISABLE
- BSTR *FE2_FXM_METADATA_ENABLE* [get]
FE2_FXM_METADATA_ENABLE
- BSTR *FE2_H264_SPEED* [get]
FE2_H264_SPEED
- BSTR *FE2_CODEC_VORBIS* [get]
FE2_CODEC_VORBIS
- BSTR *FE2_MUXER_WEBM* [get]
FE2_MUXER_WEBM
- BSTR *FE2_VORBIS_BITRATE* [get]
FE2_VORBIS_BITRATE
- BSTR *FE2_CODEC_VP8* [get]
FE2_CODEC_VP8
- BSTR *FE2_VP8_2PASS_MAX_SECTION* [get]
FE2_VP8_2PASS_MAX_SECTION
- BSTR *FE2_VP8_2PASS_MIN_SECTION* [get]
FE2_VP8_2PASS_MIN_SECTION
- BSTR *FE2_VP8_ALTREF* [get]
FE2_VP8_ALTREF
- BSTR *FE2_VP8_AR_MAX_FRAMES* [get]
FE2_VP8_AR_MAX_FRAMES
- BSTR *FE2_VP8_AR_STRENGTH* [get]
FE2_VP8_AR_STRENGTH
- BSTR *FE2_VP8_AR_TYPE* [get]
FE2_VP8_AR_TYPE
- BSTR *FE2_VP8_BITRATE* [get]
FE2_VP8_BITRATE
- BSTR *FE2_VP8_CXMODE* [get]
FE2_VP8_CXMODE

- BSTR [FE2_VP8_DROP_THRESH](#) [get]
[FE2_VP8_DROP_THRESH](#)
- BSTR [FE2_VP8_KFFREQ](#) [get]
[FE2_VP8_KFFREQ](#)
- BSTR [FE2_VP8_KFINTTYPE](#) [get]
[FE2_VP8_KFINTTYPE](#)
- BSTR [FE2_VP8_LAG](#) [get]
[FE2_VP8_LAG](#)
- BSTR [FE2_VP8_MAX_Q](#) [get]
[FE2_VP8_MAX_Q](#)
- BSTR [FE2_VP8_MB_STATIC_THRESHOLD](#) [get]
[FE2_VP8_MB_STATIC_THRESHOLD](#)
- BSTR [FE2_VP8_MIN_Q](#) [get]
[FE2_VP8_MIN_Q](#)
- BSTR [FE2_VP8_NOISE_REDUCTION](#) [get]
[FE2_VP8_NOISE_REDUCTION](#)
- BSTR [FE2_VP8_OVERSHOOT_PCT](#) [get]
[FE2_VP8_OVERSHOOT_PCT](#)
- BSTR [FE2_VP8_PROFILE](#) [get]
[FE2_VP8_PROFILE](#)
- BSTR [FE2_VP8_RC_MODE](#) [get]
[FE2_VP8_RC_MODE](#)
- BSTR [FE2_VP8_SHARPNESS](#) [get]
[FE2_VP8_SHARPNESS](#)
- BSTR [FE2_VP8_STREAM_INITIAL_BUFFER](#) [get]
[FE2_VP8_STREAM_INITIAL_BUFFER](#)
- BSTR [FE2_VP8_STREAM_MAX_BUFFER](#) [get]
[FE2_VP8_STREAM_MAX_BUFFER](#)
- BSTR [FE2_VP8_STREAM_OPTIMAL_BUFFER](#) [get]
[FE2_VP8_STREAM_OPTIMAL_BUFFER](#)
- BSTR [FE2_VP8_THREADS](#) [get]
[FE2_VP8_THREADS](#)
- BSTR [FE2_VP8_TOKEN_PARTITIONS](#) [get]
[FE2_VP8_TOKEN_PARTITIONS](#)
- BSTR [FE2_VP8_UNDERSHOOT_PCT](#) [get]
[FE2_VP8_UNDERSHOOT_PCT](#)

14.2.1 Detailed Description

Main interface for accessing [engine](#) functions and obtaining instances of other COM interfaces. This is the main interface from which instances of [IFlixPlgn](#), [IEncodingStatus](#), [ISwfOptions](#), and [IVideoOptions](#) can be obtained.

Definition at line 588 of file `flixengine_com.idl`.

14.2.2 Member Function Documentation

14.2.2.1 HRESULT flxengine_com::IFlix::addCodec (const BSTR *plgn_name*, [out, retval] IFlixPlgn ** *pPlgn*)

Wrapper for [Flix2_AddCodec\(\)](#).

Note:

plgn_name is converted to a char

14.2.2.2 HRESULT flxengine_com::IFlix::addFilter (const BSTR *plgn_name*, [out, retval] IFlixPlgn ** *pPlgn*)

Wrapper for [Flix2_AddFilter\(\)](#).

Note:

plgn_name is converted to a char

14.2.2.3 HRESULT flxengine_com::IFlix::addMuxer (const BSTR *plgn_name*, [out, retval] IFlixPlgn ** *pPlgn*)

Wrapper for [Flix2_AddMuxer\(\)](#).

Note:

plgn_name is converted to a char

14.2.2.4 HRESULT flxengine_com::IFlix::com_version ([out, retval] BSTR * *com_v*)

Returns the COM library version from the file version resource.

14.2.2.5 HRESULT flxengine_com::IFlix::copyright ([out, retval] BSTR * *c*)

Wrapper for [Flix2_Copyright\(\)](#).

14.2.2.6 HRESULT flxengine_com::IFlix::encode ()

Wrapper for [Flix2_Encode\(\)](#).

**14.2.2.7 HRESULT flxengine_com::IFlix::encodingStatus ([out, retval] IEncodingStatus **
encstatus)**

Returns an instance of [IEncodingStatus](#).

**14.2.2.8 HRESULT flxengine_com::IFlix::errno_ ([out] FE2_errno * *flxerrno*, [out] on2s32 *
syserrno)**

Wrapper for [Flix2_Errno\(\)](#).

14.2.2.9 HRESULT flxengine_com::IFlix::flxerrno ([out, retval] FE2_errno * *flxerrno*)

Wrapper for [Flix2_Errno\(\)](#). Returns the flxerrno portion of [Flix2_Errno\(\)](#) should the language not handle reference parameters that well, e.g., PHP4

**14.2.2.10 HRESULT flxengine_com::IFlix::getEncoderState ([out, retval] FE2_EncState *
pEncState)**

Wrapper for [Flix2_GetEncoderState\(\)](#).

**14.2.2.11 HRESULT flxengine_com::IFlix::getExportAudio ([out, retval] on2bool *
bpExportAudio)**

Wrapper for [Flix2_GetExportAudio\(\)](#).

**14.2.2.12 HRESULT flxengine_com::IFlix::getExportVideo ([out, retval] on2bool *
bpExportVideo)**

Wrapper for [Flix2_GetExportVideo\(\)](#).

**14.2.2.13 HRESULT flxengine_com::IFlix::getExportVideoType ([out, retval]
FE2_ExportedVideoType * *pExportVideoType*)**

Wrapper for [Flix2_GetExportVideoType\(\)](#).

14.2.2.14 HRESULT flxengine_com::IFlix::getLogLevel ([out, retval] on2s32 * *level*)

Wrapper for [Flix2_GetLogLevel\(\)](#).

**14.2.2.15 HRESULT flxengine_com::IFlix::getOverwriteExistingFiles ([out, retval] on2bool *
bpOverwriteExistingFiles)**

Wrapper for [Flix2_GetOverwriteExistingFiles\(\)](#).

14.2.2.16 HRESULT flxengine_com::IFlix::getSourceDuration ([out, retval] on2s32 * *duration*)

Wrapper for [Flix2_GetSourceDuration\(\)](#).

14.2.2.17 HRESULT flxengine_com::IFlix::isEncoderRunning ([out, retval] on2bool * *bpIsEncoderRunning*)

Wrapper for [Flix2_IsEncoderRunning\(\)](#).

14.2.2.18 HRESULT flxengine_com::IFlix::reset ()

Wrapper for [Flix2_Reset\(\)](#).

14.2.2.19 HRESULT flxengine_com::IFlix::setExportAudio (const on2bool *bExportAudio*)

Wrapper for [Flix2_SetExportAudio\(\)](#).

14.2.2.20 HRESULT flxengine_com::IFlix::setExportVideo (const on2bool *bExportVideo*)

Wrapper for [Flix2_SetExportVideo\(\)](#).

14.2.2.21 HRESULT flxengine_com::IFlix::setExportVideoType (const FE2_ExportedVideoType *exportVideoType*)

Wrapper for [Flix2_SetExportVideoType\(\)](#).

14.2.2.22 HRESULT flxengine_com::IFlix::setInputFile (const BSTR *inputFile*)

Wrapper for [Flix2_SetInputFile\(\)](#).

14.2.2.23 HRESULT flxengine_com::IFlix::setLogLevel (on2s32 *level*)

Wrapper for [Flix2_SetLogLevel\(\)](#).

14.2.2.24 HRESULT flxengine_com::IFlix::setLogPath (BSTR *logpath*)

Wrapper for [Flix2_SetLogPath\(\)](#).

14.2.2.25 HRESULT flxengine_com::IFlix::setOutputFile (const BSTR *outputFile*)

Wrapper for [Flix2_SetOutputFile\(\)](#).

14.2.2.26 HRESULT flxengine_com::IFlix::setOverwriteExistingFiles (const on2bool *bOverwriteExistingFiles*)

Wrapper for [Flix2_SetOverwriteExistingFiles\(\)](#).

14.2.2.27 HRESULT flxengine_com::IFlix::stopEncoding ()

Wrapper for [Flix2_StopEncoding\(\)](#).

14.2.2.28 HRESULT flxengine_com::IFlix::swfOptions ([out, retval] ISwfOptions ** *swfoptions*)

Returns an instance of [ISwfOptions](#).

14.2.2.29 HRESULT flxengine_com::IFlix::syserrno ([out, retval] on2s32 * *syserrno*)

Wrapper for [Flix2_Errno\(\)](#). Returns the syserrno portion of [Flix2_Errno\(\)](#) should the language not handle reference parameters that well, e.g., PHP4

14.2.2.30 HRESULT flxengine_com::IFlix::validate ()

Wrapper for [Flix2_Validate\(\)](#).

14.2.2.31 HRESULT flxengine_com::IFlix::version ([out, retval] BSTR * *v*)

Wrapper for [Flix2_Version\(\)](#).

14.2.2.32 HRESULT flxengine_com::IFlix::videoOptions ([out, retval] IVideoOptions ** *vidoptions*)

Returns an instance of [IVideoOptions](#).

14.2.3 Property Documentation**14.2.3.1 BSTR flxengine_com::IFlix::FE2_3G2_FASTSTART [get]**

[FE2_3G2_FASTSTART](#)

14.2.3.2 BSTR flxengine_com::IFlix::FE2_3GP_FASTSTART [get]

[FE2_3GP_FASTSTART](#)

14.2.3.3 BSTR flxengine_com::IFlix::FE2_AAC_BITRATE [get]

[FE2_AAC_BITRATE](#)

14.2.3.4 BSTR flxengine_com::IFlix::FE2_AACPLUS_BITRATE [get]

[FE2_AACPLUS_BITRATE](#)

14.2.3.5 BSTR flxengine_com::IFlix::FE2_AACPLUS_PARAMETRIC_STEREO [get]

[FE2_AACPLUS_PARAMETRIC_STEREO](#)

14.2.3.6 BSTR flxengine_com::IFlix::FE2_ADAPTIVE_DEINTERLACE_MODE [get]

[FE2_ADAPTIVE_DEINTERLACE_MODE](#)

14.2.3.7 BSTR flxengine_com::IFlix::FE2_AMR_BITRATE [get]

[FE2_AMR_BITRATE](#)

14.2.3.8 BSTR flxengine_com::IFlix::FE2_BCHS_BRIGHTNESS [get]

[FE2_BCHS_BRIGHTNESS](#)

14.2.3.9 BSTR flxengine_com::IFlix::FE2_BCHS_CONTRAST [get]

[FE2_BCHS_CONTRAST](#)

14.2.3.10 BSTR flxengine_com::IFlix::FE2_BCHS_HUE [get]

[FE2_BCHS_HUE](#)

14.2.3.11 BSTR flxengine_com::IFlix::FE2_BCHS_SATURATION [get]

[FE2_BCHS_SATURATION](#)

14.2.3.12 BSTR flxengine_com::IFlix::FE2_BLUR_FILTER [get]

[FE2_BLUR_FILTER](#)

14.2.3.13 BSTR flxengine_com::IFlix::FE2_BLUR_MASKSIZE [get]

[FE2_BLUR_MASKSIZE](#)

14.2.3.14 BSTR flxengine_com::IFlix::FE2_CODEC_AAC [get]

[FE2_CODEC_AAC](#)

14.2.3.15 BSTR flxengine_com::IFlix::FE2_CODEC_AACPLUS [get]

[FE2_CODEC_AACPLUS](#)

14.2.3.16 BSTR flxengine_com::IFlix::FE2_CODEC_AMR_NB [get]

[FE2_CODEC_AMR_NB](#)

14.2.3.17 BSTR flxengine_com::IFlix::FE2_CODEC_H263 [get]

[FE2_CODEC_H263](#)

14.2.3.18 BSTR flxengine_com::IFlix::FE2_CODEC_H263_BASELINE [get]

[FE2_CODEC_H263_BASELINE](#)

14.2.3.19 BSTR flxengine_com::IFlix::FE2_CODEC_H264 [get]

[FE2_CODEC_H264](#)

14.2.3.20 BSTR flxengine_com::IFlix::FE2_CODEC_LAME [get]

[FE2_CODEC_LAME](#)

14.2.3.21 BSTR flxengine_com::IFlix::FE2_CODEC_PCM [get]

[FE2_CODEC_PCM](#)

14.2.3.22 BSTR flxengine_com::IFlix::FE2_CODEC_VORBIS [get]

[FE2_CODEC_VORBIS](#)

14.2.3.23 BSTR flxengine_com::IFlix::FE2_CODEC_VP6 [get]

[FE2_CODEC_VP6](#)

14.2.3.24 BSTR flxengine_com::IFlix::FE2_CODEC_VP6ALPHA [get]

[FE2_CODEC_VP6ALPHA](#)

14.2.3.25 BSTR flxengine_com::IFlix::FE2_CODEC_VP8 [get]

[FE2_CODEC_VP8](#)

14.2.3.26 BSTR flxengine_com::IFlix::FE2_CODECPARAM_BITRATE [get]

[FE2_CODECPARAM_BITRATE](#)

14.2.3.27 BSTR flxengine_com::IFlix::FE2_CROP_BOTTOM [get]

[FE2_CROP_BOTTOM](#)

14.2.3.28 BSTR flxengine_com::IFlix::FE2_CROP_LEFT [get]

[FE2_CROP_LEFT](#)

14.2.3.29 BSTR flxengine_com::IFlix::FE2_CROP_RIGHT [get]

[FE2_CROP_RIGHT](#)

14.2.3.30 BSTR flxengine_com::IFlix::FE2_CROP_TOP [get]

[FE2_CROP_TOP](#)

14.2.3.31 BSTR flxengine_com::IFlix::FE2_CUT_START_SEC [get]

[FE2_CUT_START_SEC](#)

14.2.3.32 BSTR flxengine_com::IFlix::FE2_CUT_STOP_SEC [get]

[FE2_CUT_STOP_SEC](#)

14.2.3.33 BSTR flxengine_com::IFlix::FE2_CUT_USE_SEEK [get]

[FE2_CUT_USE_SEEK](#)

14.2.3.34 BSTR flxengine_com::IFlix::FE2_DENOISE_NOISE_LEVEL [get]

[FE2_DENOISE_NOISE_LEVEL](#)

14.2.3.35 BSTR flxengine_com::IFlix::FE2_FILTER_ADAPTIVE_DEINTERLACE [get]

[FE2_FILTER_ADAPTIVE_DEINTERLACE](#)

14.2.3.36 BSTR flxengine_com::IFlix::FE2_FILTER_BCHS [get]

[FE2_FILTER_BCHS](#)

14.2.3.37 BSTR flxengine_com::IFlix::FE2_FILTER_BLUR [get]

[FE2_FILTER_BLUR](#)

14.2.3.38 BSTR flxengine_com::IFlix::FE2_FILTER_CROP [get]

[FE2_FILTER_CROP](#)

14.2.3.39 BSTR flxengine_com::IFlix::FE2_FILTER_CUT [get]

[FE2_FILTER_CUT](#)

14.2.3.40 BSTR flxengine_com::IFlix::FE2_FILTER_DENOISE [get]

[FE2_FILTER_DENOISE](#)

14.2.3.41 BSTR flxengine_com::IFlix::FE2_FILTER_FRAMERATE [get]

[FE2_FILTER_FRAMERATE](#)

14.2.3.42 BSTR flxengine_com::IFlix::FE2_FILTER_HIGHPASS [get]

[FE2_FILTER_HIGHPASS](#)

14.2.3.43 BSTR flxengine_com::IFlix::FE2_FILTER_LOWPASS [get]

[FE2_FILTER_LOWPASS](#)

14.2.3.44 BSTR flxengine_com::IFlix::FE2_FILTER_MIRROR [get]

[FE2_FILTER_MIRROR](#)

14.2.3.45 BSTR flxengine_com::IFlix::FE2_FILTER_OVERLAY [get]

[FE2_FILTER_OVERLAY](#)

14.2.3.46 BSTR flxengine_com::IFlix::FE2_FILTER_PNGEX [get]

[FE2_FILTER_PNGEX](#)

14.2.3.47 BSTR flxengine_com::IFlix::FE2_FILTER_RESAMPLE [get]

[FE2_FILTER_RESAMPLE](#)

14.2.3.48 BSTR flxengine_com::IFlix::FE2_FILTER_ROTATE [get]

[FE2_FILTER_ROTATE](#)

14.2.3.49 BSTR flxengine_com::IFlix::FE2_FILTER_SCALE [get]

[FE2_FILTER_SCALE](#)

14.2.3.50 BSTR flxengine_com::IFlix::FE2_FILTER_SHARPEN [get]

[FE2_FILTER_SHARPEN](#)

14.2.3.51 BSTR flxengine_com::IFlix::FE2_FLV_CUEPT_EVENT [get]

[FE2_FLV_CUEPT_EVENT](#)

14.2.3.52 BSTR flxengine_com::IFlix::FE2_FLV_CUEPT_NAV [get]

[FE2_FLV_CUEPT_NAV](#)

14.2.3.53 BSTR flxengine_com::IFlix::FE2_FLV_CUEPT_PARAM [get]

[FE2_FLV_CUEPT_PARAM](#)

14.2.3.54 BSTR flxengine_com::IFlix::FE2_FLV_METADATA_DISABLE [get]

[FE2_FLV_METADATA_DISABLE](#)

14.2.3.55 BSTR flxengine_com::IFlix::FE2_FLV_METADATA_ENABLE [get]

[FE2_FLV_METADATA_ENABLE](#)

14.2.3.56 BSTR flxengine_com::IFlix::FE2_FRAMERATE_DECIMATE [get]

[FE2_FRAMERATE_DECIMATE](#)

14.2.3.57 BSTR flxengine_com::IFlix::FE2_FRAMERATE_FPS [get]

[FE2_FRAMERATE_FPS](#)

14.2.3.58 BSTR flxengine_com::IFlix::FE2_FXM_CUEPT_EVENT [get]

[FE2_FXM_CUEPT_EVENT](#)

14.2.3.59 BSTR flxengine_com::IFlix::FE2_FXM_CUEPT_NAV [get]

[FE2_FXM_CUEPT_NAV](#)

14.2.3.60 BSTR flxengine_com::IFlix::FE2_FXM_CUEPT_PARAM [get]

[FE2_FXM_CUEPT_PARAM](#)

14.2.3.61 BSTR flxengine_com::IFlix::FE2_FXM_METADATA_DISABLE [get]

[FE2_FXM_METADATA_DISABLE](#)

14.2.3.62 BSTR flxengine_com::IFlix::FE2_FXM_METADATA_ENABLE [get]

[FE2_FXM_METADATA_ENABLE](#)

14.2.3.63 BSTR flxengine_com::IFlix::FE2_H263_BITRATE [get]

[FE2_H263_BITRATE](#)

14.2.3.64 BSTR flxengine_com::IFlix::FE2_H263_KFFREQ [get]

[FE2_H263_KFFREQ](#)

14.2.3.65 BSTR flxengine_com::IFlix::FE2_H263_KFINTTYPE [get]

[FE2_H263_KFINTTYPE](#)

14.2.3.66 BSTR flxengine_com::IFlix::FE2_H263_MAX_Q [get]

[FE2_H263_MAX_Q](#)

14.2.3.67 BSTR flxengine_com::IFlix::FE2_H263_MIN_Q [get]

[FE2_H263_MIN_Q](#)

14.2.3.68 BSTR flxengine_com::IFlix::FE2_H263_RC_MODE [get]

[FE2_H263_RC_MODE](#)

14.2.3.69 BSTR flxengine_com::IFlix::FE2_H264_B_FRAME_RATE [get]

[FE2_H264_B_FRAME_RATE](#)

14.2.3.70 BSTR flxengine_com::IFlix::FE2_H264_BITRATE [get]

[FE2_H264_BITRATE](#)

14.2.3.71 BSTR flxengine_com::IFlix::FE2_H264_KFFREQ [get]

[FE2_H264_KFFREQ](#)

14.2.3.72 BSTR flxengine_com::IFlix::FE2_H264_KFINTTYPE [get]

[FE2_H264_KFINTTYPE](#)

14.2.3.73 BSTR flxengine_com::IFlix::FE2_H264_PROFILE [get]

[FE2_H264_PROFILE](#)

14.2.3.74 BSTR flxengine_com::IFlix::FE2_H264_RC_MODE [get]

[FE2_H264_RC_MODE](#)

14.2.3.75 BSTR flxengine_com::IFlix::FE2_H264_SPEED [get]

[FE2_H264_SPEED](#)

14.2.3.76 BSTR flxengine_com::IFlix::FE2_HIGHPASS_CUTOFF [get]

[FE2_HIGHPASS_CUTOFF](#)

14.2.3.77 BSTR flxengine_com::IFlix::FE2_HIGHPASS_Q [get]

[FE2_HIGHPASS_Q](#)

14.2.3.78 BSTR flxengine_com::IFlix::FE2_ISOMEDIA_FASTSTART [get]

[FE2_ISOMEDIA_FASTSTART](#)

14.2.3.79 BSTR flxengine_com::IFlix::FE2_LAME_BITRATE [get]

[FE2_LAME_BITRATE](#)

14.2.3.80 BSTR flxengine_com::IFlix::FE2_LAME_CHANNELS [get]

[FE2_LAME_CHANNELS](#)

14.2.3.81 BSTR flxengine_com::IFlix::FE2_LAME_QUALITY [get]

[FE2_LAME_QUALITY](#)

14.2.3.82 BSTR flxengine_com::IFlix::FE2_LAME_RC_MODE [get]

[FE2_LAME_RC_MODE](#)

14.2.3.83 BSTR flxengine_com::IFlix::FE2_LOWPASS_CUTOFF [get]

[FE2_LOWPASS_CUTOFF](#)

14.2.3.84 BSTR flxengine_com::IFlix::FE2_LOWPASS_Q [get]

[FE2_LOWPASS_Q](#)

14.2.3.85 BSTR flxengine_com::IFlix::FE2_MIRROR_HORIZONTAL [get]

[FE2_MIRROR_HORIZONTAL](#)

14.2.3.86 BSTR flxengine_com::IFlix::FE2_MIRROR_VERTICAL [get]

[FE2_MIRROR_VERTICAL](#)

14.2.3.87 BSTR flxengine_com::IFlix::FE2_MOV_FASTSTART [get]

[FE2_MOV_FASTSTART](#)

14.2.3.88 BSTR flxengine_com::IFlix::FE2_MP4_FASTSTART [get]

[FE2_MP4_FASTSTART](#)

14.2.3.89 BSTR flxengine_com::IFlix::FE2_MUXER_3G2 [get]

[FE2_MUXER_3G2](#)

14.2.3.90 BSTR flxengine_com::IFlix::FE2_MUXER_3GP [get]

[FE2_MUXER_3GP](#)

14.2.3.91 BSTR flxengine_com::IFlix::FE2_MUXER_FLV [get]

[FE2_MUXER_FLV](#)

14.2.3.92 BSTR flxengine_com::IFlix::FE2_MUXER_FXM [get]

[FE2_MUXER_FXM](#)

14.2.3.93 BSTR flxengine_com::IFlix::FE2_MUXER_MOV [get]

[FE2_MUXER_MOV](#)

14.2.3.94 BSTR flxengine_com::IFlix::FE2_MUXER_MP4 [get]

[FE2_MUXER_MP4](#)

14.2.3.95 BSTR flxengine_com::IFlix::FE2_MUXER_SWF [get]

[FE2_MUXER_SWF](#)

14.2.3.96 BSTR flxengine_com::IFlix::FE2_MUXER_WEBM [get]

[FE2_MUXER_WEBM](#)

14.2.3.97 BSTR flxengine_com::IFlix::FE2_OVERLAY_FILE [get]

[FE2_OVERLAY_FILE](#)

14.2.3.98 BSTR flxengine_com::IFlix::FE2_OVERLAY_MASK_B [get]

[FE2_OVERLAY_MASK_B](#)

14.2.3.99 BSTR flxengine_com::IFlix::FE2_OVERLAY_MASK_G [get]

[FE2_OVERLAY_MASK_G](#)

14.2.3.100 BSTR flxengine_com::IFlix::FE2_OVERLAY_MASK_R [get]

[FE2_OVERLAY_MASK_R](#)

14.2.3.101 BSTR flxengine_com::IFlix::FE2_OVERLAY_MASK_RGB [get]

[FE2_OVERLAY_MASK_RGB](#)

14.2.3.102 BSTR flxengine_com::IFlix::FE2_OVERLAY_MASK_X [get]

[FE2_OVERLAY_MASK_X](#)

14.2.3.103 BSTR flxengine_com::IFlix::FE2_OVERLAY_MASK_XY [get]

[FE2_OVERLAY_MASK_XY](#)

14.2.3.104 BSTR flxengine_com::IFlix::FE2_OVERLAY_MASK_Y [get]

[FE2_OVERLAY_MASK_Y](#)

14.2.3.105 BSTR flxengine_com::IFlix::FE2_OVERLAY_POS [get]

[FE2_OVERLAY_POS](#)

14.2.3.106 BSTR flixengine_com::IFlix::FE2_OVERLAY_POS_X [get]

[FE2_OVERLAY_POS_X](#)

14.2.3.107 BSTR flixengine_com::IFlix::FE2_OVERLAY_POS_Y [get]

[FE2_OVERLAY_POS_Y](#)

14.2.3.108 BSTR flixengine_com::IFlix::FE2_PNGEX_AUTO_EXPORT_COUNT [get]

[FE2_PNGEX_AUTO_EXPORT_COUNT](#)

14.2.3.109 BSTR flixengine_com::IFlix::FE2_PNGEX_AUTO_EXPORT_END_TIME [get]

[FE2_PNGEX_AUTO_EXPORT_END_TIME](#)

14.2.3.110 BSTR flixengine_com::IFlix::FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD [get]

[FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD](#)

14.2.3.111 BSTR flixengine_com::IFlix::FE2_PNGEX_AUTO_EXPORT_START_TIME [get]

[FE2_PNGEX_AUTO_EXPORT_START_TIME](#)

14.2.3.112 BSTR flixengine_com::IFlix::FE2_PNGEX_COMPRESSION_LEVEL [get]

[FE2_PNGEX_COMPRESSION_LEVEL](#)

14.2.3.113 BSTR flixengine_com::IFlix::FE2_PNGEX_ENABLE_ALPHA [get]

[FE2_PNGEX_ENABLE_ALPHA](#)

14.2.3.114 BSTR flixengine_com::IFlix::FE2_PNGEX_EXPORT_CUE_POINTS [get]

[FE2_PNGEX_EXPORT_CUE_POINTS](#)

14.2.3.115 BSTR flixengine_com::IFlix::FE2_PNGEX_EXPORT_FIRST_FRAME_PNG [get]

[FE2_PNGEX_EXPORT_FIRST_FRAME_PNG](#)

14.2.3.116 BSTR flixengine_com::IFlix::FE2_PNGEX_EXPORT_INTERVAL [get]

[FE2_PNGEX_EXPORT_INTERVAL](#)

14.2.3.117 BSTR flixengine_com::IFlix::FE2_PNGEX_EXPORT_TIME_STRING [get]

[FE2_PNGEX_EXPORT_TIME_STRING](#)

14.2.3.118 BSTR flixengine_com::IFlix::FE2_PNGEX_FILENAME_PREFIX [get]

[FE2_PNGEX_FILENAME_PREFIX](#)

14.2.3.119 BSTR flixengine_com::IFlix::FE2_PNGEX_FILENAME_SUFFIX [get]

[FE2_PNGEX_FILENAME_SUFFIX](#)

14.2.3.120 BSTR flixengine_com::IFlix::FE2_PNGEX_HEIGHT [get]

[FE2_PNGEX_HEIGHT](#)

14.2.3.121 BSTR flixengine_com::IFlix::FE2_PNGEX_OUTPUT_DIRECTORY [get]

[FE2_PNGEX_OUTPUT_DIRECTORY](#)

14.2.3.122 BSTR flixengine_com::IFlix::FE2_PNGEX_WIDTH [get]

[FE2_PNGEX_WIDTH](#)

14.2.3.123 BSTR flixengine_com::IFlix::FE2_RESAMPLE_CHANNELS [get]

[FE2_RESAMPLE_CHANNELS](#)

14.2.3.124 BSTR flixengine_com::IFlix::FE2_RESAMPLE_RATE [get]

[FE2_RESAMPLE_RATE](#)

14.2.3.125 BSTR flixengine_com::IFlix::FE2_ROTATE_ANGLE [get]

[FE2_ROTATE_ANGLE](#)

14.2.3.126 BSTR flixengine_com::IFlix::FE2_SCALE_HEIGHT [get]

[FE2_SCALE_HEIGHT](#)

14.2.3.127 BSTR flixengine_com::IFlix::FE2_SCALE_WIDTH [get]

[FE2_SCALE_WIDTH](#)

14.2.3.128 BSTR flxengine_com::IFlix::FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR [get]

[FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#)

14.2.3.129 BSTR flxengine_com::IFlix::FE2_SWF_ADD_VARIABLE [get]

[FE2_SWF_ADD_VARIABLE](#)

14.2.3.130 BSTR flxengine_com::IFlix::FE2_SWF_DELETE_VARIABLE [get]

[FE2_SWF_DELETE_VARIABLE](#)

14.2.3.131 BSTR flxengine_com::IFlix::FE2_SWF_EMBEDDED_URL [get]

[FE2_SWF_EMBEDDED_URL](#)

14.2.3.132 BSTR flxengine_com::IFlix::FE2_SWF_EMBEDDED_URL_TARGET [get]

[FE2_SWF_EMBEDDED_URL_TARGET](#)

14.2.3.133 BSTR flxengine_com::IFlix::FE2_SWF_EMBEDDED_URL_TYPE [get]

[FE2_SWF_EMBEDDED_URL_TYPE](#)

14.2.3.134 BSTR flxengine_com::IFlix::FE2_SWF_FIXED_PRELOAD_PCT [get]

[FE2_SWF_FIXED_PRELOAD_PCT](#)

14.2.3.135 BSTR flxengine_com::IFlix::FE2_SWF_FRAMERATE [get]

[FE2_SWF_FRAMERATE](#)

14.2.3.136 BSTR flxengine_com::IFlix::FE2_SWF_HEIGHT [get]

[FE2_SWF_HEIGHT](#)

14.2.3.137 BSTR flxengine_com::IFlix::FE2_SWF_LOOP_COUNT [get]

[FE2_SWF_LOOP_COUNT](#)

14.2.3.138 BSTR flxengine_com::IFlix::FE2_SWF_ON_END_OPTION [get]

[FE2_SWF_ON_END_OPTION](#)

14.2.3.139 BSTR flixengine_com::IFlix::FE2_SWF_ON_END_URL [get]

[FE2_SWF_ON_END_URL](#)

14.2.3.140 BSTR flixengine_com::IFlix::FE2_SWF_ON_START_OPTION [get]

[FE2_SWF_ON_START_OPTION](#)

14.2.3.141 BSTR flixengine_com::IFlix::FE2_SWF_PRELOAD_TYPE [get]

[FE2_SWF_PRELOAD_TYPE](#)

14.2.3.142 BSTR flixengine_com::IFlix::FE2_SWF_START_BLANK_FRAME [get]

[FE2_SWF_START_BLANK_FRAME](#)

14.2.3.143 BSTR flixengine_com::IFlix::FE2_SWF_START_WAIT_SEC [get]

[FE2_SWF_START_WAIT_SEC](#)

14.2.3.144 BSTR flixengine_com::IFlix::FE2_SWF_WIDTH [get]

[FE2_SWF_WIDTH](#)

14.2.3.145 BSTR flixengine_com::IFlix::FE2_VCODECPARAM_KFFREQ [get]

[FE2_VCODECPARAM_KFFREQ](#)

14.2.3.146 BSTR flixengine_com::IFlix::FE2_VCODECPARAM_KFINTTYPE [get]

[FE2_VCODECPARAM_KFINTTYPE](#)

14.2.3.147 BSTR flixengine_com::IFlix::FE2_VCODECPARAM_RC_MODE [get]

[FE2_VCODECPARAM_RC_MODE](#)

14.2.3.148 BSTR flixengine_com::IFlix::FE2_VORBIS_BITRATE [get]

[FE2_VORBIS_BITRATE](#)

14.2.3.149 BSTR flixengine_com::IFlix::FE2_VP6_2PASS_MAX_SECTION [get]

[FE2_VP6_2PASS_MAX_SECTION](#)

14.2.3.150 BSTR flxengine_com::IFlix::FE2_VP6_2PASS_MIN_SECTION [get]

[FE2_VP6_2PASS_MIN_SECTION](#)

14.2.3.151 BSTR flxengine_com::IFlix::FE2_VP6_BITRATE [get]

[FE2_VP6_BITRATE](#)

14.2.3.152 BSTR flxengine_com::IFlix::FE2_VP6_CONCURRENCY [get]

[FE2_VP6_CONCURRENCY](#)

14.2.3.153 BSTR flxengine_com::IFlix::FE2_VP6_CXMODE [get]

[FE2_VP6_CXMODE](#)

14.2.3.154 BSTR flxengine_com::IFlix::FE2_VP6_KFFREQ [get]

[FE2_VP6_KFFREQ](#)

14.2.3.155 BSTR flxengine_com::IFlix::FE2_VP6_KFINTTYPE [get]

[FE2_VP6_KFINTTYPE](#)

14.2.3.156 BSTR flxengine_com::IFlix::FE2_VP6_MAX_Q [get]

[FE2_VP6_MAX_Q](#)

14.2.3.157 BSTR flxengine_com::IFlix::FE2_VP6_MIN_Q [get]

[FE2_VP6_MIN_Q](#)

14.2.3.158 BSTR flxengine_com::IFlix::FE2_VP6_NOISE_REDUCTION [get]

[FE2_VP6_NOISE_REDUCTION](#)

14.2.3.159 BSTR flxengine_com::IFlix::FE2_VP6_PROFILE [get]

[FE2_VP6_PROFILE](#)

14.2.3.160 BSTR flxengine_com::IFlix::FE2_VP6_RC_MODE [get]

[FE2_VP6_RC_MODE](#)

14.2.3.161 BSTR flixengine_com::IFlix::FE2_VP6_SHARPNESS [get]

[FE2_VP6_SHARPNESS](#)

14.2.3.162 BSTR flixengine_com::IFlix::FE2_VP6_STREAM_MAX_BUFFER [get]

[FE2_VP6_STREAM_MAX_BUFFER](#)

14.2.3.163 BSTR flixengine_com::IFlix::FE2_VP6_STREAM_OPTIMAL_BUFFER [get]

[FE2_VP6_STREAM_OPTIMAL_BUFFER](#)

14.2.3.164 BSTR flixengine_com::IFlix::FE2_VP6_STREAM_PEAK_BITRATE [get]

[FE2_VP6_STREAM_PEAK_BITRATE](#)

14.2.3.165 BSTR flixengine_com::IFlix::FE2_VP6_STREAM_PREBUFFER [get]

[FE2_VP6_STREAM_PREBUFFER](#)

14.2.3.166 BSTR flixengine_com::IFlix::FE2_VP6_TEMPORAL_DOWN_WATERMARK
[get]

[FE2_VP6_TEMPORAL_DOWN_WATERMARK](#)

14.2.3.167 BSTR flixengine_com::IFlix::FE2_VP6_TEMPORAL_RESAMPLING [get]

[FE2_VP6_TEMPORAL_RESAMPLING](#)

14.2.3.168 BSTR flixengine_com::IFlix::FE2_VP6_UNDERSHOOT_PCT [get]

[FE2_VP6_UNDERSHOOT_PCT](#)

14.2.3.169 BSTR flixengine_com::IFlix::FE2_VP6A_2PASS_MAX_SECTION [get]

[FE2_VP6A_2PASS_MAX_SECTION](#)

14.2.3.170 BSTR flixengine_com::IFlix::FE2_VP6A_2PASS_MIN_SECTION [get]

[FE2_VP6A_2PASS_MIN_SECTION](#)

14.2.3.171 BSTR flixengine_com::IFlix::FE2_VP6A_ALPHA_BITRATE [get]

[FE2_VP6A_ALPHA_BITRATE](#)

14.2.3.172 BSTR flxengine_com::IFlix::FE2_VP6A_ALPHA_MAX_Q [get]

[FE2_VP6A_ALPHA_MAX_Q](#)

14.2.3.173 BSTR flxengine_com::IFlix::FE2_VP6A_ALPHA_MIN_Q [get]

[FE2_VP6A_ALPHA_MIN_Q](#)

14.2.3.174 BSTR flxengine_com::IFlix::FE2_VP6A_ALPHA_NOISE_REDUCTION [get]

[FE2_VP6A_ALPHA_NOISE_REDUCTION](#)

14.2.3.175 BSTR flxengine_com::IFlix::FE2_VP6A_ALPHA_SHARPNESS [get]

[FE2_VP6A_ALPHA_SHARPNESS](#)

14.2.3.176 BSTR flxengine_com::IFlix::FE2_VP6A_BITRATE [get]

[FE2_VP6A_BITRATE](#)

14.2.3.177 BSTR flxengine_com::IFlix::FE2_VP6A_CXMODE [get]

[FE2_VP6A_CXMODE](#)

14.2.3.178 BSTR flxengine_com::IFlix::FE2_VP6A_KFFREQ [get]

[FE2_VP6A_KFFREQ](#)

14.2.3.179 BSTR flxengine_com::IFlix::FE2_VP6A_KFINTTYPE [get]

[FE2_VP6A_KFINTTYPE](#)

14.2.3.180 BSTR flxengine_com::IFlix::FE2_VP6A_MAX_Q [get]

[FE2_VP6A_MAX_Q](#)

14.2.3.181 BSTR flxengine_com::IFlix::FE2_VP6A_MIN_Q [get]

[FE2_VP6A_MIN_Q](#)

14.2.3.182 BSTR flxengine_com::IFlix::FE2_VP6A_NOISE_REDUCTION [get]

[FE2_VP6A_NOISE_REDUCTION](#)

14.2.3.183 BSTR flixengine_com::IFlix::FE2_VP6A_RC_MODE [get]

[FE2_VP6A_RC_MODE](#)

14.2.3.184 BSTR flixengine_com::IFlix::FE2_VP6A_SHARPNESS [get]

[FE2_VP6A_SHARPNESS](#)

14.2.3.185 BSTR flixengine_com::IFlix::FE2_VP6A_STREAM_MAX_BUFFER [get]

[FE2_VP6A_STREAM_MAX_BUFFER](#)

14.2.3.186 BSTR flixengine_com::IFlix::FE2_VP6A_STREAM_OPTIMAL_BUFFER [get]

[FE2_VP6A_STREAM_OPTIMAL_BUFFER](#)

14.2.3.187 BSTR flixengine_com::IFlix::FE2_VP6A_STREAM_PEAK_BITRATE [get]

[FE2_VP6A_STREAM_PEAK_BITRATE](#)

14.2.3.188 BSTR flixengine_com::IFlix::FE2_VP6A_STREAM_PREBUFFER [get]

[FE2_VP6A_STREAM_PREBUFFER](#)

14.2.3.189 BSTR flixengine_com::IFlix::FE2_VP6A_TEMPORAL_DOWN_WATERMARK
[get]

[FE2_VP6A_TEMPORAL_DOWN_WATERMARK](#)

14.2.3.190 BSTR flixengine_com::IFlix::FE2_VP6A_TEMPORAL_RESAMPLING [get]

[FE2_VP6A_TEMPORAL_RESAMPLING](#)

14.2.3.191 BSTR flixengine_com::IFlix::FE2_VP6A_UNDERSHOOT_PCT [get]

[FE2_VP6A_UNDERSHOOT_PCT](#)

14.2.3.192 BSTR flixengine_com::IFlix::FE2_VP8_2PASS_MAX_SECTION [get]

[FE2_VP8_2PASS_MAX_SECTION](#)

14.2.3.193 BSTR flixengine_com::IFlix::FE2_VP8_2PASS_MIN_SECTION [get]

[FE2_VP8_2PASS_MIN_SECTION](#)

14.2.3.194 BSTR flxengine_com::IFlix::FE2_VP8_ALTREF [get]

[FE2_VP8_ALTREF](#)

14.2.3.195 BSTR flxengine_com::IFlix::FE2_VP8_AR_MAX_FRAMES [get]

[FE2_VP8_AR_MAX_FRAMES](#)

14.2.3.196 BSTR flxengine_com::IFlix::FE2_VP8_AR_STRENGTH [get]

[FE2_VP8_AR_STRENGTH](#)

14.2.3.197 BSTR flxengine_com::IFlix::FE2_VP8_AR_TYPE [get]

[FE2_VP8_AR_TYPE](#)

14.2.3.198 BSTR flxengine_com::IFlix::FE2_VP8_BITRATE [get]

[FE2_VP8_BITRATE](#)

14.2.3.199 BSTR flxengine_com::IFlix::FE2_VP8_CXMODE [get]

[FE2_VP8_CXMODE](#)

14.2.3.200 BSTR flxengine_com::IFlix::FE2_VP8_DROP_THRESH [get]

[FE2_VP8_DROP_THRESH](#)

14.2.3.201 BSTR flxengine_com::IFlix::FE2_VP8_KFFREQ [get]

[FE2_VP8_KFFREQ](#)

14.2.3.202 BSTR flxengine_com::IFlix::FE2_VP8_KFINTTYPE [get]

[FE2_VP8_KFINTTYPE](#)

14.2.3.203 BSTR flxengine_com::IFlix::FE2_VP8_LAG [get]

[FE2_VP8_LAG](#)

14.2.3.204 BSTR flxengine_com::IFlix::FE2_VP8_MAX_Q [get]

[FE2_VP8_MAX_Q](#)

14.2.3.205 BSTR flixengine_com::IFlix::FE2_VP8_MB_STATIC_THRESHOLD [get]

[FE2_VP8_MB_STATIC_THRESHOLD](#)

14.2.3.206 BSTR flixengine_com::IFlix::FE2_VP8_MIN_Q [get]

[FE2_VP8_MIN_Q](#)

14.2.3.207 BSTR flixengine_com::IFlix::FE2_VP8_NOISE_REDUCTION [get]

[FE2_VP8_NOISE_REDUCTION](#)

14.2.3.208 BSTR flixengine_com::IFlix::FE2_VP8_OVERSHOOT_PCT [get]

[FE2_VP8_OVERSHOOT_PCT](#)

14.2.3.209 BSTR flixengine_com::IFlix::FE2_VP8_PROFILE [get]

[FE2_VP8_PROFILE](#)

14.2.3.210 BSTR flixengine_com::IFlix::FE2_VP8_RC_MODE [get]

[FE2_VP8_RC_MODE](#)

14.2.3.211 BSTR flixengine_com::IFlix::FE2_VP8_SHARPNESS [get]

[FE2_VP8_SHARPNESS](#)

14.2.3.212 BSTR flixengine_com::IFlix::FE2_VP8_STREAM_INITIAL_BUFFER [get]

[FE2_VP8_STREAM_INITIAL_BUFFER](#)

14.2.3.213 BSTR flixengine_com::IFlix::FE2_VP8_STREAM_MAX_BUFFER [get]

[FE2_VP8_STREAM_MAX_BUFFER](#)

14.2.3.214 BSTR flixengine_com::IFlix::FE2_VP8_STREAM_OPTIMAL_BUFFER [get]

[FE2_VP8_STREAM_OPTIMAL_BUFFER](#)

14.2.3.215 BSTR flixengine_com::IFlix::FE2_VP8_THREADS [get]

[FE2_VP8_THREADS](#)

14.2.3.216 BSTR flixengine_com::IFlix::FE2_VP8_TOKEN_PARTITIONS [get]

[FE2_VP8_TOKEN_PARTITIONS](#)

14.2.3.217 BSTR flixengine_com::IFlix::FE2_VP8_UNDERSHOOT_PCT [get]

[FE2_VP8_UNDERSHOOT_PCT](#)

14.2.3.218 on2sc flixengine_com::IFlix::sc [get]

[on2sc](#) from the last Flix Engine function called within this interface

The documentation for this interface was generated from the following file:

- [flixengine_com.idl](#)

14.3 flixengine_com::IFlixPlgn Interface Reference

Interface for accessing [Codec](#), [Filter](#) and [Muxer](#) functions.

```
import "flixengine_com.idl";
```

Public Member Functions

- HRESULT [remove](#) ()
Wrapper for [Flix2_RemoveCodec\(\)](#) / [Flix2_RemoveFilter\(\)](#) / [Flix2_RemoveMuxer\(\)](#).
- HRESULT [setParamAsStr](#) (const BSTR name, const BSTR value)
Wrapper for [Flix2_CodecSetParamAsStr\(\)](#) / [Flix2_FilterSetParamAsStr\(\)](#) / [Flix2_MuxerSetParamAsStr\(\)](#).
- HRESULT [setParam](#) (const BSTR name, double inDbIVal)
Wrapper for [Flix2_CodecSetParam\(\)](#) / [Flix2_FilterSetParam\(\)](#) / [Flix2_MuxerSetParam\(\)](#).
- HRESULT [getParam](#) (const BSTR name,[out, retval] double *outDbIVal)
Wrapper for [Flix2_CodecGetParam\(\)](#) / [Flix2_FilterGetParam\(\)](#) / [Flix2_MuxerGetParam\(\)](#).

Properties

- [on2sc](#) [sc](#) [get]
[on2sc](#) from the last Flix Engine function called within this interface

14.3.1 Detailed Description

Interface for accessing [Codec](#), [Filter](#) and [Muxer](#) functions. **HIDDEN**

Definition at line 114 of file flixengine_com.idl.

14.3.2 Member Function Documentation

14.3.2.1 HRESULT flixengine_com::IFlixPlgn::getParam (const BSTR *name*, [out, retval] double * *outDbIVal*)

Wrapper for [Flix2_CodecGetParam\(\)](#) / [Flix2_FilterGetParam\(\)](#) / [Flix2_MuxerGetParam\(\)](#).

Note:

name is converted to a char

14.3.2.2 HRESULT flixengine_com::IFlixPlgn::remove ()

Wrapper for [Flix2_RemoveCodec\(\)](#) / [Flix2_RemoveFilter\(\)](#) / [Flix2_RemoveMuxer\(\)](#).

14.3.2.3 HRESULT flxengine_com::IFlixPlgn::setParam (const BSTR *name*, double *inDblVal*)

Wrapper for [Flix2_CodecSetParam\(\)](#) / [Flix2_FilterSetParam\(\)](#) / [Flix2_MuxerSetParam\(\)](#).

Note:

name is converted to a char

14.3.2.4 HRESULT flxengine_com::IFlixPlgn::setParamAsStr (const BSTR *name*, const BSTR *value*)

Wrapper for [Flix2_CodecSetParamAsStr\(\)](#) / [Flix2_FilterSetParamAsStr\(\)](#) / [Flix2_MuxerSetParamAsStr\(\)](#).

Note:

name is converted to a char

14.3.3 Property Documentation**14.3.3.1 on2sc flxengine_com::IFlixPlgn::sc [get]**

[on2sc](#) from the last Flix Engine function called within this interface

The documentation for this interface was generated from the following file:

- [flxengine_com.idl](#)

14.4 flxengine_com::ISwfOptions Interface Reference

Interface for accessing [SWF creation](#) functions.

```
import "flxengine_com.idl";
```

Public Member Functions

- HRESULT [reset](#) ()
Wrapper for [swf_options_Reset\(\)](#).
- HRESULT [setEmbeddedUrl](#) (const BSTR embeddedUrl)
Wrapper for [swf_options_SetEmbeddedUrl\(\)](#).
- HRESULT [setEmbeddedUrlTarget](#) (const BSTR embeddedUrlTarget)
Wrapper for [swf_options_SetEmbeddedUrlTarget\(\)](#).
- HRESULT [getEmbeddedUrlType](#) ([out, retval] [FE2_EmbeddedUrlType](#) *embeddedUrlType)
Wrapper for [swf_options_GetEmbeddedUrlType\(\)](#).
- HRESULT [setEmbeddedUrlType](#) (const [FE2_EmbeddedUrlType](#) embeddedUrlType)
Wrapper for [swf_options_SetEmbeddedUrlType\(\)](#).
- HRESULT [getInsertBlankFrameOnStart](#) ([out, retval] [on2bool](#) *pInsertBlankFrameOnStart)
Wrapper for [swf_options_GetInsertBlankFrameOnStart\(\)](#).
- HRESULT [setInsertBlankFrameOnStart](#) (const [on2bool](#) insertBlankFrameOnStart)
Wrapper for [swf_options_SetInsertBlankFrameOnStart\(\)](#).
- HRESULT [getSwfFramerateAsDouble](#) ([out, retval] double *pSwfFramerate)
Wrapper for [swf_options_GetSwfFramerateAsDouble\(\)](#).
- HRESULT [setSwfFramerateAsDouble](#) (const double swfFramerate)
Wrapper for [swf_options_SetSwfFramerateAsDouble\(\)](#).
- HRESULT [getEnablePreloader](#) ([out, retval] [on2bool](#) *pEnablePreloader)
Wrapper for [swf_options_GetEnablePreloader\(\)](#).
- HRESULT [setEnablePreloader](#) (const [on2bool](#) enablePreloader)
Wrapper for [swf_options_SetEnablePreloader\(\)](#).
- HRESULT [getPercentToPreload](#) ([out, retval] [on2s32](#) *pPercentToPreload)
Wrapper for [swf_options_GetPercentToPreload\(\)](#).
- HRESULT [setPercentToPreload](#) (const [on2s32](#) percentToPreload)
Wrapper for [swf_options_SetPercentToPreload\(\)](#).
- HRESULT [getPreloaderType](#) ([out, retval] [FE2_SwfPreloaderOptions](#) *pPreloaderType)
Wrapper for [swf_options_GetPreloaderType\(\)](#).

- HRESULT [setPreloaderType](#) (const [FE2_SwfPreloaderOptions](#) preloaderType)
Wrapper for [swf_options_SetPreloaderType\(\)](#).
- HRESULT [getAdaptivePreloaderBufferFactor](#) ([out, retval] double *pPreloaderBufferFactor)
Wrapper for [swf_options_GetAdaptivePreloaderBufferFactor\(\)](#).
- HRESULT [setAdaptivePreloaderBufferFactor](#) (const double preloaderBufferFactor)
Wrapper for [swf_options_SetAdaptivePreloaderBufferFactor\(\)](#).
- HRESULT [getMovieOnEndOptions](#) ([out, retval] [FE2_SwfOnEndOptions](#) *pOnEndOptions)
Wrapper for [swf_options_GetMovieOnEndOptions\(\)](#).
- HRESULT [setMovieOnEndOptions](#) (const [FE2_SwfOnEndOptions](#) onEndOptions)
Wrapper for [swf_options_SetMovieOnEndOptions\(\)](#).
- HRESULT [getLoopCount](#) ([out, retval] [on2s32](#) *pLoopCount)
Wrapper for [swf_options_GetLoopCount\(\)](#).
- HRESULT [setLoopCount](#) (const [on2s32](#) loopCount)
Wrapper for [swf_options_SetLoopCount\(\)](#).
- HRESULT [setLoadMovieOnEndUrl](#) (const BSTR loadMovieOnEndUrl)
Wrapper for [swf_options_SetLoadMovieOnEndUrl\(\)](#).
- HRESULT [getMovieOnStartOptions](#) ([out, retval] [FE2_SwfOnStartOptions](#) *pOnStartOptions)
Wrapper for [swf_options_GetMovieOnStartOptions\(\)](#).
- HRESULT [setMovieOnStartOptions](#) (const [FE2_SwfOnStartOptions](#) onStartOptions)
Wrapper for [swf_options_SetMovieOnStartOptions\(\)](#).
- HRESULT [getWaitTimeToStart](#) ([out, retval] [on2s32](#) *pWaitTimeToStart)
Wrapper for [swf_options_GetWaitTimeToStart\(\)](#).
- HRESULT [setWaitTimeToStart](#) (const [on2s32](#) waitTimeToStart)
Wrapper for [swf_options_SetWaitTimeToStart\(\)](#).
- HRESULT [addVariable](#) (const BSTR name, const BSTR value)
Wrapper for [swf_options_AddVariable\(\)](#).
- HRESULT [deleteVariable](#) (const [on2s32](#) index)
Wrapper for [swf_options_DeleteVariable\(\)](#).
- HRESULT [getVariableCount](#) ([out, retval] [on2s32](#) *pVariableCount)
Wrapper for [swf_options_GetVariableCount\(\)](#).
- HRESULT [updateVariable](#) (const [on2s32](#) index, const BSTR name, const BSTR value)
Wrapper for [swf_options_UpdateVariable\(\)](#).

Properties

- `on2sc sc` [get]
on2sc from the last Flix Engine function called within this interface

14.4.1 Detailed Description

Interface for accessing [SWF creation](#) functions.

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with the [SWF](#) muxer. This interface will be removed in a future release.

Definition at line 211 of file `flixengine_com.idl`.

14.4.2 Member Function Documentation

14.4.2.1 HRESULT flixengine_com::ISwfOptions::addVariable (const BSTR *name*, const BSTR *value*)

Wrapper for [swf_options_AddVariable\(\)](#).

Note:

name is converted to a `char*` *value* is converted to a `char*`

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-ADD_VARIABLE](#) parameter.

14.4.2.2 HRESULT flixengine_com::ISwfOptions::deleteVariable (const on2s32 *index*)

Wrapper for [swf_options_DeleteVariable\(\)](#).

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-DELETE_VARIABLE](#) parameter.

14.4.2.3 HRESULT flixengine_com::ISwfOptions::getAdaptivePreloaderBufferFactor ([out, retval] double * *pPreloaderBufferFactor*)

Wrapper for [swf_options_GetAdaptivePreloaderBufferFactor\(\)](#).

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter.

14.4.2.4 HRESULT flxengine_com::ISwfOptions::getEmbeddedUrlType ([out, retval] FE2_EmbeddedUrlType * *embeddedUrlType*)

Wrapper for `swf_options_GetEmbeddedUrlType()`.

Deprecated

Use the `flxengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_EMBEDDED_URL_TYPE` parameter.

14.4.2.5 HRESULT flxengine_com::ISwfOptions::getEnablePreloader ([out, retval] on2bool * *pEnablePreloader*)

Wrapper for `swf_options_GetEnablePreloader()`.

Deprecated

Use the `flxengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_PRELOAD_TYPE` parameter.

14.4.2.6 HRESULT flxengine_com::ISwfOptions::getInsertBlankFrameOnStart ([out, retval] on2bool * *pInsertBlankFrameOnStart*)

Wrapper for `swf_options_GetInsertBlankFrameOnStart()`.

Deprecated

Use the `flxengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_START_BLANK_FRAME` parameter.

14.4.2.7 HRESULT flxengine_com::ISwfOptions::getLoopCount ([out, retval] on2s32 * *pLoopCount*)

Wrapper for `swf_options_GetLoopCount()`.

Deprecated

Use the `flxengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_LOOP_COUNT` parameter.

14.4.2.8 HRESULT flxengine_com::ISwfOptions::getMovieOnEndOptions ([out, retval] FE2_SwfOnEndOptions * *pOnEndOptions*)

Wrapper for `swf_options_GetMovieOnEndOptions()`.

Deprecated

Use the `flxengine_com::IFlixPlgn` interface along with `FE2_MUXER_SWF` and the `FE2_SWF_ON_END_OPTION` parameter.

14.4.2.9 HRESULT flxengine_com::ISwfOptions::getMovieOnStartOptions ([out, retval] FE2_SwfOnStartOptions * *pOnStartOptions*)

Wrapper for [swf_options_GetMovieOnStartOptions\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_START_OPTION](#) parameter.

14.4.2.10 HRESULT flxengine_com::ISwfOptions::getPercentToPreload ([out, retval] on2s32 * *pPercentToPreload*)

Wrapper for [swf_options_GetPercentToPreload\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FIXED_PRELOAD_PCT](#) parameter.

14.4.2.11 HRESULT flxengine_com::ISwfOptions::getPreloaderType ([out, retval] FE2_SwfPreloaderOptions * *pPreloaderType*)

Wrapper for [swf_options_GetPreloaderType\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_PRELOAD_TYPE](#) parameter.

14.4.2.12 HRESULT flxengine_com::ISwfOptions::getSwfFramerateAsDouble ([out, retval] double * *pSwfFramerate*)

Wrapper for [swf_options_GetSwfFramerateAsDouble\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FRAMERATE](#) parameter.

14.4.2.13 HRESULT flxengine_com::ISwfOptions::getVariableCount ([out, retval] on2s32 * *pVariableCount*)

Wrapper for [swf_options_GetVariableCount\(\)](#).

Deprecated

Use the [SWF](#) muxer.

14.4.2.14 HRESULT flxengine_com::ISwfOptions::getWaitTimeToStart ([out, retval] on2s32 * *pWaitTimeToStart*)

Wrapper for [swf_options_GetWaitTimeToStart\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_WAIT_SEC](#) parameter.

14.4.2.15 HRESULT flxengine_com::ISwfOptions::reset ()

Wrapper for [swf_options_Reset\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with the [SWF](#) muxer. This interface will be removed in a future release.

14.4.2.16 HRESULT flxengine_com::ISwfOptions::setAdaptivePreloaderBufferFactor (const double *preloaderBufferFactor*)

Wrapper for [swf_options_SetAdaptivePreloaderBufferFactor\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#) parameter.

14.4.2.17 HRESULT flxengine_com::ISwfOptions::setEmbeddedUrl (const BSTR *embeddedUrl*)

Wrapper for [swf_options_SetEmbeddedUrl\(\)](#).

Note:

embeddedUrl is converted to a char*

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL](#) parameter.

14.4.2.18 HRESULT flxengine_com::ISwfOptions::setEmbeddedUrlTarget (const BSTR *embeddedUrlTarget*)

Wrapper for [swf_options_SetEmbeddedUrlTarget\(\)](#).

Note:

name is converted to a char*

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL](#) parameter.

14.4.2.19 HRESULT flixengine_com::ISwfOptions::setEmbeddedUrlType (const FE2_EmbeddedUrlType *embeddedUrlType*)

Wrapper for [swf_options_SetEmbeddedUrlType\(\)](#).

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_EMBEDDED_URL_TYPE](#) parameter.

14.4.2.20 HRESULT flixengine_com::ISwfOptions::setEnabledPreloader (const on2bool *enablePreloader*)

Wrapper for [swf_options_SetEnablePreloader\(\)](#).

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_PRELOAD_TYPE](#) parameter.

14.4.2.21 HRESULT flixengine_com::ISwfOptions::setInsertBlankFrameOnStart (const on2bool *insertBlankFrameOnStart*)

Wrapper for [swf_options_SetInsertBlankFrameOnStart\(\)](#).

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_BLANK_FRAME](#) parameter.

14.4.2.22 HRESULT flixengine_com::ISwfOptions::setLoadMovieOnEndUrl (const BSTR *loadMovieOnEndUrl*)

Wrapper for [swf_options_SetLoadMovieOnEndUrl\(\)](#).

Note:

name is converted to a char*

Deprecated

Use the [flixengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_END_URL](#) parameter.

14.4.2.23 HRESULT flxengine_com::ISwfOptions::setLoopCount (const on2s32 *loopCount*)

Wrapper for [swf_options_SetLoopCount\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-LOOP_COUNT](#) parameter.

14.4.2.24 HRESULT flxengine_com::ISwfOptions::setMovieOnEndOptions (const FE2_SwfOnEndOptions *onEndOptions*)

Wrapper for [swf_options_SetMovieOnEndOptions\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_-END_OPTION](#) parameter.

14.4.2.25 HRESULT flxengine_com::ISwfOptions::setMovieOnStartOptions (const FE2_SwfOnStartOptions *onStartOptions*)

Wrapper for [swf_options_SetMovieOnStartOptions\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ON_-START_OPTION](#) parameter.

14.4.2.26 HRESULT flxengine_com::ISwfOptions::setPercentToPreload (const on2s32 *percentToPreload*)

Wrapper for [swf_options_SetPercentToPreload\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-FIXED_PRELOAD_PCT](#) parameter.

14.4.2.27 HRESULT flxengine_com::ISwfOptions::setPreloaderType (const FE2_SwfPreloaderOptions *preloaderType*)

Wrapper for [swf_options_SetPreloaderType\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_-PRELOAD_TYPE](#) parameter.

14.4.2.28 HRESULT flxengine_com::ISwfOptions::setSwfFramerateAsDouble (const double *swfFramerate*)

Wrapper for [swf_options_SetSwfFramerateAsDouble\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_FRAMERATE](#) parameter.

14.4.2.29 HRESULT flxengine_com::ISwfOptions::setWaitTimeToStart (const on2s32 *waitTimeToStart*)

Wrapper for [swf_options_SetWaitTimeToStart\(\)](#).

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_START_WAIT_SEC](#) parameter.

14.4.2.30 HRESULT flxengine_com::ISwfOptions::updateVariable (const on2s32 *index*, const BSTR *name*, const BSTR *value*)

Wrapper for [swf_options_UpdateVariable\(\)](#).

Note:

name is converted to a char*
value is converted to a char*

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface along with [FE2_MUXER_SWF](#) and the [FE2_SWF_ADD_VARIABLE](#) parameter.

14.4.3 Property Documentation

14.4.3.1 on2sc flxengine_com::ISwfOptions::sc [get]

[on2sc](#) from the last Flix Engine function called within this interface

Deprecated

Use the [flxengine_com::IFlixPlgn](#) interface.

The documentation for this interface was generated from the following file:

- [flxengine_com.idl](#)

14.5 flxengine_com::IVideoOptions Interface Reference

Interface for accessing non-deprecated `video_options_*` functions.

```
import "flxengine_com.idl";
```

Public Member Functions

- HRESULT `reset` ()
Wrapper for `video_options_Reset()`.
- HRESULT `validate` ()
Wrapper for `video_options_Validate()`.
- HRESULT `getSwfHeight` ([out, retval] `on2s32` *lpSwfHeight)
Wrapper for `video_options_GetSwfHeight()`.
- HRESULT `setSwfHeight` (const `on2s32` lSwfHeight)
Wrapper for `video_options_SetSwfHeight()`.
- HRESULT `getSwfWidth` ([out, retval] `on2s32` *lpSwfWidth)
Wrapper for `video_options_GetSwfWidth()`.
- HRESULT `setSwfWidth` (const `on2s32` lSwfWidth)
Wrapper for `video_options_SetSwfWidth()`.
- HRESULT `getUseCustomSwfDimensions` ([out, retval] `on2bool` *bpUseCustomSwfDimensions)
Wrapper for `video_options_GetUseCustomSwfDimensions()`.
- HRESULT `setUseCustomSwfDimensions` (const `on2bool` bUseCustomSwfDimensions)
Wrapper for `video_options_SetUseCustomSwfDimensions()`.
- HRESULT `getSourceHeight` ([out, retval] `on2s32` *height)
Wrapper for `video_options_GetSourceHeight()`.
- HRESULT `getSourceWidth` ([out, retval] `on2s32` *width)
Wrapper for `video_options_GetSourceWidth()`.
- HRESULT `getVideoFramerateAsDouble` ([out, retval] double *p_fps)
Wrapper for `video_options_GetVideoFramerateAsDouble()`.
- HRESULT `setVideoFramerateAsDouble` (const double fps)
Wrapper for `video_options_SetVideoFramerateAsDouble()`.
- HRESULT `getUseSourceFramerate` ([out, retval] `on2bool` *bpUseSourceFramerate)
Wrapper for `video_options_GetUseSourceFramerate()`.
- HRESULT `setUseSourceFramerate` (const `on2bool` bUseSourceFramerate)
Wrapper for `video_options_SetUseSourceFramerate()`.

- HRESULT [getDecimateValue](#) ([out, retval] [on2u32](#) *pValue)
Wrapper for [video_options_GetDecimateValue\(\)](#).
- HRESULT [setDecimateValue](#) (const [on2u32](#) value)
Wrapper for [video_options_SetDecimateValue\(\)](#).
- HRESULT [addFLVCuePoint](#) (const BSTR pName, const double time, const [FE2_CuePointType](#) type)
Wrapper for [video_options_AddFLVCuePoint\(\)](#).
- HRESULT [addFLVCuePointParameter](#) (const BSTR pCuePointName, const BSTR pName, const BSTR pValue)
Wrapper for [video_options_AddFLVCuePointParameter\(\)](#).

Properties

- [on2sc](#) [sc](#) [get]
[on2sc](#) from the last Flix Engine function called within this interface

14.5.1 Detailed Description

Interface for accessing non-deprecated [video_options_*](#) functions.

Definition at line 457 of file [flixengine_com.idl](#).

14.5.2 Member Function Documentation

14.5.2.1 HRESULT [flixengine_com::IVideoOptions::addFLVCuePoint](#) (const BSTR *pName*, const double *time*, const [FE2_CuePointType](#) *type*)

Wrapper for [video_options_AddFLVCuePoint\(\)](#).

Note:

pName is converted to a char*

Deprecated

See [video_options_AddFLVCuePoint\(\)](#) for details.

14.5.2.2 HRESULT [flixengine_com::IVideoOptions::addFLVCuePointParameter](#) (const BSTR *pCuePointName*, const BSTR *pName*, const BSTR *pValue*)

Wrapper for [video_options_AddFLVCuePointParameter\(\)](#).

Note:

pCuePointName is converted to a char* *pName* is converted to a char* *pValue* is converted to a char*

Deprecated

See [video_options_AddFLVCuePointParameter\(\)](#) for details.

14.5.2.3 HRESULT flxengine_com::IVideoOptions::getDecimateValue ([out, retval] on2u32 * *pValue*)

Wrapper for [video_options_GetDecimateValue\(\)](#).

Deprecated

See [video_options_GetDecimateValue\(\)](#) for details.

14.5.2.4 HRESULT flxengine_com::IVideoOptions::getSourceHeight ([out, retval] on2s32 * *height*)

Wrapper for [video_options_GetSourceHeight\(\)](#).

14.5.2.5 HRESULT flxengine_com::IVideoOptions::getSourceWidth ([out, retval] on2s32 * *width*)

Wrapper for [video_options_GetSourceWidth\(\)](#).

14.5.2.6 HRESULT flxengine_com::IVideoOptions::getSwfHeight ([out, retval] on2s32 * *lpSwfHeight*)

Wrapper for [video_options_GetSwfHeight\(\)](#).

Deprecated

See [video_options_GetSwfHeight\(\)](#) for details.

14.5.2.7 HRESULT flxengine_com::IVideoOptions::getSwfWidth ([out, retval] on2s32 * *lpSwfWidth*)

Wrapper for [video_options_GetSwfWidth\(\)](#).

Deprecated

See [video_options_GetSwfWidth\(\)](#) for details.

14.5.2.8 HRESULT flxengine_com::IVideoOptions::getUseCustomSwfDimensions ([out, retval] on2bool * *bpUseCustomSwfDimensions*)

Wrapper for [video_options_GetUseCustomSwfDimensions\(\)](#).

Deprecated

See [video_options_GetUseCustomSwfDimensions\(\)](#) for details.

**14.5.2.9 HRESULT flxengine_com::IVideoOptions::getUseSourceFramerate ([out, retval]
on2bool * *bpUseSourceFramerate*)**

Wrapper for [video_options_GetUseSourceFramerate\(\)](#).

Deprecated

See [video_options_GetUseSourceFramerate\(\)](#) for details.

**14.5.2.10 HRESULT flxengine_com::IVideoOptions::getVideoFramerateAsDouble ([out, retval]
double * *p_fps*)**

Wrapper for [video_options_GetVideoFramerateAsDouble\(\)](#).

Deprecated

See [video_options_GetVideoFramerateAsDouble\(\)](#) for details.

14.5.2.11 HRESULT flxengine_com::IVideoOptions::reset ()

Wrapper for [video_options_Reset\(\)](#).

14.5.2.12 HRESULT flxengine_com::IVideoOptions::setDecimateValue (const on2u32 *value*)

Wrapper for [video_options_SetDecimateValue\(\)](#).

Deprecated

See [video_options_SetDecimateValue\(\)](#) for details.

14.5.2.13 HRESULT flxengine_com::IVideoOptions::setSwfHeight (const on2s32 *lSwfHeight*)

Wrapper for [video_options_SetSwfHeight\(\)](#).

Deprecated

See [video_options_SetSwfHeight\(\)](#) for details.

14.5.2.14 HRESULT flxengine_com::IVideoOptions::setSwfWidth (const on2s32 *lSwfWidth*)

Wrapper for [video_options_SetSwfWidth\(\)](#).

Deprecated

See [video_options_SetSwfWidth\(\)](#) for details.

14.5.2.15 HRESULT flxengine_com::IVideoOptions::setUseCustomSwfDimensions (const on2bool *bUseCustomSwfDimensions*)

Wrapper for [video_options_SetUseCustomSwfDimensions\(\)](#).

Deprecated

See [video_options_SetUseCustomSwfDimensions\(\)](#) for details.

14.5.2.16 HRESULT flxengine_com::IVideoOptions::setUseSourceFramerate (const on2bool *bUseSourceFramerate*)

Wrapper for [video_options_SetUseSourceFramerate\(\)](#).

Deprecated

See [video_options_SetUseSourceFramerate\(\)](#) for details.

14.5.2.17 HRESULT flxengine_com::IVideoOptions::setVideoFramerateAsDouble (const double *fps*)

Wrapper for [video_options_SetVideoFramerateAsDouble\(\)](#).

Deprecated

See [video_options_SetVideoFramerateAsDouble\(\)](#) for details.

14.5.2.18 HRESULT flxengine_com::IVideoOptions::validate ()

Wrapper for [video_options_Validate\(\)](#).

14.5.3 Property Documentation**14.5.3.1 on2sc flxengine_com::IVideoOptions::sc [get]**

[on2sc](#) from the last Flix Engine function called within this interface

The documentation for this interface was generated from the following file:

- [flxengine_com.idl](#)

Chapter 15

File Documentation

15.1 about.dox File Reference

15.2 flxengine2/audio_options.h File Reference

Reference module(s): [Audio Encoding Options](#).

Enumerations

- enum [FE2_AudioBitrates](#) {

[Bitrate8kbps](#),
[Bitrate16kbps](#),
[Bitrate24kbps](#),
[Bitrate32kbps](#),
[Bitrate40kbps](#),
[Bitrate48kbps](#),
[Bitrate56kbps](#),
[Bitrate64kbps](#),
[Bitrate80kbps](#),
[Bitrate96kbps](#),
[Bitrate112kbps](#),
[Bitrate128kbps](#),
[Bitrate144kbps](#),
[Bitrate160kbps](#),
[Bitrate192kbps](#),
[Bitrate224kbps](#),
[Bitrate256kbps](#),
[Bitrate320kbps](#) }

Valid bitrates for [FE2_CODEC_LAME](#). For use in calls to [Flix2_CodecSetParam\(\)](#) and [Flix2_CodecGetParam\(\)](#).

- enum [FE2_FlvAudioFormat](#) {

[FlvAudioUncompressed](#),
[FlvAudioMp3](#) }

Valid output audio formats for use in calls to [audio_options_GetFlvAudioFormat\(\)](#) and [audio_options_SetFlvAudioFormat\(\)](#).

- enum [FE2_AudioSamplingrates](#) {

[Hertz11025](#),
[Hertz22050](#),
[Hertz44100](#) }

Sample rates for use with [FE2_FILTER_RESAMPLE](#) and the [FE2_RESAMPLE_RATE](#) parameter.

Functions

- [on2sc audio_options_Reset](#) ([FLIX2HANDLE](#) flx)
Reset the audio encoding options to their default values.
- [on2sc audio_options_Validate](#) ([FLIX2HANDLE](#) flx)
Validate the current audio encoding options.
- [on2sc audio_options_SetBitrate](#) ([FLIX2HANDLE](#) flx, const [FE2_AudioBitrates](#) bitrate)
Set the bitrate to use in mp3 encoding.
- [on2sc audio_options_GetBitrate](#) (const [FLIX2HANDLE](#) flx, [FE2_AudioBitrates](#) *pBitrate)
Retrieve the current mp3 bitrate.
- [on2sc audio_options_SetFlvAudioFormat](#) ([FLIX2HANDLE](#) flx, const [FE2_FlvAudioFormat](#) format)
Set the output audio format.
- [on2sc audio_options_GetFlvAudioFormat](#) (const [FLIX2HANDLE](#) flx, [FE2_FlvAudioFormat](#) *pFormat)
Retrieve the current output audio format.
- [on2sc audio_options_SetSamplingrate](#) ([FLIX2HANDLE](#) flx, const [FE2_AudioSamplingrates](#) samplingrate)
Set the output audio sample rate.
- [on2sc audio_options_GetSamplingrate](#) (const [FLIX2HANDLE](#) flx, [FE2_AudioSamplingrates](#) *pSamplingrate)
Retrieve the current output sample rate.
- [on2sc audio_options_SetStereo](#) ([FLIX2HANDLE](#) flx, const [on2bool](#) stereo)
Set the number of output audio channels.
- [on2sc audio_options_GetStereo](#) (const [FLIX2HANDLE](#) flx, [on2bool](#) *pStereo)
Retrieve the current number of output audio channels.

15.2.1 Detailed Description

Reference module(s): [Audio Encoding Options](#).

Definition in file [audio_options.h](#).

15.3 flxengine2/codec_constants.h File Reference

Reference module(s): [Codecs](#). #include "codecs/codec_common.h"

```
#include "codecs/h263.h"
#include "codecs/h264.h"
#include "codecs/vp6.h"
#include "codecs/vp6_alpha.h"
#include "codecs/vp8.h"
#include "codecs/aac.h"
#include "codecs/amr.h"
#include "codecs/lame.h"
#include "codecs/vorbis.h"
```

Defines

- #define [FE2_CODEC_PCM](#)
'Codec' name for use with [Flix2_AddCodec\(\)](#)

15.3.1 Detailed Description

Reference module(s): [Codecs](#).

Definition in file [codec_constants.h](#).

15.4 flxengine2/codecs/aac.h File Reference

Reference module(s): [AAC](#), [AAC+](#).

Defines

- #define [FE2_CODEC_AAC](#)
AAC. Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_AAC_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_CODEC_AACPLUS](#)
AAC+ (HE-AAC). Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_AACPLUS_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_AACPLUS_PARAMETRIC_STEREO](#)
Parameter for parametric stereo.

15.4.1 Detailed Description

Reference module(s): [AAC](#), [AAC+](#).

Definition in file [aac.h](#).

15.5 flxengine2/codecs/amr.h File Reference

Reference module(s): [AMR_NB](#) - FFmpeg.

Defines

- #define [FE2_CODEC_AMR_NB](#)
AMR Narrowband. Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_AMR_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).

15.5.1 Detailed Description

Reference module(s): [AMR_NB](#) - FFmpeg.

Definition in file [amr.h](#).

15.6 flixengine2/codecs/codec_common.h File Reference

Defines

- #define [FE2_CODECPARAM_BITRATE](#)
Codec parameter for stream bitrate.
- #define [FE2_VCODECPARAM_RC_MODE](#)
Video codec parameter for the rate control mode.
- #define [FE2_VCODECPARAM_KFINTTYPE](#)
Video codec parameter for the keyframe interval type.
- #define [FE2_VCODECPARAM_KFFREQ](#)
Video codec parameter for keyframe frequency.

Enumerations

- enum [FE2_VideoBitrateControls](#) {
 [CBR_1PASSControl](#),
 [VBR_1PASSControl](#),
 [CBR_2PASSControl](#),
 [VBR_2PASSControl](#) }
Encoder rate control types, influences quality.
- enum [FE2_VideoKeyframeTypes](#) {
 [MAX_KEYFRAMES](#),
 [FIXED_KEYFRAMES](#) }
Key frame interval types, influences quality.

15.7 flxengine2/codecs/h263.h File Reference

Reference module(s): [H263 - FFmpeg](#).

Defines

- #define [FE2_CODEC_H263](#)
Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_CODEC_H263_BASELINE](#)
Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_H263_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_H263_KFINTTYPE](#)
Alias for [FE2_VCODECPARAM_KFINTTYPE](#).
- #define [FE2_H263_KFFREQ](#)
Alias for [FE2_VCODECPARAM_KFFREQ](#).
- #define [FE2_H263_RC_MODE](#)
Alias for [FE2_VCODECPARAM_RC_MODE](#).
- #define [FE2_H263_MIN_Q](#)
Codec parameter for minimum quantizer.
- #define [FE2_H263_MAX_Q](#)
Codec parameter for maximum quantizer.

15.7.1 Detailed Description

Reference module(s): [H263 - FFmpeg](#).

Definition in file [h263.h](#).

15.8 flxengine2/codecs/h264.h File Reference

Reference module(s): [H264](#).

Defines

- #define [FE2_CODEC_H264](#)
Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_H264_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_H264_KFINTTYPE](#)
Alias for [FE2_VCODECPARAM_KFINTTYPE](#).
- #define [FE2_H264_KFFREQ](#)
Alias for [FE2_VCODECPARAM_KFFREQ](#).
- #define [FE2_H264_RC_MODE](#)
Alias for [FE2_VCODECPARAM_RC_MODE](#).
- #define [FE2_H264_PROFILE](#)
Codec parameter for encoding profile.
- #define [FE2_H264_B_FRAME_RATE](#)
Codec parameter for B frame rate.
- #define [FE2_H264_SPEED](#)
Controls frame analysis, affecting encoder speed and inversely output quality.

Typedefs

- typedef enum [h264profile](#) [h264profile_t](#)
Valid profiles for use with the [FE2_H264_PROFILE](#) parameter.

Enumerations

- enum [h264profile](#) {
 [BASE_H264PROFILE](#),
 [MAIN_H264PROFILE](#),
 [HIGH_H264PROFILE](#) }
Valid profiles for use with the [FE2_H264_PROFILE](#) parameter.

15.8.1 Detailed Description

Reference module(s): [H264](#).

Definition in file [h264.h](#).

15.9 flxengine2/codecs/lame.h File Reference

Reference module(s): [MP3 - LAME](#).

Defines

- `#define FE2_CODEC_LAME`
Codec name for use with [Flix2_AddCodec\(\)](#).
- `#define FE2_LAME_BITRATE`
Alias for [FE2_CODECPARAM_BITRATE](#).
- `#define FE2_LAME_QUALITY`
Codec parameter for controlling LAME library's algorithm selection.
- `#define FE2_LAME_RC_MODE`
Codec parameter for controlling LAME library's rate control method.
- `#define FE2_LAME_CHANNELS`
Codec parameter for number of output channels.

Typedefs

- `typedef enum lame_rcmode lame_rcmode_t`
Rate control modes analogous to those found in `<lame/lame.h>`.

Enumerations

- `enum lame_rcmode {`
 [LAME_CBR](#),
 [LAME_ABR](#),
 [LAME_VBR_rh](#),
 [LAME_VBR_mtrh](#) }
Rate control modes analogous to those found in `<lame/lame.h>`.

15.9.1 Detailed Description

Reference module(s): [MP3 - LAME](#).

Definition in file [lame.h](#).

15.10 flxengine2/codecs/vorbis.h File Reference

Reference module(s): [libvorbis](#) - FFmpeg.

Defines

- #define [FE2_CODEC_VORBIS](#)
Vorbis. Codec name for use with [Flix2_AddCodec\(\)](#).
- #define [FE2_VORBIS_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).

15.10.1 Detailed Description

Reference module(s): [libvorbis](#) - FFmpeg.

Definition in file [vorbis.h](#).

15.11 flxengine2/codecs/vp6.h File Reference

Reference module(s): [VP6](#).

Defines

Codec name

- `#define FE2_CODEC_VP6`
Codec name for use with `Flix2_AddCodec()`.

Advanced VP6 settings

- `#define FE2_VP6_CONCURRENCY`
Codec parameter for concurrency level.
- `#define FE2_VP6_UNDERSHOOT_PCT`
Codec parameter for undershoot percentage.
- `#define FE2_VP6_MIN_Q`
Codec parameter for minimum quantizer.
- `#define FE2_VP6_MAX_Q`
Codec parameter for maximum quantizer.
- `#define FE2_VP6_TEMPORAL_RESAMPLING`
Codec parameter for temporal resampling.
- `#define FE2_VP6_TEMPORAL_DOWN_WATERMARK`
Codec parameter for temporal down watermark percentage.

CBR-Specific

- `#define FE2_VP6_STREAM_PEAK_BITRATE`
The maximum bitrate allowed in the stream.
- `#define FE2_VP6_STREAM_PREBUFFER`
Seconds of preload that are necessary before starting playback.
- `#define FE2_VP6_STREAM_OPTIMAL_BUFFER`
Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.
- `#define FE2_VP6_STREAM_MAX_BUFFER`
The maximum size of the buffer, in seconds.

VBR-Specific

- `#define FE2_VP6_2PASS_MIN_SECTION`
VBR_2PASSControl minimum section datarate
- `#define FE2_VP6_2PASS_MAX_SECTION`
VBR_2PASSControl maximum section datarate

General VP6 settings

- `#define FE2_VP6_BITRATE`
Alias for `FE2_CODECPARAM_BITRATE`.
- `#define FE2_VP6_KFINTTYPE`
Alias for `FE2_VCODECPARAM_KFINTTYPE`.
- `#define FE2_VP6_KFFREQ`
Alias for `FE2_VCODECPARAM_KFFREQ`.
- `#define FE2_VP6_RC_MODE`
Alias for `FE2_VCODECPARAM_RC_MODE`.
- `#define FE2_VP6_CXMODE`
VP6 compress mode.
- `#define FE2_VP6_SHARPNESS`
Codec parameter for sharpness.
- `#define FE2_VP6_NOISE_REDUCTION`
Codec parameter for noise reduction.
- `#define FE2_VP6_PROFILE`
Codec parameter for encoding profile.
- `enum FE2_CompressMode {`
 `COMPRESSMODE_GOOD,`
 `COMPRESSMODE_BEST }`
Valid compress modes for VP6, influences encoder speed.
- `enum vp6profile {`
 `VP6_E,`
 `VP6_S }`
Valid profiles for use with the `FE2_VP6_PROFILE` parameter.
- `typedef enum vp6profile vp6profile_t`
Valid profiles for use with the `FE2_VP6_PROFILE` parameter.

15.11.1 Detailed Description

Reference module(s): `VP6`.

Definition in file `vp6.h`.

15.12 flxengine2/codecs/vp6_alpha.h File Reference

Reference module(s): [VP6 with Alpha](#).

Defines

Codec name

- #define [FE2_CODEC_VP6ALPHA](#)
Codec name for use with [Flix2_AddCodec\(\)](#).

General VP6 settings

- #define [FE2_VP6A_BITRATE](#)
Alias for [FE2_CODECPARAM_BITRATE](#).
- #define [FE2_VP6A_ALPHA_BITRATE](#)
Compressed bitrate of the alpha channel in kbit/s.
- #define [FE2_VP6A_KFINTTYPE](#)
Alias for [FE2_VP6_KFINTTYPE](#).
- #define [FE2_VP6A_KFFREQ](#)
Alias for [FE2_VP6_KFFREQ](#).
- #define [FE2_VP6A_RC_MODE](#)
Alias for [FE2_VCODECPARAM_RC_MODE](#).
- #define [FE2_VP6A_CXMODE](#)
Alias for [FE2_VP6_CXMODE](#).
- #define [FE2_VP6A_SHARPNESS](#)
Alias for [FE2_VP6_SHARPNESS](#).
- #define [FE2_VP6A_ALPHA_SHARPNESS](#)
Sharpness for the alpha channel.
- #define [FE2_VP6A_NOISE_REDUCTION](#)
Alias for [FE2_VP6_NOISE_REDUCTION](#).
- #define [FE2_VP6A_ALPHA_NOISE_REDUCTION](#)
Noise reduction for the alpha channel.

Advanced VP6 settings

- #define [FE2_VP6A_UNDERSHOOT_PCT](#)
Alias for [FE2_VP6_UNDERSHOOT_PCT](#).
- #define [FE2_VP6A_MIN_Q](#)
Alias for [FE2_VP6_MIN_Q](#).
- #define [FE2_VP6A_ALPHA_MIN_Q](#)

Minimum quantizer for the alpha channel.

- `#define FE2_VP6A_MAX_Q`
Alias for `FE2_VP6_MAX_Q`.
- `#define FE2_VP6A_ALPHA_MAX_Q`
Maximum quantizer for the alpha channel.
- `#define FE2_VP6A_TEMPORAL_RESAMPLING`
Alias for `FE2_VP6_TEMPORAL_RESAMPLING`.
- `#define FE2_VP6A_TEMPORAL_DOWN_WATERMARK`
Alias for `FE2_VP6_TEMPORAL_DOWN_WATERMARK`.
- `#define FE2_VP6A_STREAM_PEAK_BITRATE`
Alias for `FE2_VP6_STREAM_PEAK_BITRATE`.
- `#define FE2_VP6A_STREAM_PREBUFFER`
Alias for `FE2_VP6_STREAM_PREBUFFER`.
- `#define FE2_VP6A_STREAM_OPTIMAL_BUFFER`
Alias for `FE2_VP6_STREAM_OPTIMAL_BUFFER`.
- `#define FE2_VP6A_STREAM_MAX_BUFFER`
Alias for `FE2_VP6_STREAM_MAX_BUFFER`.
- `#define FE2_VP6A_2PASS_MIN_SECTION`
Alias for `FE2_VP6_2PASS_MIN_SECTION`.
- `#define FE2_VP6A_2PASS_MAX_SECTION`
Alias for `FE2_VP6_2PASS_MAX_SECTION`.

15.12.1 Detailed Description

Reference module(s): [VP6 with Alpha](#).

Definition in file [vp6_alpha.h](#).

15.13 flxengine2/codecs/vp8.h File Reference

Reference module(s): [VP8](#).

Defines

- `#define FE2_VP8_PROFILE`
Bitstream profile.
- `#define FE2_VP8_ALTREF`
Enable the use of alternate reference frames.
- `#define FE2_VP8_AR_MAX_FRAMES`
Max number of frames blurred creating alternate reference.
- `#define FE2_VP8_AR_TYPE`
Filter type to use w/alternate reference.
- `#define FE2_VP8_AR_STRENGTH`
Filter strength for the alternate reference.
- `#define FE2_VP8_MB_STATIC_THRESHOLD`
Threshold for macroblocks treated static.
- `#define FE2_VP8_TOKEN_PARTITIONS`
Number of token partitions.
- `#define FE2_VP8_LAG`
Allow lagged encoding If set, this value allows the encoder to consume a number of input frames before producing output frames. This allows the encoder to base decisions for the current frame on future frames.
- `#define FE2_VP8_THREADS`
Number of threads to use A reasonable selection would be the number of cores on the system.

Codec name

- `#define FE2_CODEC_VP8`
Codec name for use with [Flix2_AddCodec\(\)](#).

General VP8 settings

- `#define FE2_VP8_BITRATE`
Alias for [FE2_CODECPARAM_BITRATE](#).
- `#define FE2_VP8_KFINTTYPE`
Alias for [FE2_VCODECPARAM_KFINTTYPE](#).
- `#define FE2_VP8_KFFREQ`
Alias for [FE2_VCODECPARAM_KFFREQ](#).

- `#define FE2_VP8_RC_MODE`
Alias for `FE2_VCODECPARAM_RC_MODE`.
- `#define FE2_VP8_CXMODE`
VP8 compress mode.
- `#define FE2_VP8_SHARPNESS`
Codec parameter for sharpness.
- `#define FE2_VP8_NOISE_REDUCTION`
Controls encoder noise reduction preprocessing.

Advanced VP8 settings

- `#define FE2_VP8_UNDERSHOOT_PCT`
Codec parameter for undershoot percentage.
- `#define FE2_VP8_OVERSHOOT_PCT`
Codec parameter for undershoot percentage.
- `#define FE2_VP8_MIN_Q`
Codec parameter for minimum quantizer.
- `#define FE2_VP8_MAX_Q`
Codec parameter for maximum quantizer.
- `#define FE2_VP8_DROP_THRESH`
Threshold controlling encoder frame dropping.

CBR-Specific

- `#define FE2_VP8_STREAM_INITIAL_BUFFER`
Seconds of preload that are necessary before starting playback.
- `#define FE2_VP8_STREAM_OPTIMAL_BUFFER`
Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.
- `#define FE2_VP8_STREAM_MAX_BUFFER`
The maximum size of the buffer, in seconds.

VBR-Specific

- `#define FE2_VP8_2PASS_MIN_SECTION`
VBR_2PASSControl minimum section datarate
- `#define FE2_VP8_2PASS_MAX_SECTION`
VBR_2PASSControl maximum section datarate

15.13.1 Detailed Description

Reference module(s): `VP8`.

Definition in file `vp8.h`.

15.14 flxengine2/encoding_status.h File Reference

Reference module(s): [Encoding Statistics](#).

Functions

- [on2sc encoding_status_GetAverageBitrate](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pBitrate)
Retrieve the encoder's average (video) bitrate.
- [on2sc encoding_status_GetAverageFramesize](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pFramesize)
Retrieve the encoder's average (video) frame size.
- [on2sc encoding_status_GetMaximumFramesize](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pMaxFramesize)
Retrieve the encoder's maximum (video) frame size.
- [on2sc encoding_status_GetMinimumFramesize](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pMinFramesize)
Retrieve the encoder's minimum (video) frame size.
- [on2sc encoding_status_GetTotalFrames](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pTotalFrames)
Retrieve the total number of (video) frames encoded.
- [on2sc encoding_status_GetElapsedTime](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pElapsedTime)
Retrieve the total elapsed encode time.
- [on2sc encoding_status_GetEndTime](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pEndTime)
Retrieve the encode completion time.
- [on2sc encoding_status_GetStartTime](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *pStartTime)
Retrieve the encode start time.
- [on2sc encoding_status_PercentComplete](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *percent)
Retrieve the percent of the encode complete.

15.14.1 Detailed Description

Reference module(s): [Encoding Statistics](#).

Definition in file [encoding_status.h](#).

15.15 flixengine2/filter_constants.h File Reference

Reference module(s): [Filters](#). #include "filters/cut.h"

#include "filters/adaptive_deinterlace.h"

#include "filters/bchs.h"

#include "filters/blur.h"

#include "filters/crop.h"

#include "filters/denoise.h"

#include "filters/framerate.h"

#include "filters/mirror.h"

#include "filters/overlay.h"

#include "filters/png_export.h"

#include "filters/rotate.h"

#include "filters/scale.h"

#include "filters/sharpen.h"

#include "filters/highpass.h"

#include "filters/lowpass.h"

#include "filters/resample.h"

15.15.1 Detailed Description

Reference module(s): [Filters](#).

Definition in file [filter_constants.h](#).

15.16 flxengine2/filters/adaptive_deinterlace.h File Reference

Reference module(s): [Deinterlace](#).

Defines

- #define [FE2_FILTER_ADAPTIVE_DEINTERLACE](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_ADAPTIVE_DEINTERLACE_MODE](#)
Specifies deinterlace mode to be applied to source image.

Typedefs

- typedef enum [deinterlacemode](#) [deintmode_t](#)

Enumerations

- enum [deinterlacemode](#) {
 [DEINTERLACE_NONE](#),
 [DEINTERLACE_1_2_1_BLUR](#),
 [DEINTERLACE_DROP_FIELD](#),
 [DEINTERLACE_ADAPTIVE](#) }

15.16.1 Detailed Description

Reference module(s): [Deinterlace](#).

Definition in file [adaptive_deinterlace.h](#).

15.17 flixengine2/filters/bchs.h File Reference

Reference module(s): [Brightness/Contrast/Hue/Saturation](#).

Defines

- #define [FE2_FILTER_BCHS](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_BCHS_BRIGHTNESS](#)
Parameter for the brightness adjustment factor.
- #define [FE2_BCHS_CONTRAST](#)
Parameter for the contrast adjustment factor.
- #define [FE2_BCHS_HUE](#)
Parameter for the hue adjustment factor.
- #define [FE2_BCHS_SATURATION](#)
Parameter for the saturation adjustment factor.

15.17.1 Detailed Description

Reference module(s): [Brightness/Contrast/Hue/Saturation](#).

Definition in file [bchs.h](#).

15.18 flixengine2/filters/blur.h File Reference

Reference module(s): [Blur](#).

Defines

- #define [FE2_FILTER_BLUR](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_BLUR_FILTER](#)
Specifies blur filter to be applied to source.
- #define [FE2_BLUR_MASKSIZE](#)
Specifies the kernel/mask size to be used.

Typedefs

- typedef enum [masksiz](#) [masksiz_t](#)
Filter mask/kernel size.
- typedef enum [blurfilter](#) [blurfilter_t](#)

Enumerations

- enum [masksiz](#) {
 [MASK_3x3](#),
 [MASK_5x5](#) }
Filter mask/kernel size.
- enum [blurfilter](#) {
 [BLUR_LOWPASS](#),
 [BLUR_GAUSS](#) }

15.18.1 Detailed Description

Reference module(s): [Blur](#).

Definition in file [blur.h](#).

15.19 flixengine2/filters/crop.h File Reference

Reference module(s): [Crop](#).

Defines

- #define [FE2_FILTER_CROP](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_CROP_TOP](#)
Filter parameter for setting bounding box's top coordinate.
- #define [FE2_CROP_BOTTOM](#)
Filter parameter for setting bounding box's bottom coordinate.
- #define [FE2_CROP_LEFT](#)
Filter parameter for setting bounding box's left coordinate.
- #define [FE2_CROP_RIGHT](#)
Filter parameter for setting bounding box's right coordinate.

15.19.1 Detailed Description

Reference module(s): [Crop](#).

Definition in file [crop.h](#).

15.20 flixengine2/filters/cut.h File Reference

Reference module(s): [Cut](#).

Defines

- #define [FE2_FILTER_CUT](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_CUT_START_SEC](#)
Filter parameter for cut start time.
- #define [FE2_CUT_STOP_SEC](#)
Filter parameter for setting the cut stop time.
- #define [FE2_CUT_USE_SEEK](#)
Set [FE2_CUT_START_SEC](#) application method.

15.20.1 Detailed Description

Reference module(s): [Cut](#).

Definition in file [cut.h](#).

15.21 flixengine2/filters/denoise.h File Reference

Reference module(s): [Denoise](#).

Defines

- #define [FE2_FILTER_DENOISE](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_DENOISE_NOISE_LEVEL](#)
Specifies noise level of the source.

15.21.1 Detailed Description

Reference module(s): [Denoise](#).

Definition in file [denoise.h](#).

15.22 flxengine2/filters/framerate.h File Reference

Reference module(s): [Frame Rate](#).

Defines

- #define [FE2_FILTER_FRAMERATE](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_FRAMERATE_FPS](#)
Filter parameter name for frames per second value.
- #define [FE2_FRAMERATE_DECIMATE](#)
Filter parameter for the decimation interval.

15.22.1 Detailed Description

Reference module(s): [Frame Rate](#).

Definition in file [framerate.h](#).

15.23 flixengine2/filters/highpass.h File Reference

Reference module(s): [Highpass](#).

Defines

- #define [FE2_FILTER_HIGHPASS](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_HIGHPASS_Q](#)
Filter parameter for shape constant ("Q" coefficient).
- #define [FE2_HIGHPASS_CUTOFF](#)
Filter parameter for cutoff frequency.

15.23.1 Detailed Description

Reference module(s): [Highpass](#).

Definition in file [highpass.h](#).

15.24 flxengine2/filters/lowpass.h File Reference

Reference module(s): [Lowpass](#).

Defines

- #define [FE2_FILTER_LOWPASS](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_LOWPASS_Q](#)
Filter parameter for shape constant ("Q" coefficient).
- #define [FE2_LOWPASS_CUTOFF](#)
Filter parameter for cutoff frequency.

15.24.1 Detailed Description

Reference module(s): [Lowpass](#).

Definition in file [lowpass.h](#).

15.25 flixengine2/filters/mirror.h File Reference

Reference module(s): [Mirror](#).

Defines

- #define [FE2_FILTER_MIRROR](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_MIRROR_HORIZONTAL](#)
Specifies horizontal disposition.
- #define [FE2_MIRROR_VERTICAL](#)
Specifies vertical disposition.

15.25.1 Detailed Description

Reference module(s): [Mirror](#).

Definition in file [mirror.h](#).

15.26 flxengine2/filters/overlay.h File Reference

Reference module(s): [Overlay \(Watermark\)](#).

Defines

- #define [FE2_FILTER_OVERLAY](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_OVERLAY_FILE](#)
Set the path to the overlay image file.
- #define [FE2_OVERLAY_MASK_XY](#)
Use the pixel at coordinates ([FE2_OVERLAY_MASK_X](#),[FE2_OVERLAY_MASK_Y](#)) to determine the transparent color.
- #define [FE2_OVERLAY_MASK_X](#)
X coordinate of pixel to use for transparency.
- #define [FE2_OVERLAY_MASK_Y](#)
Y coordinate of pixel to use for transparency.
- #define [FE2_OVERLAY_MASK_RGB](#)
Use the RGB value ([FE2_OVERLAY_MASK_R](#), [FE2_OVERLAY_MASK_G](#), [FE2_OVERLAY_MASK_B](#)) as the transparency color.
- #define [FE2_OVERLAY_MASK_R](#)
Red component of the transparency color.
- #define [FE2_OVERLAY_MASK_G](#)
Green component of the transparency color.
- #define [FE2_OVERLAY_MASK_B](#)
Blue component of the transparency color.
- #define [FE2_OVERLAY_POS](#)
Set the overlay position. Valid modes are defined by [FE2_OverlayPositionMode](#).
- #define [FE2_OVERLAY_POS_X](#)
X coordinate of overlay top left on video.
- #define [FE2_OVERLAY_POS_Y](#)
Y coordinate of overlay top left on video.

Enumerations

- `enum FE2_OverlayPositionMode {`
 `FE2_OVERLAY_POS_MODE_TOPLEFT,`
 `FE2_OVERLAY_POS_MODE_BOTLEFT,`
 `FE2_OVERLAY_POS_MODE_CENTER,`
 `FE2_OVERLAY_POS_MODE_TOPRIGHT,`
 `FE2_OVERLAY_POS_MODE_BOTRIGHT,`
 `FE2_OVERLAY_POS_MODE_XY }`
 Position modes for use with `FE2_OVERLAY_POS`.

15.26.1 Detailed Description

Reference module(s): [Overlay \(Watermark\)](#).

Definition in file [overlay.h](#).

15.27 flxengine2/filters/png_export.h File Reference

Reference module(s): [PNG Image Export \(Thumbnail\)](#).

Defines

- #define [FE2_FILTER_PNGEX](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_PNGEX_OUTPUT_DIRECTORY](#)
Filter parameter for setting PNG image output directory.
- #define [FE2_PNGEX_FILENAME_PREFIX](#)
Filter parameter for setting PNG image file name prefix.
- #define [FE2_PNGEX_FILENAME_SUFFIX](#)
Filter parameter for setting PNG image file name suffix.
- #define [FE2_PNGEX_ENABLE_ALPHA](#)
Filter parameter for preserving source video alpha channel data in exported PNG images.
- #define [FE2_PNGEX_EXPORT_INTERVAL](#)
Filter parameter for enabling PNG image export at a millisecond interval.
- #define [FE2_PNGEX_EXPORT_TIME_STRING](#)
Filter parameter for setting a string of PNG export times in milliseconds.
- #define [FE2_PNGEX_EXPORT_CUE_POINTS](#)
Filter parameter for enabling export of PNG images at cue points.
- #define [FE2_PNGEX_AUTO_EXPORT_COUNT](#)
Filter parameter for enabling automatic PNG image generation.
- #define [FE2_PNGEX_AUTO_EXPORT_START_TIME](#)
Filter parameter for setting PNG auto generation start time.
- #define [FE2_PNGEX_AUTO_EXPORT_END_TIME](#)
Filter parameter for setting PNG auto generation end time.
- #define [FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD](#)
Filter parameter for randomizing auto generated times.
- #define [FE2_PNGEX_EXPORT_FIRST_FRAME_PNG](#)
Filter parameter for enabling PNG image creation using the first video frame.
- #define [FE2_PNGEX_COMPRESSION_LEVEL](#)
Filter parameter for setting compression level used by libpng.
- #define [FE2_PNGEX_WIDTH](#)

Filter parameter for setting PNG image width.

- `#define` [FE2_PNGEX_HEIGHT](#)

Filter parameter for setting PNG image height.

Enumerations

- `enum` [FE2_PNGExCuePtMode](#) {
 [FE2_PNGEX_CP_EVENT](#),
 [FE2_PNGEX_CP_NAV](#),
 [FE2_PNGEX_CP_ALL](#) }

Constants for use with [FE2_PNGEX_EXPORT_CUE_POINTS](#).

15.27.1 Detailed Description

Reference module(s): [PNG Image Export \(Thumbnail\)](#).

Definition in file [png_export.h](#).

15.28 flxengine2/filters/resample.h File Reference

Reference module(s): [Resample](#).

Defines

- #define [FE2_FILTER_RESAMPLE](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_RESAMPLE_RATE](#)
Desired sample rate in Hertz (Hz).
- #define [FE2_RESAMPLE_CHANNELS](#)
Filter parameter for channels.

15.28.1 Detailed Description

Reference module(s): [Resample](#).

Definition in file [resample.h](#).

15.29 flixengine2/filters/rotate.h File Reference

Reference module(s): [Rotate](#).

Defines

- #define [FE2_FILTER_ROTATE](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_ROTATE_ANGLE](#)
Specifies the image rotation angle in degrees.

15.29.1 Detailed Description

Reference module(s): [Rotate](#).

Definition in file [rotate.h](#).

15.30 flixengine2/filters/scale.h File Reference

Reference module(s): [Scale](#).

Defines

- #define [FE2_FILTER_SCALE](#)
Filter name for use with [Flix2_AddFilter\(\)](#).
- #define [FE2_SCALE_WIDTH](#)
Filter parameter name for scaled width.
- #define [FE2_SCALE_HEIGHT](#)
Filter parameter name for scaled height.

15.30.1 Detailed Description

Reference module(s): [Scale](#).

Definition in file [scale.h](#).

15.31 flixengine2/filters/sharpen.h File Reference

Reference module(s): [Sharpen](#).

Defines

- #define [FE2_FILTER_SHARPEN](#)
Filter name for use with [Flix2_AddFilter\(\)](#).

15.31.1 Detailed Description

Reference module(s): [Sharpen](#).

Definition in file [sharpen.h](#).

15.32 flixengine2/flixengine2.h File Reference

Reference module(s): [Main Engine Interface](#), [Filter Manipulation](#), [Codec Manipulation](#), [Muxer Manipulation](#). #include "on2types.h"

```
#include "filter_constants.h"
#include "codec_constants.h"
#include "muxer_constants.h"
#include "video_options.h"
#include "audio_options.h"
#include "swf_options.h"
#include "encoding_status.h"
#include "media_editor_options.h"
#include "overlay_options.h"
```

Defines

- #define [FLIXENGINE_API](#)
- #define [FLIXENGINE_VERSION_CHIEF](#)
- #define [FLIXENGINE_VERSION_MAJOR](#)
- #define [FLIXENGINE_VERSION_MINOR](#)
- #define [FLIXENGINE_VERSION_PATCH](#)
- #define [FLIXENGINE_VERSION_EXTRA](#)
- #define [FLIXENGINE_VERSION_STR](#)

Typedefs

- typedef void * [FLIX2HANDLE](#)
- typedef void * [FLIX2PLGNHANDLE](#)

Enumerations

- enum [FE2_ExportedVideoType](#) {
 [ExportSWF3To6Video](#),
 [ExportSWF7PlusVideo](#),
 [ExportSWFVectorVideo](#),
 [ExportFLVVideo](#) }

Output file types for use in calls to [Flix2_SetExportVideoType\(\)](#) and [Flix2_GetExportVideoType\(\)](#).

- enum [FE2_EncState](#) {
 [EncStateIdle](#),
 [EncStateRunning](#),
 [EncStateQueued](#) }

Encoder state returned by [Flix2_GetEncoderState\(\)](#).

- enum `FE2_errno` {
`ErrNone`,
`ErrSys`,
`ErrFileIO`,
`ErrFileOpen`,
`ErrFileDecode`,
`ErrFileDecodeA`,
`ErrFileDecodeV`,
`ErrEncodeA`,
`ErrEncodeV` }

Flix engine error state returned by `Flix2_Errno()`.

Functions

- `on2sc Flix2_Create (FLIX2HANDLE *pFlix)`
Create a handle to the flix engine.
- `on2sc Flix2_Destroy (FLIX2HANDLE flix)`
Frees resources associated with a `FLIX2HANDLE`.
- `const char * Flix2_Version ()`
Returns the library version as a string.
- `const on2tc * Flix2_Copyright ()`
Returns copyright information for this library as a string.
- `on2sc Flix2_SetOutputFile (FLIX2HANDLE flix, const on2tc *outputFile)`
Set the destination file for the encode session.
- `on2sc Flix2_GetOutputFile (const FLIX2HANDLE flix, on2tc *pOutputFile, int32_t *len)`
Get the destination file for the encode session.
- `on2sc Flix2_SetOverwriteExistingFiles (FLIX2HANDLE flix, const on2bool bOverwriteExistingFiles)`
Enable/disable overwriting of existing output files.
- `on2sc Flix2_GetOverwriteExistingFiles (const FLIX2HANDLE flix, on2bool *bpOverwriteExistingFiles)`
Retrieve the engine's current behavior regarding existing output files.
- `on2sc Flix2_SetExportAudio (FLIX2HANDLE flix, const on2bool bExportAudio)`
Enable/disable audio output in the current encoding session.
- `on2sc Flix2_GetExportAudio (const FLIX2HANDLE flix, on2bool *bpExportAudio)`
Retrieve the engine's current behavior regarding audio output.
- `on2sc Flix2_SetExportVideo (FLIX2HANDLE flix, const on2bool bExportVideo)`

Enable/disable video output in the current encoding session.

- `on2sc Flix2_GetExportVideo` (const `FLIX2HANDLE` flix, `on2bool` *bpExportVideo)
Retrieve the engine's current behavior regarding video output.
- `on2sc Flix2_SetExportVideoType` (`FLIX2HANDLE` flix, const `FE2_ExportedVideoType` exportVideoType)
Set the output file type.
- `on2sc Flix2_GetExportVideoType` (const `FLIX2HANDLE` flix, `FE2_ExportedVideoType` *pExportVideoType)
Retrieve the current output file type.
- `on2sc Flix2_SetInputFile` (`FLIX2HANDLE` flix, const `on2tc` *inputFile)
Set the source file for the encode session.
- `on2sc Flix2_GetInputFile` (const `FLIX2HANDLE` flix, `on2tc` *pInputFile, `int32_t` *len)
Get the source file for the encode session.
- `on2sc Flix2_GetSourceDuration` (const `FLIX2HANDLE` flix, `int32_t` *duration)
Get the duration, in milliseconds, of the source file.
- `on2sc Flix2_Encode` (`FLIX2HANDLE` flix)
Start encoding an output file.
- `on2sc Flix2_StopEncoding` (`FLIX2HANDLE` flix)
Cancel a running encoding session.
- `on2sc Flix2_Reset` (`FLIX2HANDLE` flix)
Reset the engine to its defaults.
- `on2sc Flix2_Validate` (const `FLIX2HANDLE` flix)
Validate the current encoder settings.
- `on2sc Flix2_IsEncoderRunning` (const `FLIX2HANDLE` flix, `on2bool` *bpIsEncoderRunning)
Check the status of an encode.
- `on2sc Flix2_GetEncoderState` (const `FLIX2HANDLE` flix, `FE2_EncState` *pEncState)
Retrieve the current state of the encoder.
- `on2sc Flix2_SetLogLevel` (`FLIX2HANDLE` flix, `int32_t` level)
Set the library-wide debug log level.
- `on2sc Flix2_GetLogLevel` (const `FLIX2HANDLE` flix, `int32_t` *level)
Retrieve current library-wide debug log level.
- `on2sc Flix2_SetLogPath` (`FLIX2HANDLE` flix, `on2tc` *logpath)
Set the library's log file path.
- `on2sc Flix2_GetLogPath` (const `FLIX2HANDLE` flix, `on2tc` *logpath, `int32_t` *len)

Retrieve the library's current log file path.

- `on2sc Flix2_AddFilter` (`FLIX2PLGNHANDLE *pPlgn`, const `FLIX2HANDLE` `flix`, const char `*plgn_name`)

Add a filter to the encoder's filter chain.

- `on2sc Flix2_RemoveFilter` (`FLIX2PLGNHANDLE` `plgn`)

Remove a filter to the encoder's filter chain.

- `on2sc Flix2_FilterSetParamAsStr` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, const `on2tc *value`)

Set the value of a parameter in a filter instance using a string representation.

- `on2sc Flix2_FilterGetParamAsStr` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, `on2tc *value`, `int32_t *len`)

Retrieve the value of a parameter in a filter instance represented as a string.

- `on2sc Flix2_FilterSetParam` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, double `inDbfVal`)

Set the value of a parameter in a filter instance using a double representation.

- `on2sc Flix2_FilterGetParam` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, double `*outDbfVal`)

Retrieve the value of a parameter in a filter instance represented as a double.

- `on2sc Flix2_AddCodec` (`FLIX2PLGNHANDLE *pPlgn`, const `FLIX2HANDLE` `flix`, const char `*plgn_name`)

Add a codec to the encoder's codec chain.

- `on2sc Flix2_RemoveCodec` (`FLIX2PLGNHANDLE` `plgn`)

Remove a codec from the encoder's codec chain.

- `on2sc Flix2_CodecSetParamAsStr` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, const `on2tc *value`)

Set the value of a parameter in a codec instance using a string representation.

- `on2sc Flix2_CodecGetParamAsStr` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, `on2tc *value`, `int32_t *len`)

Retrieve the value of a parameter in a codec instance represented as a string.

- `on2sc Flix2_CodecSetParam` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, double `inDbfVal`)

Set the value of a parameter in a codec instance using a double representation.

- `on2sc Flix2_CodecGetParam` (`FLIX2PLGNHANDLE` `plgn`, const char `*name`, double `*outDbfVal`)

Retrieve the value of a parameter in a codec instance represented as a double.

- `on2sc Flix2_AddMuxer` (`FLIX2PLGNHANDLE *pPlgn`, const `FLIX2HANDLE` `flix`, const char `*plgn_name`)

Set the desired muxer.

- `on2sc Flix2_RemoveMuxer` (`FLIX2PLGNHANDLE` `plgn`)

Remove the specified muxer.

- [on2sc Flix2_MuxerSetParamAsStr](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, const [on2tc](#) *value)

Set the value of a parameter in a muxer instance using a string representation.

- [on2sc Flix2_MuxerGetParamAsStr](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, [on2tc](#) *value, [int32_t](#) *len)

Retrieve the value of a parameter in a muxer instance represented as a string.

- [on2sc Flix2_MuxerSetParam](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, double inDbfVal)

Set the value of a parameter in a muxer instance using a double representation.

- [on2sc Flix2_MuxerGetParam](#) ([FLIX2PLGNHANDLE](#) plgn, const char *name, double *outDbfVal)

Retrieve the value of a parameter in a muxer instance represented as a double.

- [on2sc Flix2_Errno](#) (const [FLIX2HANDLE](#) flix, [FE2_errno](#) *flixerrno, [int32_t](#) *syserrno)

Retrieve the current error state of the engine.

15.32.1 Detailed Description

Reference module(s): [Main Engine Interface](#), [Filter Manipulation](#), [Codec Manipulation](#), [Muxer Manipulation](#).

Definition in file [flixengine2.h](#).

15.32.2 Define Documentation

15.32.2.1 #define FLIXENGINE_API

Definition at line 24 of file [flixengine2.h](#).

15.32.2.2 #define FLIXENGINE_VERSION_CHIEF

Definition at line 33 of file [flixengine2.h](#).

15.32.2.3 #define FLIXENGINE_VERSION_EXTRA

Definition at line 37 of file [flixengine2.h](#).

15.32.2.4 #define FLIXENGINE_VERSION_MAJOR

Definition at line 34 of file [flixengine2.h](#).

15.32.2.5 #define FLIXENGINE_VERSION_MINOR

Definition at line 35 of file [flixengine2.h](#).

15.32.2.6 #define FLIXENGINE_VERSION_PATCH

Definition at line 36 of file flixengine2.h.

15.32.2.7 #define FLIXENGINE_VERSION_STR

HIDDEN

Definition at line 43 of file flixengine2.h.

15.32.3 Typedef Documentation

15.32.3.1 typedef void* FLIX2HANDLE

handle type passed to all [Flix Engine API](#) functions

Definition at line 50 of file flixengine2.h.

15.32.3.2 typedef void * FLIX2PLGNHANDLE

filter plgn handle passed to [Filter Manipulation](#) functions

Definition at line 50 of file flixengine2.h.

15.32.4 Function Documentation

15.32.4.1 on2sc Flix2_GetLogLevel (const FLIX2HANDLE *flix*, int32_t * *level*)

Retrieve current library-wide debug log level.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#)

→ *level* Storage location to receive the result

Return values:

[ON2_OK](#) on success

[ON2_INVALID_PARAMS](#) should one or more of the preconditions fail

Precondition:

flix is not NULL

level is not NULL

15.32.4.2 on2sc Flix2_GetLogPath (const FLIX2HANDLE *flix*, on2tc * *logpath*, int32_t * *len*)

Retrieve the library's current log file path.

Parameters:

← *flix* Handle to the flix engine returned from [Flix2_Create\(\)](#)

→ *logpath* Storage location for the log file path

↔ *len* on input, the max number of bytes available in *logpath*. On return, the number of bytes used.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flx is not NULL

len is not NULL

15.32.4.3 on2sc Flix2_SetLogLevel (FLIX2HANDLE *flx*, int32_t *level*)

Set the library-wide debug log level.

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#)

← *level* The log level to set the library to

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flx is not NULL

level is within the range 0-4

Remarks:

Valid values for *level* are:

- 0= none (Default)
- 1= informational
- 2= errors+asserts
- 3= debug
- 4= heavy/program trace logging

15.32.4.4 on2sc Flix2_SetLogPath (FLIX2HANDLE *flx*, on2tc * *logpath*)

Set the library's log file path. The log file is opened for appending whenever a log message needs to be written and closed immediately thereafter.

Parameters:

← *flx* Handle to the flx engine returned from [Flix2_Create\(\)](#)

← *logpath* Path to the log file

Return values:

ON2_OK the engine successfully set the log file

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for *logpath*

Precondition:

flix is not NULL

logpath is not NULL

Note:

No attempt is made to ensure the file can be created by this function, nor will it create any non-existent directories in *logpath*.

Remarks:

logpath can be the name of a console, e.g. CONOUT\$

Attention:

Currently the *logpath* and log level, set via [flixengine_com::IFlix::setLogLevel\(\)](#), are global properties. Log messages will be prepended with the thread id of origin for disambiguation.

15.33 flxengine2/media_editor_options.h File Reference

Reference module(s): [Filter Manipulation](#), [Video Filters](#).

Functions

- [on2sc editor_options_Reset](#) (FLIX2HANDLE flx)
Reset the media editor options to their defaults.
- [on2sc editor_options_Validate](#) (const FLIX2HANDLE flx)
Ensure the current media editor settings are valid.

Deprecated functions

- [on2sc editor_options_GetCrop](#) (const FLIX2HANDLE flx, on2bool *pCrop)
Determine if the crop filter is enabled.
- [on2sc editor_options_SetCrop](#) (FLIX2HANDLE flx, const on2bool crop)
Enable/disable the crop filter.
- [on2sc editor_options_GetCropBounds](#) (const FLIX2HANDLE flx, int32_t *pTop, int32_t *pLeft, int32_t *pBottom, int32_t *pRight)
Get the current bounding box used for cropping.
- [on2sc editor_options_SetCropBounds](#) (FLIX2HANDLE flx, const int32_t top, const int32_t left, const int32_t bottom, const int32_t right)
Set the current bounding box used for cropping.
- [on2sc editor_options_GetBrightness](#) (const FLIX2HANDLE flx, int32_t *pBrightness)
Get the current brightness adjustment factor.
- [on2sc editor_options_SetBrightness](#) (FLIX2HANDLE flx, const int32_t brightness)
Set the brightness adjustment factor.
- [on2sc editor_options_GetUseBrightness](#) (const FLIX2HANDLE flx, on2bool *pUseBrightness)
Determine if the brightness filter is enabled.
- [on2sc editor_options_SetUseBrightness](#) (FLIX2HANDLE flx, const on2bool bUseBrightness)
Enable/disable the brightness filter.
- [on2sc editor_options_GetContrast](#) (const FLIX2HANDLE flx, double *pContrast)
Get the current contrast adjustment factor.
- [on2sc editor_options_SetContrast](#) (FLIX2HANDLE flx, const double contrast)
Set the contrast adjustment factor.
- [on2sc editor_options_GetUseContrast](#) (const FLIX2HANDLE flx, on2bool *pUseContrast)
Determine if the contrast filter is enabled.
- [on2sc editor_options_SetUseContrast](#) (FLIX2HANDLE flx, const on2bool bUseContrast)
Enable/disable the contrast filter.

- [on2sc editor_options_GetHue](#) (const [FLIX2HANDLE](#) flix, [int32_t](#) *pHue)
Get the current hue adjustment factor.
- [on2sc editor_options_SetHue](#) ([FLIX2HANDLE](#) flix, const [int32_t](#) hue)
Set the hue adjustment factor.
- [on2sc editor_options_GetUseHue](#) (const [FLIX2HANDLE](#) flix, [on2bool](#) *pUseHue)
Determine if the hue filter is enabled.
- [on2sc editor_options_SetUseHue](#) ([FLIX2HANDLE](#) flix, const [on2bool](#) bUseHue)
Enable/disable the hue filter.
- [on2sc editor_options_GetSaturation](#) (const [FLIX2HANDLE](#) flix, double *pSaturation)
Get the current saturation adjustment factor.
- [on2sc editor_options_SetSaturation](#) ([FLIX2HANDLE](#) flix, const double saturation)
Set the saturation adjustment factor.
- [on2sc editor_options_GetUseSaturation](#) (const [FLIX2HANDLE](#) flix, [on2bool](#) *pUseSaturation)
Determine if the saturation filter is enabled.
- [on2sc editor_options_SetUseSaturation](#) ([FLIX2HANDLE](#) flix, const [on2bool](#) bUseSaturation)
Enable/disable the saturation filter.
- [on2sc editor_options_GetUseCut](#) (const [FLIX2HANDLE](#) flix, [on2bool](#) *pUseCut)
Determine if the cut filter is enabled.
- [on2sc editor_options_SetUseCut](#) ([FLIX2HANDLE](#) flix, const [on2bool](#) bUseCut)
Enable/disable the cut filter.
- [on2sc editor_options_GetCutStartTime](#) (const [FLIX2HANDLE](#) flix, double *pStartTime)
Get the current cut start time.
- [on2sc editor_options_SetCutStartTime](#) ([FLIX2HANDLE](#) flix, const double start_time)
Set the cut start time.
- [on2sc editor_options_GetCutStopTime](#) (const [FLIX2HANDLE](#) flix, double *pEndTime)
Get the current cut stop time.
- [on2sc editor_options_SetCutStopTime](#) ([FLIX2HANDLE](#) flix, const double end_time)
Set the cut stop time.

15.33.1 Detailed Description

Reference module(s): [Filter Manipulation](#), [Video Filters](#).

Definition in file [media_editor_options.h](#).

15.34 flxengine2/muxer_constants.h File Reference

Reference module(s): [Muxers](#). #include "muxers/flv.h"

```
#include "muxers/fxm.h"
```

```
#include "muxers/isomedia.h"
```

```
#include "muxers/swf.h"
```

```
#include "muxers/webm.h"
```

15.34.1 Detailed Description

Reference module(s): [Muxers](#).

Definition in file [muxer_constants.h](#).

15.35 flixengine2/muxers/flv.h File Reference

Reference module(s): [FLV Muxer](#).

Defines

- #define [FE2_MUXER_FLV](#)
FLV muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_FLV_CUEPT_EVENT](#)
Set an event cue point.
- #define [FE2_FLV_CUEPT_NAV](#)
Set a navigation cue point.
- #define [FE2_FLV_CUEPT_PARAM](#)
Add a name/value pair to an existing cue point.
- #define [FE2_FLV_METADATA_ENABLE](#)
Enable output of meta data element.
- #define [FE2_FLV_METADATA_DISABLE](#)
Disable output of meta data element.

Typedefs

- typedef enum [flv_metadata](#) [flvmetadata_t](#)
Supported FLV onMetaData elements.

Enumerations

- enum [flv_metadata](#) {
 [MD_DURATION](#),
 [MD_DATASIZE](#),
 [MD_AUDIO_SIZE](#),
 [MD_VIDEO_SIZE](#),
 [MD_AUDIO_DATARATE](#),
 [MD_VIDEO_DATARATE](#),
 [MD_AUDIO_CODECID](#),
 [MD_VIDEO_CODECID](#),
 [MD_WIDTH](#),
 [MD_HEIGHT](#),
 [MD_FRAMERATE](#),
 [MD_CANSEEKTOEND](#),
};

```
MD_LASTTIMESTAMP,  
MD_LASTKEYFRAMETIMESTAMP,  
MD_LASTKEYFRAMELOCATION,  
MD_KEYFRAMES }
```

Supported FLV onMetaData elements.

15.35.1 Detailed Description

Reference module(s): [FLV Muxer](#).

Definition in file [flv.h](#).

15.36 flixengine2/muxers/fxm.h File Reference

Reference module(s): [FXM Muxer](#).

Defines

- #define [FE2_MUXER_FXM](#)
FXM muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_FXM_CUEPT_EVENT](#)
Set an event cue point.
- #define [FE2_FXM_CUEPT_NAV](#)
Set a navigation cue point.
- #define [FE2_FXM_CUEPT_PARAM](#)
Add a name/value pair to an existing cue point.
- #define [FE2_FXM_METADATA_ENABLE](#)
Enable output of meta data element.
- #define [FE2_FXM_METADATA_DISABLE](#)
Disable output of meta data element.

Typedefs

- typedef enum [flv_metadata fxmmetadata_t](#)

15.36.1 Detailed Description

Reference module(s): [FXM Muxer](#).

Definition in file [fxm.h](#).

15.37 flxengine2/muxers/isomedia.h File Reference

Reference module(s): [3GPP Muxer](#), [3GPP2 Muxer](#), [MOV Muxer](#), [MP4 Muxer](#).

Defines

- #define [FE2_ISOMEDIA_FASTSTART](#)
Influence 'moov' box placement.
- #define [FE2_MUXER_3GP](#)
3GPP muxer. For use with [Flix2_AddMuxer\(\)](#)
- #define [FE2_3GP_FASTSTART](#)
[FE2_ISOMEDIA_FASTSTART](#) alias
- #define [FE2_MUXER_3G2](#)
3GPP2 muxer. For use with [Flix2_AddMuxer\(\)](#)
- #define [FE2_3G2_FASTSTART](#)
[FE2_ISOMEDIA_FASTSTART](#) alias
- #define [FE2_MUXER_MOV](#)
MOV muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_MOV_FASTSTART](#)
[FE2_ISOMEDIA_FASTSTART](#) alias
- #define [FE2_MUXER_MP4](#)
MP4 muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_MP4_FASTSTART](#)
[FE2_ISOMEDIA_FASTSTART](#) alias

15.37.1 Detailed Description

Reference module(s): [3GPP Muxer](#), [3GPP2 Muxer](#), [MOV Muxer](#), [MP4 Muxer](#).

Definition in file [isomedia.h](#).

15.37.2 Define Documentation

15.37.2.1 #define FE2_ISOMEDIA_FASTSTART

Influence 'moov' box placement. When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 34 of file isomedia.h.

15.38 flxengine2/muxers/swf.h File Reference

Reference module(s): [SWF Muxer](#).

Defines

- #define [FE2_MUXER_SWF](#)
SWF muxer. For use with [Flix2_AddMuxer\(\)](#).
- #define [FE2_SWF_HEIGHT](#)
Set the SWF height.
- #define [FE2_SWF_WIDTH](#)
Set the SWF width.
- #define [FE2_SWF_FRAMERATE](#)
Set the SWF framerate.
- #define [FE2_SWF_EMBEDDED_URL](#)
Set the SWF's target URL.
- #define [FE2_SWF_EMBEDDED_URL_TARGET](#)
Set the target of [FE2_SWF_EMBEDDED_URL](#).
- #define [FE2_SWF_EMBEDDED_URL_TYPE](#)
Set how [FE2_SWF_EMBEDDED_URL](#) is interpreted.
- #define [FE2_SWF_LOOP_COUNT](#)
Sets the number of times the SWF should loop.
- #define [FE2_SWF_PRELOAD_TYPE](#)
Sets the type of preloader.
- #define [FE2_SWF_FIXED_PRELOAD_PCT](#)
Sets the percent of the SWF movie to preload before playback begins.
- #define [FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR](#)
Sets the adaptive preload buffer factor.
- #define [FE2_SWF_ON_END_OPTION](#)
Sets the options for the end of the SWF.
- #define [FE2_SWF_ON_END_URL](#)
Sets the URL a SWF movie will load after the current movie ends.
- #define [FE2_SWF_ON_START_OPTION](#)
Sets the options for the start of the SWF.
- #define [FE2_SWF_START_BLANK_FRAME](#)

Control the insertion of a blank first frame in the SWF.

- `#define FE2_SWF_START_WAIT_SEC`
Sets the number of seconds to wait before playback begins.
- `#define FE2_SWF_ADD_VARIABLE`
Add or update a variable in the SWF.
- `#define FE2_SWF_DELETE_VARIABLE`
Delete an existing variable in the SWF.

Enumerations

- `enum FE2_EmbeddedUrlType {`
 `EmbeddedUrlIsGetUrl,`
 `EmbeddedUrlIsLoadMovie }`
Differentiates between the type of file (HTML or SWF) set through `FE2_SWF_EMBEDDED_URL`.
- `enum FE2_SwfOnEndOptions {`
 `SwfOnMovieEndNothing,`
 `SwfOnMovieEndSTOP,`
 `SwfOnMovieEndLoop,`
 `SwfOnMovieEndUnload,`
 `SwfOnMovieEndLoadMovie }`
Actions that can be added to the last frame of a SWF file.
- `enum FE2_SwfOnStartOptions {`
 `SwfOnMovieStartAutomatically,`
 `SwfOnMovieStartOnClick,`
 `SwfOnMovieStartWait,`
 `SwfOnMovieStartEmbedSTOP }`
Actions that can be added to the start frame of the SWF file.
- `enum FE2_SwfPreloaderOptions {`
 `SwfPreloaderNone,`
 `SwfFixedPreloader,`
 `SwfAdaptivePreloader }`
Determines the type of preloader added to the SWF file.

15.38.1 Detailed Description

Reference module(s): [SWF Muxer](#).

Definition in file [swf.h](#).

15.39 flxengine2/muxers/webm.h File Reference

Reference module(s): [WebM Muxer](#).

Defines

- #define [FE2_MUXER_WEBM](#)
WebM muxer. For use with [Flix2_AddMuxer\(\)](#).

15.39.1 Detailed Description

Reference module(s): [WebM Muxer](#).

Definition in file [webm.h](#).

15.40 flixengine2/on2types.h File Reference

Reference module(s): [Base Types](#). `#include <sys/types.h>`

`#include <limits.h>`

Defines

- `#define OTC(str)`
a macro suitable for declaring a constant [on2tc](#)
- `#define ON2TC`
printf format string suitable for printing an [on2tc](#)
- `#define INT64_MIN`
- `#define INT64_MAX`
- `#define UINT64_MAX`
- `#define PRId64`
- `#define PRIu64`
- `#define ON264`
printf format string suitable for printing an [on2s64](#)
- `#define DLLIMPORT`
- `#define DLLEXPORT`
- `#define DLLLOCAL`
- `#define ON2API`
library calling convention/storage class attributes.

Typedefs

- `typedef char int8_t`
- `typedef short int16_t`
- `typedef int int32_t`
- `typedef unsigned char uint8_t`
- `typedef unsigned short uint16_t`
- `typedef unsigned int uint32_t`
- `typedef int8_t on2s8`
- `typedef uint8_t on2u8`
- `typedef int16_t on2s16`
- `typedef uint16_t on2u16`
- `typedef int32_t on2s32`
- `typedef uint32_t on2u32`
- `typedef int32_t on2bool`
- `typedef char on2tc`
- `typedef __int64 on2s64`
- `typedef unsigned __int64 on2u64`
- `typedef on2s64 int64_t`
- `typedef on2u64 uint64_t`

Enumerations

- enum [_on2bool](#) {
 [on2false](#),
 [on2true](#) }
- enum [on2sc](#) {
 [ON2_NOT_FOUND](#),
 [ON2_BUFFER_EMPTY](#),
 [ON2_BUFFER_FULL](#),
 [ON2_CONNREFUSED](#),
 [ON2_TIMEDOUT](#),
 [ON2_WOULDBLOCK](#),
 [ON2_NET_ERROR](#),
 [ON2_INVALID_VERSION](#),
 [ON2_INPROGRESS](#),
 [ON2_NOT_SUPP](#),
 [ON2_NO_MEM](#),
 [ON2_INVALID_PARAMS](#),
 [ON2_ERROR](#),
 [ON2_OK](#),
 [ON2_DONE](#) }

Common return type.

15.40.1 Detailed Description

Reference module(s): [Base Types](#).

Definition in file [on2types.h](#).

15.40.2 Define Documentation

15.40.2.1 #define DLLEXPORT

Definition at line 159 of file [on2types.h](#).

15.40.2.2 #define DLLIMPORT

Definition at line 158 of file [on2types.h](#).

15.40.2.3 #define DLLLOCAL

Definition at line 160 of file [on2types.h](#).

15.40.2.4 `#define ON2API`

library calling convention/storage class attributes. Specifies whether the function is imported through a dll or is from a static library.

Definition at line 192 of file on2types.h.

15.41 flixengine2/overlay_options.h File Reference

Reference module(s): [Overlay \(Watermark\)](#).

Functions

Deprecated functions

- [on2sc overlay_options_Reset](#) (const [FLIX2HANDLE](#) flix)
Resets the overlay options.
- [on2sc overlay_options_GetUseOverlay](#) (const [FLIX2HANDLE](#) flix, [on2bool](#) *pUseOverlay)
Determines if an overlay is to be used.
- [on2sc overlay_options_SetUseOverlay](#) ([FLIX2HANDLE](#) flix, const [on2bool](#) bUseOverlay)
Enables or disables overlay usage.
- [on2sc overlay_options_GetOverlayPath](#) (const [FLIX2HANDLE](#) flix, [on2tc](#) *pOverlayFilePath, [int32_t](#) *pLen)
Returns the path to the overlay image file.
- [on2sc overlay_options_SetOverlayPath](#) ([FLIX2HANDLE](#) flix, const [on2tc](#) *pOverlayFilePath)
Set the path to the overlay image file.
- [on2sc overlay_options_GetMaskPixelXY](#) (const [FLIX2HANDLE](#) flix, [int32_t](#) *pMaskPixelX, [int32_t](#) *pMaskPixelY)
Return the X and Y coordinates of the mask pixel.
- [on2sc overlay_options_SetMaskPixelXY](#) ([FLIX2HANDLE](#) flix, [int32_t](#) maskPixelX, [int32_t](#) maskPixelY)
Set the X and Y coordinates of the mask pixel.
- [on2sc overlay_options_GetMaskPixelRGB](#) ([FLIX2HANDLE](#) flix, [uint8_t](#) *pMaskPixelR, [uint8_t](#) *pMaskPixelG, [uint8_t](#) *pMaskPixelB)
Return the Red, Green, and Blue component values of the mask pixel.
- [on2sc overlay_options_SetMaskPixelRGB](#) ([FLIX2HANDLE](#) flix, [uint8_t](#) maskPixelR, [uint8_t](#) maskPixelG, [uint8_t](#) maskPixelB)
Set the Red, Green, and Blue component values of the mask pixel.
- [on2sc overlay_options_GetOverlayPosition](#) ([FLIX2HANDLE](#) flix, [FE2_OverlayPositionMode](#) *pMode, [uint32_t](#) *pX, [uint32_t](#) *pY)
Return the overlay position.
- [on2sc overlay_options_SetOverlayPosition](#) ([FLIX2HANDLE](#) flix, [FE2_OverlayPositionMode](#) mode, [uint32_t](#) x, [uint32_t](#) y)
Set the overlay position.

15.41.1 Detailed Description

Reference module(s): [Overlay \(Watermark\)](#).

Definition in file [overlay_options.h](#).

15.42 flixengine2/swf_options.h File Reference

Reference module(s): [SWF](#).

Functions

Deprecated functions

- [on2sc swf_options_Reset](#) ([FLIX2HANDLE](#) flix)
Resets the swf options.
- [on2sc swf_options_GetEmbeddedUrl](#) ([FLIX2HANDLE](#) flix, char *embeddedUrl, [int32_t](#) *len)
Gets the embedded URL.
- [on2sc swf_options_GetEmbeddedUrlTarget](#) ([FLIX2HANDLE](#) flix, char *embeddedUrlTarget, [int32_t](#) *len)
Gets the target of embedded URL.
- [on2sc swf_options_SetEmbeddedUrl](#) ([FLIX2HANDLE](#) flix, const char *embeddedUrl)
Sets the embedded URL of the SWF or HTML that will be loaded when the user clicks on the video.
- [on2sc swf_options_SetEmbeddedUrlTarget](#) ([FLIX2HANDLE](#) flix, const char *embeddedUrlTarget)
Sets the target of the embedded URL.
- [on2sc swf_options_GetEmbeddedUrlType](#) ([FLIX2HANDLE](#) flix, [FE2_EmbeddedUrlType](#) *embeddedUrlType)
Gets the type of embedded URL.
- [on2sc swf_options_SetEmbeddedUrlType](#) ([FLIX2HANDLE](#) flix, const [FE2_EmbeddedUrlType](#) embeddedUrlType)
Sets the type of embedded URL.
- [on2sc swf_options_GetSwfFramerate](#) ([FLIX2HANDLE](#) flix, [int32_t](#) *pSwfFramerate)
Gets the SWF framerate.
- [on2sc swf_options_SetSwfFramerate](#) ([FLIX2HANDLE](#) flix, const [int32_t](#) swfFramerate)
Sets the SWF framerate.
- [on2sc swf_options_GetInsertBlankFrameOnStart](#) ([FLIX2HANDLE](#) flix, [on2bool](#) *pInsertBlankFrameOnStart)
Gets if the engine is to insert a blank frame as the first frame of the SWF or not.
- [on2sc swf_options_SetInsertBlankFrameOnStart](#) ([FLIX2HANDLE](#) flix, const [on2bool](#) insertBlankFrameOnStart)
Tells the engine to insert a blank frame as the first frame of the SWF or not.
- [on2sc swf_options_GetSwfFramerateAsDouble](#) ([FLIX2HANDLE](#) flix, double *pSwfFramerate)
Gets the SWF framerate as a double.
- [on2sc swf_options_SetSwfFramerateAsDouble](#) ([FLIX2HANDLE](#) flix, const double swfFramerate)
Sets the SWF framerate as a double.

- [on2sc swf_options_GetEnablePreloader](#) (FLIX2HANDLE flx, on2bool *pEnablePreloader)
Gets if a video preloader is enabled or disabled.
- [on2sc swf_options_SetEnablePreloader](#) (FLIX2HANDLE flx, const on2bool enablePreloader)
Enables or disables the video preloader.
- [on2sc swf_options_GetPercentToPreload](#) (FLIX2HANDLE flx, int32_t *pPercentToPreload)
Gets the percent of the SWF movie to preload before playback begins.
- [on2sc swf_options_SetPercentToPreload](#) (FLIX2HANDLE flx, const int32_t percentToPreload)
Sets the percent of the SWF movie to preload before playback begins.
- [on2sc swf_options_GetPreloaderType](#) (FLIX2HANDLE flx, FE2_SwfPreloaderOptions *pPreloaderType)
Gets the type of preloader.
- [on2sc swf_options_SetPreloaderType](#) (FLIX2HANDLE flx, const FE2_SwfPreloaderOptions preloaderType)
Sets the type of preloader.
- [on2sc swf_options_GetAdaptivePreloaderBufferFactor](#) (FLIX2HANDLE flx, double *pPreloaderBufferFactor)
Gets the adaptive preload buffer factor.
- [on2sc swf_options_SetAdaptivePreloaderBufferFactor](#) (FLIX2HANDLE flx, const double preloaderBufferFactor)
Sets the adaptive preload buffer factor.
- [on2sc swf_options_GetMovieOnEndOptions](#) (FLIX2HANDLE flx, FE2_SwfOnEndOptions *pOnEndOptions)
Gets the options for the end of the SWF.
- [on2sc swf_options_SetMovieOnEndOptions](#) (FLIX2HANDLE flx, const FE2_SwfOnEndOptions onEndOptions)
Sets the options for the end of the SWF.
- [on2sc swf_options_GetLoopCount](#) (FLIX2HANDLE flx, int32_t *pLoopCount)
Gets the number of times the SWF should loop.
- [on2sc swf_options_SetLoopCount](#) (FLIX2HANDLE flx, const int32_t loopCount)
Sets the number of times the SWF should loop.
- [on2sc swf_options_GetLoadMovieOnEndUrl](#) (FLIX2HANDLE flx, char *pLoadMovieOnEndUrl, int32_t *len)
Gets the URL of a SWF movie to load after the current movie ends.
- [on2sc swf_options_SetLoadMovieOnEndUrl](#) (FLIX2HANDLE flx, const char *loadMovieOnEndUrl)
Sets the URL of a SWF movie to load after the current movie ends.
- [on2sc swf_options_GetMovieOnStartOptions](#) (FLIX2HANDLE flx, FE2_SwfOnStartOptions *pOnStartOptions)
Gets the options for the start of the SWF.

- `on2sc swf_options_SetMovieOnStartOptions (FLIX2HANDLE flix, const FE2_SwfOnStartOptions onStartOptions)`
Sets the options for the start of the SWF.
- `on2sc swf_options_GetWaitTimeToStart (FLIX2HANDLE flix, int32_t *pWaitTimeToStart)`
Gets the number of seconds to wait before playback begins.
- `on2sc swf_options_SetWaitTimeToStart (FLIX2HANDLE flix, const int32_t waitTimeToStart)`
Sets the number of seconds to wait before playback begins.
- `on2sc swf_options_AddVariable (FLIX2HANDLE flix, const char *name, const char *value)`
Adds a custom SWF variable as a name/value pair.
- `on2sc swf_options_DeleteVariable (FLIX2HANDLE flix, const int32_t index)`
Deletes a custom SWF variable.
- `on2sc swf_options_GetVariableCount (FLIX2HANDLE flix, int32_t *pVariableCount)`
Gets the number of custom SWF variables already added.
- `on2sc swf_options_UpdateVariable (FLIX2HANDLE flix, const int32_t index, const char *name, const char *value)`
Updates an already existing SWF variable.

15.42.1 Detailed Description

Reference module(s): [SWF](#).

Definition in file [swf_options.h](#).

15.43 flxengine2/video_options.h File Reference

Reference module(s): [Video Encoding Options](#), [Frame Rate](#), [Scale](#).

Enumerations

- enum [FE2_VideoCodec](#) {
[CODEC_NULL](#),
[CODEC_H263](#),
[CODEC_SCREENVIDEO](#),
[CODEC_VP6](#),
[CODEC_VP6ALPHA](#) }
Output video codec types, influences quality/compatibility.
- enum [FE2_CuePointType](#) {
[CUE_EVENT](#),
[CUE_NAVIGATION](#) }
Cue point type for use with [video_options_AddFLVCuePoint\(\)](#).

Functions

- [on2sc video_options_Reset](#) ([FLIX2HANDLE](#) flx)
Reset the video options to their defaults.
- [on2sc video_options_Validate](#) (const [FLIX2HANDLE](#) flx)
Ensure the current video settings are valid.
- [on2sc video_options_GetImageQuality](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpImageQuality)
Get the current image quality factor.
- [on2sc video_options_SetImageQuality](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lImageQuality)
Set the image quality factor.
- [on2sc video_options_GetKeyframeInterval](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpKeyframeInterval)
Get the current interval between keyframes.
- [on2sc video_options_SetKeyframeInterval](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lKeyframeInterval)
Set the interval between keyframes.
- [on2sc video_options_GetKeyframeIntervalType](#) (const [FLIX2HANDLE](#) flx, [FE2_VideoKeyframeTypes](#) *pKeyframeIntervalType)
Get the current keyframe interval type.
- [on2sc video_options_SetKeyframeIntervalType](#) ([FLIX2HANDLE](#) flx, const [FE2_VideoKeyframeTypes](#) keyframeIntervalType)

Set the keyframe interval type.

- `on2sc video_options_GetMaximumBitrate` (const `FLIX2HANDLE` `flx`, `int32_t` `*lpMaximumBitrate`)

Get the current maximum bitrate target.

- `on2sc video_options_SetMaximumBitrate` (`FLIX2HANDLE` `flx`, const `int32_t` `lMaximumBitrate`)

Set the maximum bitrate target.

- `on2sc video_options_GetRateControlType` (const `FLIX2HANDLE` `flx`, `FE2_VideoBitrateControls` `*pRateControlType`)

Get the current rate control type.

- `on2sc video_options_SetRateControlType` (`FLIX2HANDLE` `flx`, const `FE2_VideoBitrateControls` `rateControlType`)

Set the rate control type.

- `on2sc video_options_GetSwfHeight` (const `FLIX2HANDLE` `flx`, `int32_t` `*lpSwfHeight`)

Gets the SWF height.

- `on2sc video_options_SetSwfHeight` (`FLIX2HANDLE` `flx`, const `int32_t` `lSwfHeight`)

Sets the SWF height.

- `on2sc video_options_GetSwfWidth` (const `FLIX2HANDLE` `flx`, `int32_t` `*lpSwfWidth`)

Gets the SWF width.

- `on2sc video_options_SetSwfWidth` (`FLIX2HANDLE` `flx`, const `int32_t` `lSwfWidth`)

Sets the SWF width.

- `on2sc video_options_GetUseCustomSwfDimensions` (const `FLIX2HANDLE` `flx`, `on2bool` `*bpUseCustomSwfDimensions`)

Determine if the engine is using the SWF width and height for the SWF.

- `on2sc video_options_SetUseCustomSwfDimensions` (`FLIX2HANDLE` `flx`, const `on2bool` `bUseCustomSwfDimensions`)

Tells the engine to use the SWF width and SWF height for the SWF.

- `on2sc video_options_GetUseMaximumBitrate` (const `FLIX2HANDLE` `flx`, `on2bool` `*bpUseMaximumBitrate`)

Determine if the maximum bitrate target will be used.

- `on2sc video_options_SetUseMaximumBitrate` (`FLIX2HANDLE` `flx`, const `on2bool` `bUseMaximumBitrate`)

Enable/disable the maximum bitrate target value.

- `on2sc video_options_GetSwfFramerate` (const `FLIX2HANDLE` `flx`, `int32_t` `*pSwfFramerate`)

Gets the SWF framerate.

- `on2sc video_options_SetSwfFramerate` (`FLIX2HANDLE` `flx`, const `int32_t` `swfFramerate`)

Sets the SWF framerate.

- `on2sc video_options_GetVideoCodec` (const `FLIX2HANDLE` flx, `FE2_VideoCodec` *pVideoCodec)
Determine which video codec will be used.
- `on2sc video_options_SetVideoCodec` (`FLIX2HANDLE` flx, const `FE2_VideoCodec` videoCodec)
Select the video codec to be used.
- `on2sc video_options_GetAlphaPercentage` (const `FLIX2HANDLE` flx, `int32_t` *percentage)
Gets the percentage of the video bitrate to be used to encode the alpha channel.
- `on2sc video_options_SetAlphaPercentage` (`FLIX2HANDLE` flx, const `int32_t` percentage)
Sets the percentage of the video bitrate to be used to encode the alpha channel.
- `on2sc video_options_GetSourceHeight` (const `FLIX2HANDLE` flx, `int32_t` *height)
Get the height of the source video.
- `on2sc video_options_GetSourceWidth` (const `FLIX2HANDLE` flx, `int32_t` *width)
Get the width of the source video.
- `on2sc video_options_AddFLVCuePoint` (`FLIX2HANDLE` flx, const char *pName, const double time, const `FE2_CuePointType` type)
Adds a cue point to the FLV.
- `on2sc video_options_AddFLVCuePointParameter` (`FLIX2HANDLE` flx, const char *pCuePointName, const char *pName, const char *pValue)
Adds a name/value parameter to an already added cue point.
- `on2sc video_options_GetCompressMode` (const `FLIX2HANDLE` flx, `FE2_CompressMode` *mode)
Get the current VP6 compression mode.
- `on2sc video_options_SetCompressMode` (`FLIX2HANDLE` flx, `FE2_CompressMode` mode)
Set the VP6 compression mode.

Deprecated functions

- `on2sc video_options_GetImageHeight` (const `FLIX2HANDLE` flx, `int32_t` *lpImageHeight)
Get the current scaled image height.
- `on2sc video_options_SetImageHeight` (`FLIX2HANDLE` flx, const `int32_t` lImageHeight)
Set the scaled image height.
- `on2sc video_options_GetImageWidth` (const `FLIX2HANDLE` flx, `int32_t` *lpImageWidth)
Get the current scaled image width.
- `on2sc video_options_SetImageWidth` (`FLIX2HANDLE` flx, const `int32_t` lImageWidth)
Set the scaled image width.
- `on2sc video_options_GetDeinterlace` (const `FLIX2HANDLE` flx, `on2bool` *lpDeinterlace)
Determine if the deinterlace filter is enabled.

- [on2sc video_options_SetDeinterlace](#) ([FLIX2HANDLE](#) flx, const [on2bool](#) lDeinterlace)
Enable/disable the deinterlace filter.
- [on2sc video_options_GetVideoFramerate](#) (const [FLIX2HANDLE](#) flx, [int32_t](#) *lpVideoFramerate)
Get the current video framerate.
- [on2sc video_options_SetVideoFramerate](#) ([FLIX2HANDLE](#) flx, const [int32_t](#) lVideoFramerate)
Set the video framerate.
- [on2sc video_options_GetUseSourceDimensions](#) (const [FLIX2HANDLE](#) flx, [on2bool](#) *bpUseSourceDimensions)
Determine if the source's dimensions will be used instead of the scaled dimensions.
- [on2sc video_options_SetUseSourceDimensions](#) ([FLIX2HANDLE](#) flx, const [on2bool](#) bUseSourceDimensions)
Switch between source and scaled dimensions.
- [on2sc video_options_GetVideoFramerateAsDouble](#) (const [FLIX2HANDLE](#) flx, double *p_fps)
Get the current video framerate.
- [on2sc video_options_SetVideoFramerateAsDouble](#) ([FLIX2HANDLE](#) flx, const double fps)
Set the video framerate.
- [on2sc video_options_GetUseSourceFramerate](#) ([FLIX2HANDLE](#) flx, [on2bool](#) *bpUseSourceFramerate)
Determine if the source's framerate will be used instead of the modified framerate.
- [on2sc video_options_SetUseSourceFramerate](#) ([FLIX2HANDLE](#) flx, const [on2bool](#) bUseSourceFramerate)
Switch between source and scaled framerate.
- [on2sc video_options_GetDecimateValue](#) (const [FLIX2HANDLE](#) flx, [uint32_t](#) *pValue)
Get the current decimation of the video framerate.
- [on2sc video_options_SetDecimateValue](#) ([FLIX2HANDLE](#) flx, const [uint32_t](#) value)
Set the decimation of the video framerate.

15.43.1 Detailed Description

Reference module(s): [Video Encoding Options](#), [Frame Rate](#), [Scale](#).

Definition in file [video_options.h](#).

15.44 flixengine_com.idl File Reference

Reference module(s): [Flix Engine COM](#).

Data Structures

- interface [flixengine_com::IFlixPlgn](#)
Interface for accessing [Codec](#), [Filter](#) and [Muxer](#) functions.
- interface [flixengine_com::IEncodingStatus](#)
Interface for accessing [encoding statistics](#) functions.
- interface [flixengine_com::ISwfOptions](#)
Interface for accessing [SWF creation](#) functions.
- interface [flixengine_com::IVideoOptions](#)
Interface for accessing non-deprecated [video_options_](#) functions.*
- interface [flixengine_com::IFlix](#)
Main interface for accessing [engine](#) functions and obtaining instances of other COM interfaces.

Namespaces

- namespace [flixengine_com](#)

Functions

- `midl_pragma warning (disable:2111 2362) import"oidl.idl"`

15.44.1 Detailed Description

Reference module(s): [Flix Engine COM](#).

Definition in file [flixengine_com.idl](#).

15.44.2 Function Documentation

15.44.2.1 `midl_pragma warning (disable:2111 2362)`

15.45 `mainpage.dox` File Reference

15.46 migration.dox File Reference

15.47 samples.dox File Reference

Index

- [_on2bool](#)
 - [basetypes, 557](#)
 - [3GPP - FFmpeg, 513](#)
 - [3GPP2 - FFmpeg, 515](#)
 - [AAC, 337](#)
 - [AAC+, 339](#)
 - [about.dox, 641](#)
 - [addCodec](#)
 - [flxengine_com::IFlix, 600](#)
 - [addFilter](#)
 - [flxengine_com::IFlix, 600](#)
 - [addFLVCuePoint](#)
 - [flxengine_com::IVideoOptions, 636](#)
 - [addFLVCuePointParameter](#)
 - [flxengine_com::IVideoOptions, 636](#)
 - [addMuxer](#)
 - [flxengine_com::IFlix, 600](#)
 - [addVariable](#)
 - [flxengine_com::ISwfOptions, 628](#)
 - [AMR_NB - FFmpeg, 341](#)
 - [Audio Codecs, 336](#)
 - [Audio Encoding Options, 323](#)
 - [Audio Filters, 388](#)
 - [audio_codecs](#)
 - [FE2_CODEC_PCM, 336](#)
 - [audio_options_GetBitrate](#)
 - [audopts_deprecated, 326](#)
 - [audio_options_GetFlvAudioFormat](#)
 - [audopts_deprecated, 327](#)
 - [audio_options_GetSamplingrate](#)
 - [audopts_deprecated, 327](#)
 - [audio_options_GetStereo](#)
 - [audopts_deprecated, 328](#)
 - [audio_options_Reset](#)
 - [audopts_deprecated, 328](#)
 - [audio_options_SetBitrate](#)
 - [audopts_deprecated, 329](#)
 - [audio_options_SetFlvAudioFormat](#)
 - [audopts_deprecated, 329](#)
 - [audio_options_SetSamplingrate](#)
 - [audopts_deprecated, 330](#)
 - [audio_options_SetStereo](#)
 - [audopts_deprecated, 330](#)
 - [audio_options_Validate](#)
 - [audopts_deprecated, 331](#)
- [audopts](#)
 - [Bitrate112kbps, 324](#)
 - [Bitrate128kbps, 324](#)
 - [Bitrate144kbps, 324](#)
 - [Bitrate160kbps, 324](#)
 - [Bitrate16kbps, 324](#)
 - [Bitrate192kbps, 324](#)
 - [Bitrate224kbps, 324](#)
 - [Bitrate24kbps, 324](#)
 - [Bitrate256kbps, 324](#)
 - [Bitrate320kbps, 324](#)
 - [Bitrate32kbps, 324](#)
 - [Bitrate40kbps, 324](#)
 - [Bitrate48kbps, 324](#)
 - [Bitrate56kbps, 324](#)
 - [Bitrate64kbps, 324](#)
 - [Bitrate80kbps, 324](#)
 - [Bitrate8kbps, 324](#)
 - [Bitrate96kbps, 324](#)
 - [FE2_AudioBitrates, 324](#)
 - [audopts_deprecated](#)
 - [FlvAudioMp3, 326](#)
 - [FlvAudioUncompressed, 326](#)
 - [Hertz11025, 326](#)
 - [Hertz22050, 326](#)
 - [Hertz44100, 326](#)
 - [audopts_deprecated](#)
 - [audio_options_GetBitrate, 326](#)
 - [audio_options_GetFlvAudioFormat, 327](#)
 - [audio_options_GetSamplingrate, 327](#)
 - [audio_options_GetStereo, 328](#)
 - [audio_options_Reset, 328](#)
 - [audio_options_SetBitrate, 329](#)
 - [audio_options_SetFlvAudioFormat, 329](#)
 - [audio_options_SetSamplingrate, 330](#)
 - [audio_options_SetStereo, 330](#)
 - [audio_options_Validate, 331](#)
 - [FE2_AudioSamplingrates, 326](#)
 - [FE2_FlvAudioFormat, 326](#)
 - [averageBitrate](#)
 - [flxengine_com::IEncodingStatus, 584](#)
 - [averageFramesize](#)
 - [flxengine_com::IEncodingStatus, 584](#)

- Base Types, [554](#)
- BASE_H264PROFILE
 - codec_h264, [349](#)
- basetypes
 - _on2bool, [557](#)
 - int16_t, [556](#)
 - int32_t, [556](#)
 - INT64_MAX, [555](#)
 - INT64_MIN, [555](#)
 - int64_t, [556](#)
 - int8_t, [556](#)
 - ON264, [555](#)
 - ON2_BUFFER_EMPTY, [558](#)
 - ON2_BUFFER_FULL, [558](#)
 - ON2_CONNREFUSED, [558](#)
 - ON2_DONE, [558](#)
 - ON2_ERROR, [558](#)
 - ON2_INPROGRESS, [558](#)
 - ON2_INVALID_PARAMS, [558](#)
 - ON2_INVALID_VERSION, [558](#)
 - ON2_NET_ERROR, [558](#)
 - ON2_NO_MEM, [558](#)
 - ON2_NOT_FOUND, [558](#)
 - ON2_NOT_SUPP, [558](#)
 - ON2_OK, [558](#)
 - ON2_TIMEDOUT, [558](#)
 - ON2_WOULD_BLOCK, [558](#)
 - on2bool, [556](#)
 - on2false, [557](#)
 - on2s16, [556](#)
 - on2s32, [556](#)
 - on2s64, [556](#)
 - on2s8, [556](#)
 - on2sc, [558](#)
 - ON2TC, [555](#)
 - on2tc, [557](#)
 - on2true, [557](#)
 - on2u16, [557](#)
 - on2u32, [557](#)
 - on2u64, [557](#)
 - on2u8, [557](#)
 - OTC, [555](#)
 - PRId64, [555](#)
 - PRId64, [556](#)
 - uint16_t, [557](#)
 - uint32_t, [557](#)
 - UINT64_MAX, [556](#)
 - uint64_t, [557](#)
 - uint8_t, [557](#)
- Bitrate112kbps
 - audopts, [324](#)
- Bitrate128kbps
 - audopts, [324](#)
- Bitrate144kbps
 - audopts, [324](#)
- Bitrate160kbps
 - audopts, [324](#)
- Bitrate16kbps
 - audopts, [324](#)
- Bitrate192kbps
 - audopts, [324](#)
- Bitrate224kbps
 - audopts, [324](#)
- Bitrate24kbps
 - audopts, [324](#)
- Bitrate256kbps
 - audopts, [324](#)
- Bitrate320kbps
 - audopts, [324](#)
- Bitrate32kbps
 - audopts, [324](#)
- Bitrate40kbps
 - audopts, [324](#)
- Bitrate48kbps
 - audopts, [324](#)
- Bitrate56kbps
 - audopts, [324](#)
- Bitrate64kbps
 - audopts, [324](#)
- Bitrate80kbps
 - audopts, [324](#)
- Bitrate8kbps
 - audopts, [324](#)
- Bitrate96kbps
 - audopts, [324](#)
- Blur, [402](#)
- BLUR_GAUSS
 - filter_blur, [404](#)
- BLUR_LOWPASS
 - filter_blur, [403](#)
- blurfilter
 - filter_blur, [403](#)
- blurfilter_t
 - filter_blur, [403](#)
- Brightness/Contrast/Hue/Saturation, [392](#)
- CBR_1PASSControl
 - codecpam_common, [334](#)
- CBR_2PASSControl
 - codecpam_common, [334](#)
- Codec Manipulation, [495](#)
- CODEC_H263
 - vidopts_deprecated, [566](#)
- codec_h264
 - BASE_H264PROFILE, [349](#)
 - HIGH_H264PROFILE, [349](#)
 - MAIN_H264PROFILE, [349](#)
- codec_lame

- LAME_ABR, [352](#)
- LAME_CBR, [352](#)
- LAME_VBR_mtrh, [352](#)
- LAME_VBR_rh, [352](#)
- CODEC_NULL
 - vidopts_deprecated, [566](#)
- CODEC_SCREENVIDEO
 - vidopts_deprecated, [566](#)
- CODEC_VP6
 - vidopts_deprecated, [566](#)
- codec_vp6
 - COMPRESSMODE_BEST, [363](#)
 - COMPRESSMODE_GOOD, [363](#)
 - VP6_E, [363](#)
 - VP6_S, [363](#)
- CODEC_VP6ALPHA
 - vidopts_deprecated, [566](#)
- codec_aac
 - FE2_AAC_BITRATE, [338](#)
 - FE2_CODEC_AAC, [338](#)
- codec_aacplus
 - FE2_AACPLUS_BITRATE, [340](#)
 - FE2_AACPLUS_PARAMETRIC_STEREO, [340](#)
 - FE2_CODEC_AACPLUS, [340](#)
- codec_amrlavc
 - FE2_AMR_BITRATE, [342](#)
 - FE2_CODEC_AMR_NB, [342](#)
- codec_h263
 - FE2_CODEC_H263, [343](#)
 - FE2_CODEC_H263_BASELINE, [343](#)
 - FE2_H263_BITRATE, [344](#)
 - FE2_H263_KFFREQ, [344](#)
 - FE2_H263_KFINTTYPE, [344](#)
 - FE2_H263_MAX_Q, [345](#)
 - FE2_H263_MIN_Q, [345](#)
 - FE2_H263_RC_MODE, [345](#)
- codec_h264
 - FE2_CODEC_H264, [347](#)
 - FE2_H264_B_FRAME_RATE, [347](#)
 - FE2_H264_BITRATE, [347](#)
 - FE2_H264_KFFREQ, [348](#)
 - FE2_H264_KFINTTYPE, [348](#)
 - FE2_H264_PROFILE, [348](#)
 - FE2_H264_RC_MODE, [348](#)
 - FE2_H264_SPEED, [349](#)
 - h264profile, [349](#)
 - h264profile_t, [349](#)
- codec_lame
 - FE2_CODEC_LAME, [351](#)
 - FE2_LAME_BITRATE, [351](#)
 - FE2_LAME_CHANNELS, [351](#)
 - FE2_LAME_QUALITY, [351](#)
 - FE2_LAME_RC_MODE, [352](#)
 - lame_rcmode, [352](#)
 - lame_rcmode_t, [352](#)
- codec_vorbislavc
 - FE2_CODEC_VORBIS, [353](#)
 - FE2_VORBIS_BITRATE, [353](#)
- codec_vp6
 - FE2_CODEC_VP6, [357](#)
 - FE2_CompressMode, [363](#)
 - FE2_VP6_2PASS_MAX_SECTION, [357](#)
 - FE2_VP6_2PASS_MIN_SECTION, [357](#)
 - FE2_VP6_BITRATE, [357](#)
 - FE2_VP6_CONCURRENCY, [357](#)
 - FE2_VP6_CXMODE, [358](#)
 - FE2_VP6_KFFREQ, [359](#)
 - FE2_VP6_KFINTTYPE, [359](#)
 - FE2_VP6_MAX_Q, [359](#)
 - FE2_VP6_MIN_Q, [359](#)
 - FE2_VP6_NOISE_REDUCTION, [360](#)
 - FE2_VP6_PROFILE, [360](#)
 - FE2_VP6_RC_MODE, [360](#)
 - FE2_VP6_SHARPNESS, [360](#)
 - FE2_VP6_STREAM_MAX_BUFFER, [361](#)
 - FE2_VP6_STREAM_OPTIMAL_BUFFER, [361](#)
 - FE2_VP6_STREAM_PEAK_BITRATE, [361](#)
 - FE2_VP6_STREAM_PREBUFFER, [361](#)
 - FE2_VP6_TEMPORAL_DOWN_-_WATERMARK, [362](#)
 - FE2_VP6_TEMPORAL_RESAMPLING, [362](#)
 - FE2_VP6_UNDERSHOOT_PCT, [362](#)
 - vp6profile, [363](#)
 - vp6profile_t, [362](#)
- codec_vp6a
 - FE2_CODEC_VP6ALPHA, [366](#)
 - FE2_VP6A_2PASS_MAX_SECTION, [366](#)
 - FE2_VP6A_2PASS_MIN_SECTION, [366](#)
 - FE2_VP6A_ALPHA_BITRATE, [366](#)
 - FE2_VP6A_ALPHA_MAX_Q, [366](#)
 - FE2_VP6A_ALPHA_MIN_Q, [366](#)
 - FE2_VP6A_ALPHA_NOISE_REDUCTION, [367](#)
 - FE2_VP6A_ALPHA_SHARPNESS, [367](#)
 - FE2_VP6A_BITRATE, [367](#)
 - FE2_VP6A_CXMODE, [367](#)
 - FE2_VP6A_KFFREQ, [367](#)
 - FE2_VP6A_KFINTTYPE, [368](#)
 - FE2_VP6A_MAX_Q, [368](#)
 - FE2_VP6A_MIN_Q, [368](#)
 - FE2_VP6A_NOISE_REDUCTION, [369](#)
 - FE2_VP6A_RC_MODE, [369](#)
 - FE2_VP6A_SHARPNESS, [369](#)
 - FE2_VP6A_STREAM_MAX_BUFFER, [369](#)
 - FE2_VP6A_STREAM_OPTIMAL_BUFFER, [369](#)

- FE2_VP6A_STREAM_PEAK_BITRATE, [370](#)
- FE2_VP6A_STREAM_PREBUFFER, [370](#)
- FE2_VP6A_TEMPORAL_DOWN_-
WATERMARK, [370](#)
- FE2_VP6A_TEMPORAL_RESAMPLING, [371](#)
- FE2_VP6A_UNDERSHOOT_PCT, [371](#)
- codec_vp8
 - FE2_CODEC_VP8, [374](#)
 - FE2_VP8_2PASS_MAX_SECTION, [374](#)
 - FE2_VP8_2PASS_MIN_SECTION, [374](#)
 - FE2_VP8_ALTREF, [374](#)
 - FE2_VP8_AR_MAX_FRAMES, [375](#)
 - FE2_VP8_AR_STRENGTH, [375](#)
 - FE2_VP8_AR_TYPE, [375](#)
 - FE2_VP8_BITRATE, [375](#)
 - FE2_VP8_CXMODE, [375](#)
 - FE2_VP8_DROP_THRESH, [376](#)
 - FE2_VP8_KFFREQ, [376](#)
 - FE2_VP8_KFINTTYPE, [376](#)
 - FE2_VP8_LAG, [376](#)
 - FE2_VP8_MAX_Q, [377](#)
 - FE2_VP8_MB_STATIC_THRESHOLD, [377](#)
 - FE2_VP8_MIN_Q, [377](#)
 - FE2_VP8_NOISE_REDUCTION, [377](#)
 - FE2_VP8_OVERSHOOT_PCT, [378](#)
 - FE2_VP8_PROFILE, [378](#)
 - FE2_VP8_RC_MODE, [378](#)
 - FE2_VP8_SHARPNESS, [378](#)
 - FE2_VP8_STREAM_INITIAL_BUFFER, [378](#)
 - FE2_VP8_STREAM_MAX_BUFFER, [379](#)
 - FE2_VP8_STREAM_OPTIMAL_BUFFER, [379](#)
 - FE2_VP8_THREADS, [379](#)
 - FE2_VP8_TOKEN_PARTITIONS, [379](#)
 - FE2_VP8_UNDERSHOOT_PCT, [380](#)
- codecfunc
 - Flix2_AddCodec, [495](#)
 - Flix2_CodecGetParam, [496](#)
 - Flix2_CodecGetParamAsStr, [496](#)
 - Flix2_CodecSetParam, [497](#)
 - Flix2_CodecSetParamAsStr, [497](#)
 - Flix2_RemoveCodec, [497](#)
- codecparm_common
 - CBR_1PASSControl, [334](#)
 - CBR_2PASSControl, [334](#)
 - FIXED_KEYFRAMES, [334](#)
 - MAX_KEYFRAMES, [334](#)
 - VBR_1PASSControl, [334](#)
 - VBR_2PASSControl, [334](#)
- codecparm_common
 - FE2_CODECPARAM_BITRATE, [333](#)
 - FE2_VCODECPARAM_KFFREQ, [333](#)
 - FE2_VCODECPARAM_KFINTTYPE, [333](#)
 - FE2_VCODECPARAM_RC_MODE, [334](#)
 - FE2_VideoBitrateControls, [334](#)
 - FE2_VideoKeyframeTypes, [334](#)
- Codecs, [332](#)
- com_version
 - flxengine_com::IFlix, [600](#)
- Common Codec Parameters, [333](#)
- COMPRESSMODE_BEST
 - codec_vp6, [363](#)
- COMPRESSMODE_GOOD
 - codec_vp6, [363](#)
- copyright
 - flxengine_com::IFlix, [600](#)
- Crop, [405](#)
- CUE_EVENT
 - vidopts_deprecated, [566](#)
- CUE_NAVIGATION
 - vidopts_deprecated, [566](#)
- Cut, [421](#)
- Deinterlace, [389](#)
- DEINTERLACE_1_2_1_BLUR
 - filter_adaptive_deinterlace, [390](#)
- DEINTERLACE_ADAPTIVE
 - filter_adaptive_deinterlace, [390](#)
- DEINTERLACE_DROP_FIELD
 - filter_adaptive_deinterlace, [390](#)
- DEINTERLACE_NONE
 - filter_adaptive_deinterlace, [390](#)
- deinterlacemode
 - filter_adaptive_deinterlace, [390](#)
- deintmode_t
 - filter_adaptive_deinterlace, [390](#)
- deleteVariable
 - flxengine_com::ISwfOptions, [628](#)
- Denoise, [427](#)
- Deprecated, [325](#), [503](#), [564](#)
- DLLEXPORT
 - on2types.h, [699](#)
- DLLIMPORT
 - on2types.h, [699](#)
- DLLLOCAL
 - on2types.h, [699](#)
- editor_options_GetBrightness
 - filter_bchs, [395](#)
 - filter_crop, [408](#)
- editor_options_GetContrast
 - filter_bchs, [395](#)
 - filter_crop, [408](#)
- editor_options_GetCrop
 - filter_crop, [409](#)

- editor_options_GetCropBounds
 - filter_crop, [409](#)
- editor_options_GetCutStartTime
 - filter_crop, [410](#)
 - filter_cut, [423](#)
- editor_options_GetCutStopTime
 - filter_crop, [410](#)
 - filter_cut, [423](#)
- editor_options_GetHue
 - filter_bchs, [395](#)
 - filter_crop, [411](#)
- editor_options_GetSaturation
 - filter_bchs, [396](#)
 - filter_crop, [411](#)
- editor_options_GetUseBrightness
 - filter_bchs, [396](#)
 - filter_crop, [412](#)
- editor_options_GetUseContrast
 - filter_bchs, [396](#)
 - filter_crop, [412](#)
- editor_options_GetUseCut
 - filter_crop, [413](#)
 - filter_cut, [424](#)
- editor_options_GetUseHue
 - filter_bchs, [397](#)
 - filter_crop, [413](#)
- editor_options_GetUseSaturation
 - filter_bchs, [397](#)
 - filter_crop, [413](#)
- editor_options_Reset
 - filterfunc_deprecated, [503](#)
- editor_options_SetBrightness
 - filter_bchs, [398](#)
 - filter_crop, [414](#)
- editor_options_SetContrast
 - filter_bchs, [398](#)
 - filter_crop, [414](#)
- editor_options_SetCrop
 - filter_crop, [415](#)
- editor_options_SetCropBounds
 - filter_crop, [415](#)
- editor_options_SetCutStartTime
 - filter_crop, [416](#)
 - filter_cut, [424](#)
- editor_options_SetCutStopTime
 - filter_crop, [416](#)
 - filter_cut, [425](#)
- editor_options_SetHue
 - filter_bchs, [399](#)
 - filter_crop, [417](#)
- editor_options_SetSaturation
 - filter_bchs, [399](#)
 - filter_crop, [418](#)
- editor_options_SetUseBrightness
 - filter_bchs, [400](#)
 - filter_crop, [418](#)
- editor_options_SetUseContrast
 - filter_bchs, [400](#)
 - filter_crop, [418](#)
- editor_options_SetUseCut
 - filter_crop, [419](#)
 - filter_cut, [425](#)
- editor_options_SetUseHue
 - filter_bchs, [400](#)
 - filter_crop, [419](#)
- editor_options_SetUseSaturation
 - filter_bchs, [401](#)
 - filter_crop, [420](#)
- editor_options_Validate
 - filterfunc_deprecated, [503](#)
- elapsedTime
 - flixengine_com::IEncodingStatus, [584](#)
- EmbeddedUrlIsGetUrl
 - muxer_swf, [530](#)
- EmbeddedUrlIsLoadMovie
 - muxer_swf, [530](#)
- encode
 - flixengine_com::IFlix, [600](#)
- Encoding Statistics, [381](#)
- encoding_status_GetAverageBitrate
 - encstatus, [381](#)
- encoding_status_GetAverageFramesize
 - encstatus, [382](#)
- encoding_status_GetElapsedTime
 - encstatus, [382](#)
- encoding_status_GetEndTime
 - encstatus, [382](#)
- encoding_status_GetMaximumFramesize
 - encstatus, [383](#)
- encoding_status_GetMinimumFramesize
 - encstatus, [383](#)
- encoding_status_GetStartTime
 - encstatus, [383](#)
- encoding_status_GetTotalFrames
 - encstatus, [384](#)
- encoding_status_PercentComplete
 - encstatus, [384](#)
- encodingStatus
 - flixengine_com::IFlix, [600](#)
- EncStateIdle
 - mainengine, [479](#)
- EncStateQueued
 - mainengine, [479](#)
- EncStateRunning
 - mainengine, [479](#)
- encstatus
 - encoding_status_GetAverageBitrate, [381](#)
 - encoding_status_GetAverageFramesize, [382](#)

- encoding_status_GetElapsedTime, 382
- encoding_status_GetEndTime, 382
- encoding_status_GetMaximumFramesize, 383
- encoding_status_GetMinimumFramesize, 383
- encoding_status_GetStartTime, 383
- encoding_status_GetTotalFrames, 384
- encoding_status_PercentComplete, 384
- endTime
 - flxengine_com::IEncodingStatus, 584
- ErrEncodeA
 - mainengine, 479
- ErrEncodeV
 - mainengine, 479
- ErrFileDecode
 - mainengine, 479
- ErrFileDecodeA
 - mainengine, 479
- ErrFileDecodeV
 - mainengine, 479
- ErrFileIO
 - mainengine, 479
- ErrFileOpen
 - mainengine, 479
- errno_
 - flxengine_com::IFlix, 601
- ErrNone
 - mainengine, 479
- ErrSys
 - mainengine, 479
- ExportFLVVideo
 - mainengine, 480
- ExportSWF3To6Video
 - mainengine, 480
- ExportSWF7PlusVideo
 - mainengine, 480
- ExportSWFVectorVideo
 - mainengine, 480
- FE2_OVERLAY_POS_MODE_BOTLEFT
 - filter_overlay, 445
- FE2_OVERLAY_POS_MODE_BOTRIGHT
 - filter_overlay, 445
- FE2_OVERLAY_POS_MODE_CENTER
 - filter_overlay, 445
- FE2_OVERLAY_POS_MODE_TOPLEFT
 - filter_overlay, 445
- FE2_OVERLAY_POS_MODE_TOPRIGHT
 - filter_overlay, 445
- FE2_OVERLAY_POS_MODE_XY
 - filter_overlay, 445
- FE2_PNGEX_CP_ALL
 - filter_pngexporter, 460
- FE2_PNGEX_CP_EVENT
 - filter_pngexporter, 460
- FE2_PNGEX_CP_NAV
 - filter_pngexporter, 460
- FE2_3G2_FASTSTART
 - flxengine_com::IFlix, 603
 - muxer_3g2lavf, 515
- FE2_3GP_FASTSTART
 - flxengine_com::IFlix, 603
 - muxer_3gpLavf, 513
- FE2_AAC_BITRATE
 - codec_aac, 338
 - flxengine_com::IFlix, 603
- FE2_AACPLUS_BITRATE
 - codec_aacplus, 340
 - flxengine_com::IFlix, 603
- FE2_AACPLUS_PARAMETRIC_STEREO
 - codec_aacplus, 340
 - flxengine_com::IFlix, 603
- FE2_ADAPTIVE_DEINTERLACE_MODE
 - filter_adaptive_deinterlace, 390
 - flxengine_com::IFlix, 603
- FE2_AMR_BITRATE
 - codec_amrlavc, 342
 - flxengine_com::IFlix, 603
- FE2_AudioBitrates
 - audopts, 324
- FE2_AudioSamplingrates
 - audopts_deprecated, 326
- FE2_BCHS_BRIGHTNESS
 - filter_bchs, 393
 - flxengine_com::IFlix, 604
- FE2_BCHS_CONTRAST
 - filter_bchs, 394
 - flxengine_com::IFlix, 604
- FE2_BCHS_HUE
 - filter_bchs, 394
 - flxengine_com::IFlix, 604
- FE2_BCHS_SATURATION
 - filter_bchs, 394
 - flxengine_com::IFlix, 604
- FE2_BLUR_FILTER
 - filter_blur, 403
 - flxengine_com::IFlix, 604
- FE2_BLUR_MASKSIZE
 - filter_blur, 403
 - flxengine_com::IFlix, 604
- FE2_CODEC_AAC
 - codec_aac, 338
 - flxengine_com::IFlix, 604
- FE2_CODEC_AACPLUS
 - codec_aacplus, 340
 - flxengine_com::IFlix, 604
- FE2_CODEC_AMR_NB
 - codec_amrlavc, 342
 - flxengine_com::IFlix, 604

- FE2_CODEC_H263
 - codec_h263, 343
 - flxengine_com::IFlix, 604
- FE2_CODEC_H263_BASELINE
 - codec_h263, 343
 - flxengine_com::IFlix, 604
- FE2_CODEC_H264
 - codec_h264, 347
 - flxengine_com::IFlix, 605
- FE2_CODEC_LAME
 - codec_lame, 351
 - flxengine_com::IFlix, 605
- FE2_CODEC_PCM
 - audio_codecs, 336
 - flxengine_com::IFlix, 605
- FE2_CODEC_VORBIS
 - codec_vorbis, 353
 - flxengine_com::IFlix, 605
- FE2_CODEC_VP6
 - codec_vp6, 357
 - flxengine_com::IFlix, 605
- FE2_CODEC_VP6ALPHA
 - codec_vp6a, 366
 - flxengine_com::IFlix, 605
- FE2_CODEC_VP8
 - codec_vp8, 374
 - flxengine_com::IFlix, 605
- FE2_CODECPARAM_BITRATE
 - codecparam_common, 333
 - flxengine_com::IFlix, 605
- FE2_CompressMode
 - codec_vp6, 363
- FE2_CROP_BOTTOM
 - filter_crop, 407
 - flxengine_com::IFlix, 605
- FE2_CROP_LEFT
 - filter_crop, 407
 - flxengine_com::IFlix, 605
- FE2_CROP_RIGHT
 - filter_crop, 407
 - flxengine_com::IFlix, 605
- FE2_CROP_TOP
 - filter_crop, 408
 - flxengine_com::IFlix, 606
- FE2_CuePointType
 - vidopts_deprecated, 566
- FE2_CUT_START_SEC
 - filter_cut, 422
 - flxengine_com::IFlix, 606
- FE2_CUT_STOP_SEC
 - filter_cut, 422
 - flxengine_com::IFlix, 606
- FE2_CUT_USE SEEK
 - filter_cut, 422
- flxengine_com::IFlix, 606
- FE2_DENOISE_NOISE_LEVEL
 - filter_denoise, 427
 - flxengine_com::IFlix, 606
- FE2_EmbeddedUrlType
 - muxer_swf, 530
- FE2_EncState
 - mainengine, 479
- FE2_errno
 - mainengine, 479
- FE2_ExportedVideoType
 - mainengine, 479
- FE2_FILTER_ADAPTIVE_DEINTERLACE
 - filter_adaptive_deinterlace, 390
 - flxengine_com::IFlix, 606
- FE2_FILTER_BCHS
 - filter_bchs, 394
 - flxengine_com::IFlix, 606
- FE2_FILTER_BLUR
 - filter_blur, 403
 - flxengine_com::IFlix, 606
- FE2_FILTER_CROP
 - filter_crop, 408
 - flxengine_com::IFlix, 606
- FE2_FILTER_CUT
 - filter_cut, 422
 - flxengine_com::IFlix, 606
- FE2_FILTER_DENOISE
 - filter_denoise, 427
 - flxengine_com::IFlix, 606
- FE2_FILTER_FRAMERATE
 - filter_framerate, 429
 - flxengine_com::IFlix, 607
- FE2_FILTER_HIGHPASS
 - filter_highpass, 434
 - flxengine_com::IFlix, 607
- FE2_FILTER_LOWPASS
 - filter_lowpass, 436
 - flxengine_com::IFlix, 607
- FE2_FILTER_MIRROR
 - filter_mirror, 438
 - flxengine_com::IFlix, 607
- FE2_FILTER_OVERLAY
 - filter_overlay, 442
 - flxengine_com::IFlix, 607
- FE2_FILTER_PNGEX
 - filter_pngexporter, 455
 - flxengine_com::IFlix, 607
- FE2_FILTER_RESAMPLE
 - filter_resample, 461
 - flxengine_com::IFlix, 607
- FE2_FILTER_ROTATE
 - filter_rotate, 462
 - flxengine_com::IFlix, 607

- FE2_FILTER_SCALE
 - filter_scale, [465](#)
 - flxengine_com::IFlix, [607](#)
- FE2_FILTER_SHARPEN
 - filter_sharpen, [475](#)
 - flxengine_com::IFlix, [607](#)
- FE2_FLV_CUEPT_EVENT
 - flxengine_com::IFlix, [607](#)
 - muxer_flv, [507](#)
- FE2_FLV_CUEPT_NAV
 - flxengine_com::IFlix, [608](#)
 - muxer_flv, [507](#)
- FE2_FLV_CUEPT_PARAM
 - flxengine_com::IFlix, [608](#)
 - muxer_flv, [507](#)
- FE2_FLV_METADATA_DISABLE
 - flxengine_com::IFlix, [608](#)
 - muxer_flv, [508](#)
- FE2_FLV_METADATA_ENABLE
 - flxengine_com::IFlix, [608](#)
 - muxer_flv, [508](#)
- FE2_FlvAudioFormat
 - audopts_deprecated, [326](#)
- FE2_FRAMERATE_DECIMATE
 - filter_framerate, [429](#)
 - flxengine_com::IFlix, [608](#)
- FE2_FRAMERATE_FPS
 - filter_framerate, [429](#)
 - flxengine_com::IFlix, [608](#)
- FE2_FXM_CUEPT_EVENT
 - flxengine_com::IFlix, [608](#)
 - muxer_fxm, [511](#)
- FE2_FXM_CUEPT_NAV
 - flxengine_com::IFlix, [608](#)
 - muxer_fxm, [511](#)
- FE2_FXM_CUEPT_PARAM
 - flxengine_com::IFlix, [608](#)
 - muxer_fxm, [511](#)
- FE2_FXM_METADATA_DISABLE
 - flxengine_com::IFlix, [608](#)
 - muxer_fxm, [512](#)
- FE2_FXM_METADATA_ENABLE
 - flxengine_com::IFlix, [608](#)
 - muxer_fxm, [512](#)
- FE2_H263_BITRATE
 - codec_h263, [344](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H263_KFFREQ
 - codec_h263, [344](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H263_KFINTTYPE
 - codec_h263, [344](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H263_MAX_Q
 - codec_h263, [345](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H263_MIN_Q
 - codec_h263, [345](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H263_RC_MODE
 - codec_h263, [345](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H264_B_FRAME_RATE
 - codec_h264, [347](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H264_BITRATE
 - codec_h264, [347](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H264_KFFREQ
 - codec_h264, [348](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H264_KFINTTYPE
 - codec_h264, [348](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H264_PROFILE
 - codec_h264, [348](#)
 - flxengine_com::IFlix, [609](#)
- FE2_H264_RC_MODE
 - codec_h264, [348](#)
 - flxengine_com::IFlix, [610](#)
- FE2_H264_SPEED
 - codec_h264, [349](#)
 - flxengine_com::IFlix, [610](#)
- FE2_HIGHPASS_CUTOFF
 - filter_highpass, [434](#)
 - flxengine_com::IFlix, [610](#)
- FE2_HIGHPASS_Q
 - filter_highpass, [434](#)
 - flxengine_com::IFlix, [610](#)
- FE2_ISOMEDIA_FASTSTART
 - flxengine_com::IFlix, [610](#)
 - isomedia.h, [693](#)
- FE2_LAME_BITRATE
 - codec_lame, [351](#)
 - flxengine_com::IFlix, [610](#)
- FE2_LAME_CHANNELS
 - codec_lame, [351](#)
 - flxengine_com::IFlix, [610](#)
- FE2_LAME_QUALITY
 - codec_lame, [351](#)
 - flxengine_com::IFlix, [610](#)
- FE2_LAME_RC_MODE
 - codec_lame, [352](#)
 - flxengine_com::IFlix, [610](#)
- FE2_LOWPASS_CUTOFF
 - filter_lowpass, [436](#)
 - flxengine_com::IFlix, [610](#)
- FE2_LOWPASS_Q

- filter_lowpass, [436](#)
- flxengine_com::IFlix, [610](#)
- FE2_MIRROR_HORIZONTAL
 - filter_mirror, [438](#)
 - flxengine_com::IFlix, [611](#)
- FE2_MIRROR_VERTICAL
 - filter_mirror, [439](#)
 - flxengine_com::IFlix, [611](#)
- FE2_MOV_FASTSTART
 - flxengine_com::IFlix, [611](#)
 - muxer_movlavf, [517](#)
- FE2_MP4_FASTSTART
 - flxengine_com::IFlix, [611](#)
 - muxer_mp4lavf, [519](#)
- FE2_MUXER_3G2
 - flxengine_com::IFlix, [611](#)
 - muxer_3g2lavf, [515](#)
- FE2_MUXER_3GP
 - flxengine_com::IFlix, [611](#)
 - muxer_3gp1avf, [513](#)
- FE2_MUXER_FLV
 - flxengine_com::IFlix, [611](#)
 - muxer_flv, [508](#)
- FE2_MUXER_FXM
 - flxengine_com::IFlix, [611](#)
 - muxer_fxm, [512](#)
- FE2_MUXER_MOV
 - flxengine_com::IFlix, [611](#)
 - muxer_movlavf, [518](#)
- FE2_MUXER_MP4
 - flxengine_com::IFlix, [611](#)
 - muxer_mp4lavf, [520](#)
- FE2_MUXER_SWF
 - flxengine_com::IFlix, [611](#)
 - muxer_swf, [526](#)
- FE2_MUXER_WEBM
 - flxengine_com::IFlix, [612](#)
 - muxer_webmlavf, [553](#)
- FE2_OVERLAY_FILE
 - filter_overlay, [442](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_MASK_B
 - filter_overlay, [442](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_MASK_G
 - filter_overlay, [443](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_MASK_R
 - filter_overlay, [443](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_MASK_RGB
 - filter_overlay, [443](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_MASK_X
 - filter_overlay, [443](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_MASK_XY
 - filter_overlay, [444](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_MASK_Y
 - filter_overlay, [444](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_POS
 - filter_overlay, [444](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_POS_X
 - filter_overlay, [444](#)
 - flxengine_com::IFlix, [612](#)
- FE2_OVERLAY_POS_Y
 - filter_overlay, [445](#)
 - flxengine_com::IFlix, [613](#)
- FE2_OverlayPositionMode
 - filter_overlay, [445](#)
- FE2_PNGEX_AUTO_EXPORT_COUNT
 - filter_pngexporter, [455](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_AUTO_EXPORT_END_TIME
 - filter_pngexporter, [456](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_AUTO_EXPORT_RANDOM_-
PERIOD
 - filter_pngexporter, [456](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_AUTO_EXPORT_START_TIME
 - filter_pngexporter, [456](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_COMPRESSION_LEVEL
 - filter_pngexporter, [457](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_ENABLE_ALPHA
 - filter_pngexporter, [457](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_EXPORT_CUE_POINTS
 - filter_pngexporter, [457](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_EXPORT_FIRST_FRAME_PNG
 - filter_pngexporter, [458](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_EXPORT_INTERVAL
 - filter_pngexporter, [458](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_EXPORT_TIME_STRING
 - filter_pngexporter, [458](#)
 - flxengine_com::IFlix, [613](#)
- FE2_PNGEX_FILENAME_PREFIX
 - filter_pngexporter, [458](#)
 - flxengine_com::IFlix, [614](#)
- FE2_PNGEX_FILENAME_SUFFIX

- filter_pngexporter, [459](#)
- flxengine_com::IFlix, [614](#)
- FE2_PNGEX_HEIGHT
 - filter_pngexporter, [459](#)
 - flxengine_com::IFlix, [614](#)
- FE2_PNGEX_OUTPUT_DIRECTORY
 - filter_pngexporter, [459](#)
 - flxengine_com::IFlix, [614](#)
- FE2_PNGEX_WIDTH
 - filter_pngexporter, [459](#)
 - flxengine_com::IFlix, [614](#)
- FE2_PNGEX_CuePtMode
 - filter_pngexporter, [460](#)
- FE2_RESAMPLE_CHANNELS
 - filter_resample, [461](#)
 - flxengine_com::IFlix, [614](#)
- FE2_RESAMPLE_RATE
 - filter_resample, [461](#)
 - flxengine_com::IFlix, [614](#)
- FE2_ROTATE_ANGLE
 - filter_rotate, [462](#)
 - flxengine_com::IFlix, [614](#)
- FE2_SCALE_HEIGHT
 - filter_scale, [465](#)
 - flxengine_com::IFlix, [614](#)
- FE2_SCALE_WIDTH
 - filter_scale, [465](#)
 - flxengine_com::IFlix, [614](#)
- FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR
 - flxengine_com::IFlix, [614](#)
 - muxer_swf, [526](#)
- FE2_SWF_ADD_VARIABLE
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [526](#)
- FE2_SWF_DELETE_VARIABLE
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [527](#)
- FE2_SWF_EMBEDDED_URL
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [527](#)
- FE2_SWF_EMBEDDED_URL_TARGET
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [527](#)
- FE2_SWF_EMBEDDED_URL_TYPE
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [527](#)
- FE2_SWF_FIXED_PRELOAD_PCT
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [528](#)
- FE2_SWF_FRAMERATE
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [528](#)
- FE2_SWF_HEIGHT
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [528](#)
- FE2_SWF_LOOP_COUNT
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [528](#)
- FE2_SWF_ON_END_OPTION
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [529](#)
- FE2_SWF_ON_END_URL
 - flxengine_com::IFlix, [615](#)
 - muxer_swf, [529](#)
- FE2_SWF_ON_START_OPTION
 - flxengine_com::IFlix, [616](#)
 - muxer_swf, [529](#)
- FE2_SWF_PRELOAD_TYPE
 - flxengine_com::IFlix, [616](#)
 - muxer_swf, [529](#)
- FE2_SWF_START_BLANK_FRAME
 - flxengine_com::IFlix, [616](#)
 - muxer_swf, [529](#)
- FE2_SWF_START_WAIT_SEC
 - flxengine_com::IFlix, [616](#)
 - muxer_swf, [530](#)
- FE2_SWF_WIDTH
 - flxengine_com::IFlix, [616](#)
 - muxer_swf, [530](#)
- FE2_SwfOnEndOptions
 - muxer_swf, [530](#)
- FE2_SwfOnStartOptions
 - muxer_swf, [531](#)
- FE2_SwfPreloaderOptions
 - muxer_swf, [531](#)
- FE2_VCODECPARAM_KFFREQ
 - codecpam_common, [333](#)
 - flxengine_com::IFlix, [616](#)
- FE2_VCODECPARAM_KFINTTYPE
 - codecpam_common, [333](#)
 - flxengine_com::IFlix, [616](#)
- FE2_VCODECPARAM_RC_MODE
 - codecpam_common, [334](#)
 - flxengine_com::IFlix, [616](#)
- FE2_VideoBitrateControls
 - codecpam_common, [334](#)
- FE2_VideoCodec
 - vidopts_deprecated, [566](#)
- FE2_VideoKeyframeTypes
 - codecpam_common, [334](#)
- FE2_VORBIS_BITRATE
 - codec_vorbis, [353](#)
 - flxengine_com::IFlix, [616](#)
- FE2_VP6_2PASS_MAX_SECTION
 - codec_vp6, [357](#)
 - flxengine_com::IFlix, [616](#)
- FE2_VP6_2PASS_MIN_SECTION

- codec_vp6, 357
- flxengine_com::IFlix, 616
- FE2_VP6_BITRATE
 - codec_vp6, 357
 - flxengine_com::IFlix, 617
- FE2_VP6_CONCURRENCY
 - codec_vp6, 357
 - flxengine_com::IFlix, 617
- FE2_VP6_CXMODE
 - codec_vp6, 358
 - flxengine_com::IFlix, 617
- FE2_VP6_KFFREQ
 - codec_vp6, 359
 - flxengine_com::IFlix, 617
- FE2_VP6_KFINTTYPE
 - codec_vp6, 359
 - flxengine_com::IFlix, 617
- FE2_VP6_MAX_Q
 - codec_vp6, 359
 - flxengine_com::IFlix, 617
- FE2_VP6_MIN_Q
 - codec_vp6, 359
 - flxengine_com::IFlix, 617
- FE2_VP6_NOISE_REDUCTION
 - codec_vp6, 360
 - flxengine_com::IFlix, 617
- FE2_VP6_PROFILE
 - codec_vp6, 360
 - flxengine_com::IFlix, 617
- FE2_VP6_RC_MODE
 - codec_vp6, 360
 - flxengine_com::IFlix, 617
- FE2_VP6_SHARPNESS
 - codec_vp6, 360
 - flxengine_com::IFlix, 617
- FE2_VP6_STREAM_MAX_BUFFER
 - codec_vp6, 361
 - flxengine_com::IFlix, 618
- FE2_VP6_STREAM_OPTIMAL_BUFFER
 - codec_vp6, 361
 - flxengine_com::IFlix, 618
- FE2_VP6_STREAM_PEAK_BITRATE
 - codec_vp6, 361
 - flxengine_com::IFlix, 618
- FE2_VP6_STREAM_PREBUFFER
 - codec_vp6, 361
 - flxengine_com::IFlix, 618
- FE2_VP6_TEMPORAL_DOWN_WATERMARK
 - codec_vp6, 362
 - flxengine_com::IFlix, 618
- FE2_VP6_TEMPORAL_RESAMPLING
 - codec_vp6, 362
 - flxengine_com::IFlix, 618
- FE2_VP6_UNDERSHOOT_PCT
 - codec_vp6, 362
 - flxengine_com::IFlix, 618
- FE2_VP6A_2PASS_MAX_SECTION
 - codec_vp6a, 366
 - flxengine_com::IFlix, 618
- FE2_VP6A_2PASS_MIN_SECTION
 - codec_vp6a, 366
 - flxengine_com::IFlix, 618
- FE2_VP6A_ALPHA_BITRATE
 - codec_vp6a, 366
 - flxengine_com::IFlix, 618
- FE2_VP6A_ALPHA_MAX_Q
 - codec_vp6a, 366
 - flxengine_com::IFlix, 618
- FE2_VP6A_ALPHA_MIN_Q
 - codec_vp6a, 366
 - flxengine_com::IFlix, 619
- FE2_VP6A_ALPHA_NOISE_REDUCTION
 - codec_vp6a, 367
 - flxengine_com::IFlix, 619
- FE2_VP6A_ALPHA_SHARPNESS
 - codec_vp6a, 367
 - flxengine_com::IFlix, 619
- FE2_VP6A_BITRATE
 - codec_vp6a, 367
 - flxengine_com::IFlix, 619
- FE2_VP6A_CXMODE
 - codec_vp6a, 367
 - flxengine_com::IFlix, 619
- FE2_VP6A_KFFREQ
 - codec_vp6a, 367
 - flxengine_com::IFlix, 619
- FE2_VP6A_KFINTTYPE
 - codec_vp6a, 368
 - flxengine_com::IFlix, 619
- FE2_VP6A_MAX_Q
 - codec_vp6a, 368
 - flxengine_com::IFlix, 619
- FE2_VP6A_MIN_Q
 - codec_vp6a, 368
 - flxengine_com::IFlix, 619
- FE2_VP6A_NOISE_REDUCTION
 - codec_vp6a, 369
 - flxengine_com::IFlix, 619
- FE2_VP6A_RC_MODE
 - codec_vp6a, 369
 - flxengine_com::IFlix, 619
- FE2_VP6A_SHARPNESS
 - codec_vp6a, 369
 - flxengine_com::IFlix, 620
- FE2_VP6A_STREAM_MAX_BUFFER
 - codec_vp6a, 369
 - flxengine_com::IFlix, 620
- FE2_VP6A_STREAM_OPTIMAL_BUFFER

- codec_vp6a, [369](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP6A_STREAM_PEAK_BITRATE
 - codec_vp6a, [370](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP6A_STREAM_PREBUFFER
 - codec_vp6a, [370](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP6A_TEMPORAL_DOWN_-WATERMARK
 - codec_vp6a, [370](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP6A_TEMPORAL_RESAMPLING
 - codec_vp6a, [371](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP6A_UNDERSHOOT_PCT
 - codec_vp6a, [371](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP8_2PASS_MAX_SECTION
 - codec_vp8, [374](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP8_2PASS_MIN_SECTION
 - codec_vp8, [374](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP8_ALTREF
 - codec_vp8, [374](#)
 - flxengine_com::IFlix, [620](#)
- FE2_VP8_AR_MAX_FRAMES
 - codec_vp8, [375](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_AR_STRENGTH
 - codec_vp8, [375](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_AR_TYPE
 - codec_vp8, [375](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_BITRATE
 - codec_vp8, [375](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_CXMODE
 - codec_vp8, [375](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_DROP_THRESH
 - codec_vp8, [376](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_KFFREQ
 - codec_vp8, [376](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_KFINTTYPE
 - codec_vp8, [376](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_LAG
 - codec_vp8, [376](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_MAX_Q
 - codec_vp8, [377](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_MB_STATIC_THRESHOLD
 - codec_vp8, [377](#)
 - flxengine_com::IFlix, [621](#)
- FE2_VP8_MIN_Q
 - codec_vp8, [377](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_NOISE_REDUCTION
 - codec_vp8, [377](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_OVERSHOOT_PCT
 - codec_vp8, [378](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_PROFILE
 - codec_vp8, [378](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_RC_MODE
 - codec_vp8, [378](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_SHARPNESS
 - codec_vp8, [378](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_STREAM_INITIAL_BUFFER
 - codec_vp8, [378](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_STREAM_MAX_BUFFER
 - codec_vp8, [379](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_STREAM_OPTIMAL_BUFFER
 - codec_vp8, [379](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_THREADS
 - codec_vp8, [379](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_TOKEN_PARTITIONS
 - codec_vp8, [379](#)
 - flxengine_com::IFlix, [622](#)
- FE2_VP8_UNDERSHOOT_PCT
 - codec_vp8, [380](#)
 - flxengine_com::IFlix, [623](#)
- Filter Manipulation, [491](#)
- filter_adaptive_deinterlace
 - DEINTERLACE_1_2_1_BLUR, [390](#)
 - DEINTERLACE_ADAPTIVE, [390](#)
 - DEINTERLACE_DROP_FIELD, [390](#)
 - DEINTERLACE_NONE, [390](#)
- filter_blur
 - BLUR_GAUSS, [404](#)
 - BLUR_LOWPASS, [403](#)
 - MASK_3x3, [404](#)
 - MASK_5x5, [404](#)
- filter_overlay

- FE2_OVERLAY_POS_MODE_BOTLEFT, 445
- FE2_OVERLAY_POS_MODE_BOTRIGHT, 445
- FE2_OVERLAY_POS_MODE_CENTER, 445
- FE2_OVERLAY_POS_MODE_TOPLEFT, 445
- FE2_OVERLAY_POS_MODE_TOPRIGHT, 445
- FE2_OVERLAY_POS_MODE_XY, 445
- filter_pngexporter
 - FE2_PNGEX_CP_ALL, 460
 - FE2_PNGEX_CP_EVENT, 460
 - FE2_PNGEX_CP_NAV, 460
- filter_adaptive_deinterlace
 - deinterlacemode, 390
 - deintmode_t, 390
 - FE2_ADAPTIVE_DEINTERLACE_MODE, 390
 - FE2_FILTER_ADAPTIVE_DEINTERLACE, 390
 - video_options_GetDeinterlace, 390
 - video_options_SetDeinterlace, 391
- filter_bchs
 - editor_options_GetBrightness, 395
 - editor_options_GetContrast, 395
 - editor_options_GetHue, 395
 - editor_options_GetSaturation, 396
 - editor_options_GetUseBrightness, 396
 - editor_options_GetUseContrast, 396
 - editor_options_GetUseHue, 397
 - editor_options_GetUseSaturation, 397
 - editor_options_SetBrightness, 398
 - editor_options_SetContrast, 398
 - editor_options_SetHue, 399
 - editor_options_SetSaturation, 399
 - editor_options_SetUseBrightness, 400
 - editor_options_SetUseContrast, 400
 - editor_options_SetUseHue, 400
 - editor_options_SetUseSaturation, 401
 - FE2_BCHS_BRIGHTNESS, 393
 - FE2_BCHS_CONTRAST, 394
 - FE2_BCHS_HUE, 394
 - FE2_BCHS_SATURATION, 394
 - FE2_FILTER_BCHS, 394
- filter_blur
 - blurfilter, 403
 - blurfilter_t, 403
 - FE2_BLUR_FILTER, 403
 - FE2_BLUR_MASKSIZE, 403
 - FE2_FILTER_BLUR, 403
 - masksiz, 404
 - masksiz_t, 403
- filter_crop
 - editor_options_GetBrightness, 408
 - editor_options_GetContrast, 408
 - editor_options_GetCrop, 409
 - editor_options_GetCropBounds, 409
 - editor_options_GetCutStartTime, 410
 - editor_options_GetCutStopTime, 410
 - editor_options_GetHue, 411
 - editor_options_GetSaturation, 411
 - editor_options_GetUseBrightness, 412
 - editor_options_GetUseContrast, 412
 - editor_options_GetUseCut, 413
 - editor_options_GetUseHue, 413
 - editor_options_GetUseSaturation, 413
 - editor_options_SetBrightness, 414
 - editor_options_SetContrast, 414
 - editor_options_SetCrop, 415
 - editor_options_SetCropBounds, 415
 - editor_options_SetCutStartTime, 416
 - editor_options_SetCutStopTime, 416
 - editor_options_SetHue, 417
 - editor_options_SetSaturation, 418
 - editor_options_SetUseBrightness, 418
 - editor_options_SetUseContrast, 418
 - editor_options_SetUseCut, 419
 - editor_options_SetUseHue, 419
 - editor_options_SetUseSaturation, 420
 - FE2_CROP_BOTTOM, 407
 - FE2_CROP_LEFT, 407
 - FE2_CROP_RIGHT, 407
 - FE2_CROP_TOP, 408
 - FE2_FILTER_CROP, 408
- filter_cut
 - editor_options_GetCutStartTime, 423
 - editor_options_GetCutStopTime, 423
 - editor_options_GetUseCut, 424
 - editor_options_SetCutStartTime, 424
 - editor_options_SetCutStopTime, 425
 - editor_options_SetUseCut, 425
 - FE2_CUT_START_SEC, 422
 - FE2_CUT_STOP_SEC, 422
 - FE2_CUT_USE_SEEK, 422
 - FE2_FILTER_CUT, 422
- filter_denoise
 - FE2_DENOISE_NOISE_LEVEL, 427
 - FE2_FILTER_DENOISE, 427
- filter_framerate
 - FE2_FILTER_FRAMERATE, 429
 - FE2_FRAMERATE_DECIMATE, 429
 - FE2_FRAMERATE_FPS, 429
 - video_options_GetDecimateValue, 429
 - video_options_GetUseSourceFramerate, 430
 - video_options_GetVideoFramerate, 430

- video_options_GetVideoFramerateAsDouble, 431
- video_options_SetDecimateValue, 431
- video_options_SetUseSourceFramerate, 432
- video_options_SetVideoFramerate, 432
- video_options_SetVideoFramerateAsDouble, 432
- filter_highpass
 - FE2_FILTER_HIGHPASS, 434
 - FE2_HIGHPASS_CUTOFF, 434
 - FE2_HIGHPASS_Q, 434
- filter_lowpass
 - FE2_FILTER_LOWPASS, 436
 - FE2_LOWPASS_CUTOFF, 436
 - FE2_LOWPASS_Q, 436
- filter_mirror
 - FE2_FILTER_MIRROR, 438
 - FE2_MIRROR_HORIZONTAL, 438
 - FE2_MIRROR_VERTICAL, 439
- filter_overlay
 - FE2_FILTER_OVERLAY, 442
 - FE2_OVERLAY_FILE, 442
 - FE2_OVERLAY_MASK_B, 442
 - FE2_OVERLAY_MASK_G, 443
 - FE2_OVERLAY_MASK_R, 443
 - FE2_OVERLAY_MASK_RGB, 443
 - FE2_OVERLAY_MASK_X, 443
 - FE2_OVERLAY_MASK_XY, 444
 - FE2_OVERLAY_MASK_Y, 444
 - FE2_OVERLAY_POS, 444
 - FE2_OVERLAY_POS_X, 444
 - FE2_OVERLAY_POS_Y, 445
 - FE2_OverlayPositionMode, 445
 - overlay_options_GetMaskPixelRGB, 445
 - overlay_options_GetMaskPixelXY, 446
 - overlay_options_GetOverlayPath, 447
 - overlay_options_GetOverlayPosition, 447
 - overlay_options_GetUseOverlay, 448
 - overlay_options_Reset, 448
 - overlay_options_SetMaskPixelRGB, 449
 - overlay_options_SetMaskPixelXY, 450
 - overlay_options_SetOverlayPath, 450
 - overlay_options_SetOverlayPosition, 451
 - overlay_options_SetUseOverlay, 452
- filter_pngexporter
 - FE2_FILTER_PNGEX, 455
 - FE2_PNGEX_AUTO_EXPORT_COUNT, 455
 - FE2_PNGEX_AUTO_EXPORT_END_TIME, 456
 - FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD, 456
 - FE2_PNGEX_AUTO_EXPORT_START_TIME, 456
 - FE2_PNGEX_COMPRESSION_LEVEL, 457
 - FE2_PNGEX_ENABLE_ALPHA, 457
 - FE2_PNGEX_EXPORT_CUE_POINTS, 457
 - FE2_PNGEX_EXPORT_FIRST_FRAME_PNG, 458
 - FE2_PNGEX_EXPORT_INTERVAL, 458
 - FE2_PNGEX_EXPORT_TIME_STRING, 458
 - FE2_PNGEX_FILENAME_PREFIX, 458
 - FE2_PNGEX_FILENAME_SUFFIX, 459
 - FE2_PNGEX_HEIGHT, 459
 - FE2_PNGEX_OUTPUT_DIRECTORY, 459
 - FE2_PNGEX_WIDTH, 459
 - FE2_PNGEXCuePtMode, 460
- filter_resample
 - FE2_FILTER_RESAMPLE, 461
 - FE2_RESAMPLE_CHANNELS, 461
 - FE2_RESAMPLE_RATE, 461
- filter_rotate
 - FE2_FILTER_ROTATE, 462
 - FE2_ROTATE_ANGLE, 462
- filter_scale
 - FE2_FILTER_SCALE, 465
 - FE2_SCALE_HEIGHT, 465
 - FE2_SCALE_WIDTH, 465
 - video_options_GetDecimateValue, 466
 - video_options_GetDeinterlace, 466
 - video_options_GetImageHeight, 467
 - video_options_GetImageWidth, 467
 - video_options_GetUseSourceDimensions, 468
 - video_options_GetUseSourceFramerate, 468
 - video_options_GetVideoFramerate, 469
 - video_options_GetVideoFramerateAsDouble, 469
 - video_options_SetDecimateValue, 470
 - video_options_SetDeinterlace, 470
 - video_options_SetImageHeight, 470
 - video_options_SetImageWidth, 471
 - video_options_SetUseSourceDimensions, 472
 - video_options_SetUseSourceFramerate, 472
 - video_options_SetVideoFramerate, 473
 - video_options_SetVideoFramerateAsDouble, 473
- filter_sharpen
 - FE2_FILTER_SHARPEN, 475
- filterfunc
 - Flix2_AddFilter, 491
 - Flix2_FilterGetParam, 492
 - Flix2_FilterGetParamAsStr, 492
 - Flix2_FilterSetParam, 493
 - Flix2_FilterSetParamAsStr, 493
 - Flix2_RemoveFilter, 494
- filterfunc_deprecated
 - editor_options_Reset, 503

- editor_options_Validate, 503
- Filters, 386
- FIXED_KEYFRAMES
 - codecparm_common, 334
- Flix Engine API, 476
- Flix Engine COM, 578
- Flix2_AddCodec
 - codecfunc, 495
- Flix2_AddFilter
 - filterfunc, 491
- Flix2_AddMuxer
 - muxerfunc, 499
- Flix2_CodecGetParam
 - codecfunc, 496
- Flix2_CodecGetParamAsStr
 - codecfunc, 496
- Flix2_CodecSetParam
 - codecfunc, 497
- Flix2_CodecSetParamAsStr
 - codecfunc, 497
- Flix2_Copyright
 - mainengine, 480
- Flix2_Create
 - mainengine, 480
- Flix2_Destroy
 - mainengine, 480
- Flix2_Encode
 - mainengine, 481
- Flix2_Errno
 - mainengine, 481
- Flix2_FilterGetParam
 - filterfunc, 492
- Flix2_FilterGetParamAsStr
 - filterfunc, 492
- Flix2_FilterSetParam
 - filterfunc, 493
- Flix2_FilterSetParamAsStr
 - filterfunc, 493
- Flix2_GetEncoderState
 - mainengine, 482
- Flix2_GetExportAudio
 - mainengine, 482
- Flix2_GetExportVideo
 - mainengine, 482
- Flix2_GetExportVideoType
 - mainengine, 483
- Flix2_GetInputFile
 - mainengine, 483
- Flix2_GetLogLevel
 - flxengine2.h, 684
- Flix2_GetLogPath
 - flxengine2.h, 684
- Flix2_GetOutputFile
 - mainengine, 484
- Flix2_GetOverwriteExistingFiles
 - mainengine, 484
- Flix2_GetSourceDuration
 - mainengine, 485
- Flix2_IsEncoderRunning
 - mainengine, 485
- Flix2_MuxerGetParam
 - muxerfunc, 500
- Flix2_MuxerGetParamAsStr
 - muxerfunc, 500
- Flix2_MuxerSetParam
 - muxerfunc, 501
- Flix2_MuxerSetParamAsStr
 - muxerfunc, 501
- Flix2_RemoveCodec
 - codecfunc, 497
- Flix2_RemoveFilter
 - filterfunc, 494
- Flix2_RemoveMuxer
 - muxerfunc, 501
- Flix2_Reset
 - mainengine, 486
- Flix2_SetExportAudio
 - mainengine, 486
- Flix2_SetExportVideo
 - mainengine, 486
- Flix2_SetExportVideoType
 - mainengine, 487
- Flix2_SetInputFile
 - mainengine, 487
- Flix2_SetLogLevel
 - flxengine2.h, 685
- Flix2_SetLogPath
 - flxengine2.h, 685
- Flix2_SetOutputFile
 - mainengine, 488
- Flix2_SetOverwriteExistingFiles
 - mainengine, 488
- Flix2_StopEncoding
 - mainengine, 489
- Flix2_Validate
 - mainengine, 489
- Flix2_Version
 - mainengine, 490
- FLIX2HANDLE
 - flxengine2.h, 684
- FLIX2PLGNHANDLE
 - flxengine2.h, 684
- flxengine2.h
 - Flix2_GetLogLevel, 684
 - Flix2_GetLogPath, 684
 - Flix2_SetLogLevel, 685
 - Flix2_SetLogPath, 685
 - FLIX2HANDLE, 684

- FLIX2PLGNHANDLE, 684
- FLIXENGINE_API, 683
- FLIXENGINE_VERSION_CHIEF, 683
- FLIXENGINE_VERSION_EXTRA, 683
- FLIXENGINE_VERSION_MAJOR, 683
- FLIXENGINE_VERSION_MINOR, 683
- FLIXENGINE_VERSION_PATCH, 683
- FLIXENGINE_VERSION_STR, 684
- flixengine2/audio_options.h, 642
- flixengine2/codecs_constants.h, 644
- flixengine2/codecs/aac.h, 645
- flixengine2/codecs/amr.h, 646
- flixengine2/codecs/codecs_common.h, 647
- flixengine2/codecs/h263.h, 648
- flixengine2/codecs/h264.h, 649
- flixengine2/codecs/lame.h, 651
- flixengine2/codecs/vorbis.h, 652
- flixengine2/codecs/vp6.h, 653
- flixengine2/codecs/vp6_alpha.h, 655
- flixengine2/codecs/vp8.h, 657
- flixengine2/encoding_status.h, 659
- flixengine2/filter_constants.h, 660
- flixengine2/filters/adaptive_deinterlace.h, 661
- flixengine2/filters/bchs.h, 662
- flixengine2/filters/blur.h, 663
- flixengine2/filters/crop.h, 664
- flixengine2/filters/cut.h, 665
- flixengine2/filters/denoise.h, 666
- flixengine2/filters/framerate.h, 667
- flixengine2/filters/highpass.h, 668
- flixengine2/filters/lowpass.h, 669
- flixengine2/filters/mirror.h, 670
- flixengine2/filters/overlay.h, 671
- flixengine2/filters/png_export.h, 673
- flixengine2/filters/resample.h, 675
- flixengine2/filters/rotate.h, 676
- flixengine2/filters/scale.h, 677
- flixengine2/filters/sharpen.h, 678
- flixengine2/flixengine2.h, 679
- flixengine2/media_editor_options.h, 687
- flixengine2/muxers_constants.h, 689
- flixengine2/muxers/flv.h, 690
- flixengine2/muxers/fxm.h, 692
- flixengine2/muxers/isomedia.h, 693
- flixengine2/muxers/swf.h, 695
- flixengine2/muxers/webm.h, 697
- flixengine2/on2types.h, 698
- flixengine2/overlay_options.h, 701
- flixengine2/swf_options.h, 702
- flixengine2/video_options.h, 705
- FLIXENGINE_API
 - flixengine2.h, 683
- flixengine_com, 581
- flixengine_com.idl, 709
- warning, 709
- flixengine_com::IEncodingStatus, 583
 - averageBitrate, 584
 - averageFramesize, 584
 - elapsedTime, 584
 - endTime, 584
 - maximumFramesize, 584
 - minimumFramesize, 584
 - percentComplete, 584
 - sc, 585
 - startTime, 584
 - totalFrames, 585
- flixengine_com::IFlix, 586
 - addCodec, 600
 - addFilter, 600
 - addMuxer, 600
 - com_version, 600
 - copyright, 600
 - encode, 600
 - encodingStatus, 600
 - errno_, 601
 - FE2_3G2_FASTSTART, 603
 - FE2_3GP_FASTSTART, 603
 - FE2_AAC_BITRATE, 603
 - FE2_AACPLUS_BITRATE, 603
 - FE2_AACPLUS_PARAMETRIC_STEREO, 603
 - FE2_ADAPTIVE_DEINTERLACE_MODE, 603
 - FE2_AMR_BITRATE, 603
 - FE2_BCHS_BRIGHTNESS, 604
 - FE2_BCHS_CONTRAST, 604
 - FE2_BCHS_HUE, 604
 - FE2_BCHS_SATURATION, 604
 - FE2_BLUR_FILTER, 604
 - FE2_BLUR_MASKSIZE, 604
 - FE2_CODEC_AAC, 604
 - FE2_CODEC_AACPLUS, 604
 - FE2_CODEC_AMR_NB, 604
 - FE2_CODEC_H263, 604
 - FE2_CODEC_H263_BASELINE, 604
 - FE2_CODEC_H264, 605
 - FE2_CODEC_LAME, 605
 - FE2_CODEC_PCM, 605
 - FE2_CODEC_VORBIS, 605
 - FE2_CODEC_VP6, 605
 - FE2_CODEC_VP6ALPHA, 605
 - FE2_CODEC_VP8, 605
 - FE2_CODECPARAM_BITRATE, 605
 - FE2_CROP_BOTTOM, 605
 - FE2_CROP_LEFT, 605
 - FE2_CROP_RIGHT, 605
 - FE2_CROP_TOP, 606
 - FE2_CUT_START_SEC, 606

- FE2_CUT_STOP_SEC, 606
- FE2_CUT_USE_SEEK, 606
- FE2_DENOISE_NOISE_LEVEL, 606
- FE2_FILTER_ADAPTIVE_DEINTERLACE, 606
- FE2_FILTER_BCHS, 606
- FE2_FILTER_BLUR, 606
- FE2_FILTER_CROP, 606
- FE2_FILTER_CUT, 606
- FE2_FILTER_DENOISE, 606
- FE2_FILTER_FRAMERATE, 607
- FE2_FILTER_HIGHPASS, 607
- FE2_FILTER_LOWPASS, 607
- FE2_FILTER_MIRROR, 607
- FE2_FILTER_OVERLAY, 607
- FE2_FILTER_PNGEX, 607
- FE2_FILTER_RESAMPLE, 607
- FE2_FILTER_ROTATE, 607
- FE2_FILTER_SCALE, 607
- FE2_FILTER_SHARPEN, 607
- FE2_FLV_CUEPT_EVENT, 607
- FE2_FLV_CUEPT_NAV, 608
- FE2_FLV_CUEPT_PARAM, 608
- FE2_FLV_METADATA_DISABLE, 608
- FE2_FLV_METADATA_ENABLE, 608
- FE2_FRAMERATE_DECIMATE, 608
- FE2_FRAMERATE_FPS, 608
- FE2_FXM_CUEPT_EVENT, 608
- FE2_FXM_CUEPT_NAV, 608
- FE2_FXM_CUEPT_PARAM, 608
- FE2_FXM_METADATA_DISABLE, 608
- FE2_FXM_METADATA_ENABLE, 608
- FE2_H263_BITRATE, 609
- FE2_H263_KFFREQ, 609
- FE2_H263_KFINTTYPE, 609
- FE2_H263_MAX_Q, 609
- FE2_H263_MIN_Q, 609
- FE2_H263_RC_MODE, 609
- FE2_H264_B_FRAME_RATE, 609
- FE2_H264_BITRATE, 609
- FE2_H264_KFFREQ, 609
- FE2_H264_KFINTTYPE, 609
- FE2_H264_PROFILE, 609
- FE2_H264_RC_MODE, 610
- FE2_H264_SPEED, 610
- FE2_HIGHPASS_CUTOFF, 610
- FE2_HIGHPASS_Q, 610
- FE2_ISOMEDIA_FASTSTART, 610
- FE2_LAME_BITRATE, 610
- FE2_LAME_CHANNELS, 610
- FE2_LAME_QUALITY, 610
- FE2_LAME_RC_MODE, 610
- FE2_LOWPASS_CUTOFF, 610
- FE2_LOWPASS_Q, 610
- FE2_MIRROR_HORIZONTAL, 611
- FE2_MIRROR_VERTICAL, 611
- FE2_MOV_FASTSTART, 611
- FE2_MP4_FASTSTART, 611
- FE2_MUXER_3G2, 611
- FE2_MUXER_3GP, 611
- FE2_MUXER_FLV, 611
- FE2_MUXER_FXM, 611
- FE2_MUXER_MOV, 611
- FE2_MUXER_MP4, 611
- FE2_MUXER_SWF, 611
- FE2_MUXER_WEBM, 612
- FE2_OVERLAY_FILE, 612
- FE2_OVERLAY_MASK_B, 612
- FE2_OVERLAY_MASK_G, 612
- FE2_OVERLAY_MASK_R, 612
- FE2_OVERLAY_MASK_RGB, 612
- FE2_OVERLAY_MASK_X, 612
- FE2_OVERLAY_MASK_XY, 612
- FE2_OVERLAY_MASK_Y, 612
- FE2_OVERLAY_POS, 612
- FE2_OVERLAY_POS_X, 612
- FE2_OVERLAY_POS_Y, 613
- FE2_PNGEX_AUTO_EXPORT_COUNT, 613
- FE2_PNGEX_AUTO_EXPORT_END_-
TIME, 613
- FE2_PNGEX_AUTO_EXPORT_-
RANDOM_PERIOD, 613
- FE2_PNGEX_AUTO_EXPORT_START_-
TIME, 613
- FE2_PNGEX_COMPRESSION_LEVEL, 613
- FE2_PNGEX_ENABLE_ALPHA, 613
- FE2_PNGEX_EXPORT_CUE_POINTS, 613
- FE2_PNGEX_EXPORT_FIRST_FRAME_-
PNG, 613
- FE2_PNGEX_EXPORT_INTERVAL, 613
- FE2_PNGEX_EXPORT_TIME_STRING, 613
- FE2_PNGEX_FILENAME_PREFIX, 614
- FE2_PNGEX_FILENAME_SUFFIX, 614
- FE2_PNGEX_HEIGHT, 614
- FE2_PNGEX_OUTPUT_DIRECTORY, 614
- FE2_PNGEX_WIDTH, 614
- FE2_RESAMPLE_CHANNELS, 614
- FE2_RESAMPLE_RATE, 614
- FE2_ROTATE_ANGLE, 614
- FE2_SCALE_HEIGHT, 614
- FE2_SCALE_WIDTH, 614
- FE2_SWF_ADAPTIVE_PRELOAD_-
BUFFER_FACTOR, 614
- FE2_SWF_ADD_VARIABLE, 615
- FE2_SWF_DELETE_VARIABLE, 615
- FE2_SWF_EMBEDDED_URL, 615

- FE2_SWF_EMBEDDED_URL_TARGET, 615
- FE2_SWF_EMBEDDED_URL_TYPE, 615
- FE2_SWF_FIXED_PRELOAD_PCT, 615
- FE2_SWF_FRAMERATE, 615
- FE2_SWF_HEIGHT, 615
- FE2_SWF_LOOP_COUNT, 615
- FE2_SWF_ON_END_OPTION, 615
- FE2_SWF_ON_END_URL, 615
- FE2_SWF_ON_START_OPTION, 616
- FE2_SWF_PRELOAD_TYPE, 616
- FE2_SWF_START_BLANK_FRAME, 616
- FE2_SWF_START_WAIT_SEC, 616
- FE2_SWF_WIDTH, 616
- FE2_VCODECPARAM_KFFREQ, 616
- FE2_VCODECPARAM_KFINTTYPE, 616
- FE2_VCODECPARAM_RC_MODE, 616
- FE2_VORBIS_BITRATE, 616
- FE2_VP6_2PASS_MAX_SECTION, 616
- FE2_VP6_2PASS_MIN_SECTION, 616
- FE2_VP6_BITRATE, 617
- FE2_VP6_CONCURRENCY, 617
- FE2_VP6_CXMODE, 617
- FE2_VP6_KFFREQ, 617
- FE2_VP6_KFINTTYPE, 617
- FE2_VP6_MAX_Q, 617
- FE2_VP6_MIN_Q, 617
- FE2_VP6_NOISE_REDUCTION, 617
- FE2_VP6_PROFILE, 617
- FE2_VP6_RC_MODE, 617
- FE2_VP6_SHARPNESS, 617
- FE2_VP6_STREAM_MAX_BUFFER, 618
- FE2_VP6_STREAM_OPTIMAL_BUFFER, 618
- FE2_VP6_STREAM_PEAK_BITRATE, 618
- FE2_VP6_STREAM_PREBUFFER, 618
- FE2_VP6_TEMPORAL_DOWN_-_WATERMARK, 618
- FE2_VP6_TEMPORAL_RESAMPLING, 618
- FE2_VP6_UNDERSHOOT_PCT, 618
- FE2_VP6A_2PASS_MAX_SECTION, 618
- FE2_VP6A_2PASS_MIN_SECTION, 618
- FE2_VP6A_ALPHA_BITRATE, 618
- FE2_VP6A_ALPHA_MAX_Q, 618
- FE2_VP6A_ALPHA_MIN_Q, 619
- FE2_VP6A_ALPHA_NOISE_REDUCTION, 619
- FE2_VP6A_ALPHA_SHARPNESS, 619
- FE2_VP6A_BITRATE, 619
- FE2_VP6A_CXMODE, 619
- FE2_VP6A_KFFREQ, 619
- FE2_VP6A_KFINTTYPE, 619
- FE2_VP6A_MAX_Q, 619
- FE2_VP6A_MIN_Q, 619
- FE2_VP6A_NOISE_REDUCTION, 619
- FE2_VP6A_RC_MODE, 619
- FE2_VP6A_SHARPNESS, 620
- FE2_VP6A_STREAM_MAX_BUFFER, 620
- FE2_VP6A_STREAM_OPTIMAL_BUFFER, 620
- FE2_VP6A_STREAM_PEAK_BITRATE, 620
- FE2_VP6A_STREAM_PREBUFFER, 620
- FE2_VP6A_TEMPORAL_DOWN_-_WATERMARK, 620
- FE2_VP6A_TEMPORAL_RESAMPLING, 620
- FE2_VP6A_UNDERSHOOT_PCT, 620
- FE2_VP8_2PASS_MAX_SECTION, 620
- FE2_VP8_2PASS_MIN_SECTION, 620
- FE2_VP8_ALTREF, 620
- FE2_VP8_AR_MAX_FRAMES, 621
- FE2_VP8_AR_STRENGTH, 621
- FE2_VP8_AR_TYPE, 621
- FE2_VP8_BITRATE, 621
- FE2_VP8_CXMODE, 621
- FE2_VP8_DROP_THRESH, 621
- FE2_VP8_KFFREQ, 621
- FE2_VP8_KFINTTYPE, 621
- FE2_VP8_LAG, 621
- FE2_VP8_MAX_Q, 621
- FE2_VP8_MB_STATIC_THRESHOLD, 621
- FE2_VP8_MIN_Q, 622
- FE2_VP8_NOISE_REDUCTION, 622
- FE2_VP8_OVERSHOOT_PCT, 622
- FE2_VP8_PROFILE, 622
- FE2_VP8_RC_MODE, 622
- FE2_VP8_SHARPNESS, 622
- FE2_VP8_STREAM_INITIAL_BUFFER, 622
- FE2_VP8_STREAM_MAX_BUFFER, 622
- FE2_VP8_STREAM_OPTIMAL_BUFFER, 622
- FE2_VP8_THREADS, 622
- FE2_VP8_TOKEN_PARTITIONS, 622
- FE2_VP8_UNDERSHOOT_PCT, 623
- flixerino, 601
- getEncoderState, 601
- getExportAudio, 601
- getExportVideo, 601
- getExportVideoType, 601
- getLogLevel, 601
- getOverwriteExistingFiles, 601
- getSourceDuration, 601
- isEncoderRunning, 601
- reset, 602
- sc, 623
- setExportAudio, 602

- setExportVideo, 602
- setExportVideoType, 602
- setInputFile, 602
- setLogLevel, 602
- setLogPath, 602
- setOutputFile, 602
- setOverwriteExistingFiles, 602
- stopEncoding, 602
- swfOptions, 602
- syserrno, 603
- validate, 603
- version, 603
- videoOptions, 603
- flxengine_com::IFlixPlgn, 624
 - getParam, 624
 - remove, 624
 - sc, 625
 - setParam, 624
 - setParamAsStr, 625
- flxengine_com::ISwfOptions, 626
 - addVariable, 628
 - deleteVariable, 628
 - getAdaptivePreloaderBufferFactor, 628
 - getEmbeddedUrlType, 628
 - getEnablePreloader, 629
 - getInsertBlankFrameOnStart, 629
 - getLoopCount, 629
 - getMovieOnEndOptions, 629
 - getMovieOnStartOptions, 629
 - getPercentToPreload, 630
 - getPreloaderType, 630
 - getSwfFramerateAsDouble, 630
 - getVariableCount, 630
 - getWaitTimeToStart, 630
 - reset, 631
 - sc, 634
 - setAdaptivePreloaderBufferFactor, 631
 - setEmbeddedUrl, 631
 - setEmbeddedUrlTarget, 631
 - setEmbeddedUrlType, 632
 - setEnablePreloader, 632
 - setInsertBlankFrameOnStart, 632
 - setLoadMovieOnEndUrl, 632
 - setLoopCount, 632
 - setMovieOnEndOptions, 633
 - setMovieOnStartOptions, 633
 - setPercentToPreload, 633
 - setPreloaderType, 633
 - setSwfFramerateAsDouble, 633
 - setWaitTimeToStart, 634
 - updateVariable, 634
- flxengine_com::IVideoOptions, 635
 - addFLVCuePoint, 636
 - addFLVCuePointParameter, 636
 - getDecimateValue, 637
 - getSourceHeight, 637
 - getSourceWidth, 637
 - getSwfHeight, 637
 - getSwfWidth, 637
 - getUseCustomSwfDimensions, 637
 - getUseSourceFramerate, 637
 - getVideoFramerateAsDouble, 638
 - reset, 638
 - sc, 639
 - setDecimateValue, 638
 - setSwfHeight, 638
 - setSwfWidth, 638
 - setUseCustomSwfDimensions, 638
 - setUseSourceFramerate, 639
 - setVideoFramerateAsDouble, 639
 - validate, 639
- FLIXENGINE_VERSION_CHIEF
 - flxengine2.h, 683
- FLIXENGINE_VERSION_EXTRA
 - flxengine2.h, 683
- FLIXENGINE_VERSION_MAJOR
 - flxengine2.h, 683
- FLIXENGINE_VERSION_MINOR
 - flxengine2.h, 683
- FLIXENGINE_VERSION_PATCH
 - flxengine2.h, 683
- FLIXENGINE_VERSION_STR
 - flxengine2.h, 684
- flxerrno
 - flxengine_com::IFlix, 601
- FLV, 505
 - flv_metadata
 - muxer_flv, 508
- FlvAudioMp3
 - audopts_deprecated, 326
- FlvAudioUncompressed
 - audopts_deprecated, 326
- flvmetadata_t
 - muxer_flv, 508
- Frame Rate, 428
- FXM, 510
 - fxmmetadata_t
 - muxer_fxm, 512
- getAdaptivePreloaderBufferFactor
 - flxengine_com::ISwfOptions, 628
- getDecimateValue
 - flxengine_com::IVideoOptions, 637
- getEmbeddedUrlType
 - flxengine_com::ISwfOptions, 628
- getEnablePreloader
 - flxengine_com::ISwfOptions, 629
- getEncoderState

- flxengine_com::IFlix, 601
- getExportAudio
 - flxengine_com::IFlix, 601
- getExportVideo
 - flxengine_com::IFlix, 601
- getExportVideoType
 - flxengine_com::IFlix, 601
- getInsertBlankFrameOnStart
 - flxengine_com::ISwfOptions, 629
- getLogLevel
 - flxengine_com::IFlix, 601
- getLoopCount
 - flxengine_com::ISwfOptions, 629
- getMovieOnEndOptions
 - flxengine_com::ISwfOptions, 629
- getMovieOnStartOptions
 - flxengine_com::ISwfOptions, 629
- getOverwriteExistingFiles
 - flxengine_com::IFlix, 601
- getParam
 - flxengine_com::IFlixPlgn, 624
- getPercentToPreload
 - flxengine_com::ISwfOptions, 630
- getPreloaderType
 - flxengine_com::ISwfOptions, 630
- getSourceDuration
 - flxengine_com::IFlix, 601
- getSourceHeight
 - flxengine_com::IVideoOptions, 637
- getSourceWidth
 - flxengine_com::IVideoOptions, 637
- getSwfFramerateAsDouble
 - flxengine_com::ISwfOptions, 630
- getSwfHeight
 - flxengine_com::IVideoOptions, 637
- getSwfWidth
 - flxengine_com::IVideoOptions, 637
- getUseCustomSwfDimensions
 - flxengine_com::IVideoOptions, 637
- getUseSourceFramerate
 - flxengine_com::IVideoOptions, 637
- getVariableCount
 - flxengine_com::ISwfOptions, 630
- getVideoFramerateAsDouble
 - flxengine_com::IVideoOptions, 638
- getWaitTimeToStart
 - flxengine_com::ISwfOptions, 630
- H263 - FFmpeg, 343
- H264, 346
- h264profile
 - codec_h264, 349
- h264profile_t
 - codec_h264, 349
- Hertz11025
 - audopts_deprecated, 326
- Hertz22050
 - audopts_deprecated, 326
- Hertz44100
 - audopts_deprecated, 326
- HIGH_H264PROFILE
 - codec_h264, 349
- Highpass, 434
- int16_t
 - basetypes, 556
- int32_t
 - basetypes, 556
- INT64_MAX
 - basetypes, 555
- INT64_MIN
 - basetypes, 555
- int64_t
 - basetypes, 556
- int8_t
 - basetypes, 556
- isEncoderRunning
 - flxengine_com::IFlix, 601
- isomedia.h
 - FE2_ISOMEDIA_FASTSTART, 693
- LAME_ABR
 - codec_lame, 352
- LAME_CBR
 - codec_lame, 352
- LAME_VBR_mtrh
 - codec_lame, 352
- LAME_VBR_rh
 - codec_lame, 352
- lame_rcmode
 - codec_lame, 352
- lame_rcmode_t
 - codec_lame, 352
- libvorbis - FFmpeg, 353
- Lowpass, 436
- Main Engine Interface, 477
- MAIN_H264PROFILE
 - codec_h264, 349
- mainengine
 - EncStateIdle, 479
 - EncStateQueued, 479
 - EncStateRunning, 479
 - ErrEncodeA, 479
 - ErrEncodeV, 479
 - ErrFileDecode, 479
 - ErrFileDecodeA, 479
 - ErrFileDecodeV, 479

- ErrFileIO, [479](#)
- ErrFileOpen, [479](#)
- ErrNone, [479](#)
- ErrSys, [479](#)
- ExportFLVVideo, [480](#)
- ExportSWF3To6Video, [480](#)
- ExportSWF7PlusVideo, [480](#)
- ExportSWFVectorVideo, [480](#)
- FE2_EncState, [479](#)
- FE2_errno, [479](#)
- FE2_ExportedVideoType, [479](#)
- Flix2_Copyright, [480](#)
- Flix2_Create, [480](#)
- Flix2_Destroy, [480](#)
- Flix2_Encode, [481](#)
- Flix2_Errno, [481](#)
- Flix2_GetEncoderState, [482](#)
- Flix2_GetExportAudio, [482](#)
- Flix2_GetExportVideo, [482](#)
- Flix2_GetExportVideoType, [483](#)
- Flix2_GetInputFile, [483](#)
- Flix2_GetOutputFile, [484](#)
- Flix2_GetOverwriteExistingFiles, [484](#)
- Flix2_GetSourceDuration, [485](#)
- Flix2_IsEncoderRunning, [485](#)
- Flix2_Reset, [486](#)
- Flix2_SetExportAudio, [486](#)
- Flix2_SetExportVideo, [486](#)
- Flix2_SetExportVideoType, [487](#)
- Flix2_SetInputFile, [487](#)
- Flix2_SetOutputFile, [488](#)
- Flix2_SetOverwriteExistingFiles, [488](#)
- Flix2_StopEncoding, [489](#)
- Flix2_Validate, [489](#)
- Flix2_Version, [490](#)
- mainpage.dox, [710](#)
- MASK_3x3
 - filter_blur, [404](#)
- MASK_5x5
 - filter_blur, [404](#)
- masksiz
 - filter_blur, [404](#)
- masksiz_t
 - filter_blur, [403](#)
- MAX_KEYFRAMES
 - codeccparam_common, [334](#)
- maximumFramesize
 - flxengine_com::IEncodingStatus, [584](#)
- MD_AUDIO_CODECID
 - muxer_flv, [509](#)
- MD_AUDIO_DATARATE
 - muxer_flv, [509](#)
- MD_AUDIO_SIZE
 - muxer_flv, [509](#)
- MD_CANSEEKTOEND
 - muxer_flv, [509](#)
- MD_DATASIZE
 - muxer_flv, [509](#)
- MD_DURATION
 - muxer_flv, [509](#)
- MD_FRAMERATE
 - muxer_flv, [509](#)
- MD_HEIGHT
 - muxer_flv, [509](#)
- MD_KEYFRAMES
 - muxer_flv, [509](#)
- MD_LASTKEYFRAMELOCATION
 - muxer_flv, [509](#)
- MD_LASTKEYFRAMETIMESTAMP
 - muxer_flv, [509](#)
- MD_LASTTIMESTAMP
 - muxer_flv, [509](#)
- MD_VIDEO_CODECID
 - muxer_flv, [509](#)
- MD_VIDEO_DATARATE
 - muxer_flv, [509](#)
- MD_VIDEO_SIZE
 - muxer_flv, [509](#)
- MD_WIDTH
 - muxer_flv, [509](#)
- migration.dox, [711](#)
- minimumFramesize
 - flxengine_com::IEncodingStatus, [584](#)
- Mirror, [438](#)
- MOV - FFmpeg, [517](#)
- MP3 - LAME, [350](#)
- MP4 - FFmpeg, [519](#)
- Muxer Manipulation, [499](#)
- muxer_flv
 - MD_AUDIO_CODECID, [509](#)
 - MD_AUDIO_DATARATE, [509](#)
 - MD_AUDIO_SIZE, [509](#)
 - MD_CANSEEKTOEND, [509](#)
 - MD_DATASIZE, [509](#)
 - MD_DURATION, [509](#)
 - MD_FRAMERATE, [509](#)
 - MD_HEIGHT, [509](#)
 - MD_KEYFRAMES, [509](#)
 - MD_LASTKEYFRAMELOCATION, [509](#)
 - MD_LASTKEYFRAMETIMESTAMP, [509](#)
 - MD_LASTTIMESTAMP, [509](#)
 - MD_VIDEO_CODECID, [509](#)
 - MD_VIDEO_DATARATE, [509](#)
 - MD_VIDEO_SIZE, [509](#)
 - MD_WIDTH, [509](#)
- muxer_swf
 - EmbeddedUrlIsGetUrl, [530](#)
 - EmbeddedUrlIsLoadMovie, [530](#)

- SwfAdaptivePreloader, [531](#)
- SwfFixedPreloader, [531](#)
- SwfOnMovieEndLoadMovie, [531](#)
- SwfOnMovieEndLoop, [530](#)
- SwfOnMovieEndNothing, [530](#)
- SwfOnMovieEndSTOP, [530](#)
- SwfOnMovieEndUnload, [530](#)
- SwfOnMovieStartAutomatically, [531](#)
- SwfOnMovieStartEmbedSTOP, [531](#)
- SwfOnMovieStartOnClick, [531](#)
- SwfOnMovieStartWait, [531](#)
- SwfPreloaderNone, [531](#)
- muxer_3g2lavf
 - FE2_3G2_FASTSTART, [515](#)
 - FE2_MUXER_3G2, [515](#)
- muxer_3gp1avf
 - FE2_3GP_FASTSTART, [513](#)
 - FE2_MUXER_3GP, [513](#)
- muxer_flv
 - FE2_FLV_CUEPT_EVENT, [507](#)
 - FE2_FLV_CUEPT_NAV, [507](#)
 - FE2_FLV_CUEPT_PARAM, [507](#)
 - FE2_FLV_METADATA_DISABLE, [508](#)
 - FE2_FLV_METADATA_ENABLE, [508](#)
 - FE2_MUXER_FLV, [508](#)
 - flv_metadata, [508](#)
 - flvmetadata_t, [508](#)
- muxer_fxm
 - FE2_FXM_CUEPT_EVENT, [511](#)
 - FE2_FXM_CUEPT_NAV, [511](#)
 - FE2_FXM_CUEPT_PARAM, [511](#)
 - FE2_FXM_METADATA_DISABLE, [512](#)
 - FE2_FXM_METADATA_ENABLE, [512](#)
 - FE2_MUXER_FXM, [512](#)
 - fxmmetadata_t, [512](#)
- muxer_movlavf
 - FE2_MOV_FASTSTART, [517](#)
 - FE2_MUXER_MOV, [518](#)
- muxer_mp4lavf
 - FE2_MP4_FASTSTART, [519](#)
 - FE2_MUXER_MP4, [520](#)
- muxer_swf
 - FE2_EmbeddedUrlType, [530](#)
 - FE2_MUXER_SWF, [526](#)
 - FE2_SWF_ADAPTIVE_PRELOAD_ -
BUFFER_FACTOR, [526](#)
 - FE2_SWF_ADD_VARIABLE, [526](#)
 - FE2_SWF_DELETE_VARIABLE, [527](#)
 - FE2_SWF_EMBEDDED_URL, [527](#)
 - FE2_SWF_EMBEDDED_URL_TARGET, [527](#)
 - FE2_SWF_EMBEDDED_URL_TYPE, [527](#)
 - FE2_SWF_FIXED_PRELOAD_PCT, [528](#)
 - FE2_SWF_FRAMERATE, [528](#)
 - FE2_SWF_HEIGHT, [528](#)
 - FE2_SWF_LOOP_COUNT, [528](#)
 - FE2_SWF_ON_END_OPTION, [529](#)
 - FE2_SWF_ON_END_URL, [529](#)
 - FE2_SWF_ON_START_OPTION, [529](#)
 - FE2_SWF_PRELOAD_TYPE, [529](#)
 - FE2_SWF_START_BLANK_FRAME, [529](#)
 - FE2_SWF_START_WAIT_SEC, [530](#)
 - FE2_SWF_WIDTH, [530](#)
 - FE2_SwfOnEndOptions, [530](#)
 - FE2_SwfOnStartOptions, [531](#)
 - FE2_SwfPreloaderOptions, [531](#)
 - swf_options_AddVariable, [531](#)
 - swf_options_DeleteVariable, [532](#)
 - swf_options_ -
GetAdaptivePreloaderBufferFactor, [532](#)
 - swf_options_GetEmbeddedUrl, [533](#)
 - swf_options_GetEmbeddedUrlTarget, [534](#)
 - swf_options_GetEmbeddedUrlType, [534](#)
 - swf_options_GetEnablePreloader, [535](#)
 - swf_options_GetInsertBlankFrameOnStart, [535](#)
 - swf_options_GetLoadMovieOnEndUrl, [536](#)
 - swf_options_GetLoopCount, [536](#)
 - swf_options_GetMovieOnEndOptions, [537](#)
 - swf_options_GetMovieOnStartOptions, [537](#)
 - swf_options_GetPercentToPreload, [538](#)
 - swf_options_GetPreloaderType, [538](#)
 - swf_options_GetSwfFramerate, [539](#)
 - swf_options_GetSwfFramerateAsDouble, [539](#)
 - swf_options_GetVariableCount, [540](#)
 - swf_options_GetWaitTimeToStart, [540](#)
 - swf_options_Reset, [541](#)
 - swf_options_ -
SetAdaptivePreloaderBufferFactor, [541](#)
 - swf_options_SetEmbeddedUrl, [542](#)
 - swf_options_SetEmbeddedUrlTarget, [543](#)
 - swf_options_SetEmbeddedUrlType, [544](#)
 - swf_options_SetEnablePreloader, [545](#)
 - swf_options_SetInsertBlankFrameOnStart, [545](#)
 - swf_options_SetLoadMovieOnEndUrl, [546](#)
 - swf_options_SetLoopCount, [546](#)
 - swf_options_SetMovieOnEndOptions, [547](#)
 - swf_options_SetMovieOnStartOptions, [547](#)
 - swf_options_SetPercentToPreload, [548](#)
 - swf_options_SetPreloaderType, [549](#)
 - swf_options_SetSwfFramerate, [549](#)
 - swf_options_SetSwfFramerateAsDouble, [550](#)
 - swf_options_SetWaitTimeToStart, [550](#)
 - swf_options_UpdateVariable, [551](#)
- muxer_webmlavf

- FE2_MUXER_WEBM, 553
- muxerfunc
 - Flix2_AddMuxer, 499
 - Flix2_MuxerGetParam, 500
 - Flix2_MuxerGetParamAsStr, 500
 - Flix2_MuxerSetParam, 501
 - Flix2_MuxerSetParamAsStr, 501
 - Flix2_RemoveMuxer, 501
- Muxers, 504
- ON264
 - basetypes, 555
- ON2_BUFFER_EMPTY
 - basetypes, 558
- ON2_BUFFER_FULL
 - basetypes, 558
- ON2_CONNREFUSED
 - basetypes, 558
- ON2_DONE
 - basetypes, 558
- ON2_ERROR
 - basetypes, 558
- ON2_INPROGRESS
 - basetypes, 558
- ON2_INVALID_PARAMS
 - basetypes, 558
- ON2_INVALID_VERSION
 - basetypes, 558
- ON2_NET_ERROR
 - basetypes, 558
- ON2_NO_MEM
 - basetypes, 558
- ON2_NOT_FOUND
 - basetypes, 558
- ON2_NOT_SUPP
 - basetypes, 558
- ON2_OK
 - basetypes, 558
- ON2_TIMEOUT
 - basetypes, 558
- ON2_WOULDBLOCK
 - basetypes, 558
- ON2API
 - on2types.h, 699
- on2bool
 - basetypes, 556
- on2false
 - basetypes, 557
- on2s16
 - basetypes, 556
- on2s32
 - basetypes, 556
- on2s64
 - basetypes, 556
- on2s8
 - basetypes, 556
- on2sc
 - basetypes, 558
- ON2TC
 - basetypes, 555
- on2tc
 - basetypes, 557
- on2true
 - basetypes, 557
- on2types.h
 - DLLEXPORT, 699
 - DLLIMPORT, 699
 - DLLLOCAL, 699
 - ON2API, 699
- on2u16
 - basetypes, 557
- on2u32
 - basetypes, 557
- on2u64
 - basetypes, 557
- on2u8
 - basetypes, 557
- OTC
 - basetypes, 555
- Overlay (Watermark), 440
- overlay_options_GetMaskPixelRGB
 - filter_overlay, 445
- overlay_options_GetMaskPixelXY
 - filter_overlay, 446
- overlay_options_GetOverlayPath
 - filter_overlay, 447
- overlay_options_GetOverlayPosition
 - filter_overlay, 447
- overlay_options_GetUseOverlay
 - filter_overlay, 448
- overlay_options_Reset
 - filter_overlay, 448
- overlay_options_SetMaskPixelRGB
 - filter_overlay, 449
- overlay_options_SetMaskPixelXY
 - filter_overlay, 450
- overlay_options_SetOverlayPath
 - filter_overlay, 450
- overlay_options_SetOverlayPosition
 - filter_overlay, 451
- overlay_options_SetUseOverlay
 - filter_overlay, 452
- percentComplete
 - flixengine_com::IEncodingStatus, 584
- PNG Image Export (Thumbnail), 453
- PRId64
 - basetypes, 555

- PRIu64
 - basetypes, 556
- remove
 - flxengine_com::IFlixPlgn, 624
- Resample, 461
- reset
 - flxengine_com::IFlix, 602
 - flxengine_com::ISwfOptions, 631
 - flxengine_com::IVideoOptions, 638
- Rotate, 462
- samples.dox, 712
- sc
 - flxengine_com::IEncodingStatus, 585
 - flxengine_com::IFlix, 623
 - flxengine_com::IFlixPlgn, 625
 - flxengine_com::ISwfOptions, 634
 - flxengine_com::IVideoOptions, 639
- Scale, 463
- setAdaptivePreloaderBufferFactor
 - flxengine_com::ISwfOptions, 631
- setDecimateValue
 - flxengine_com::IVideoOptions, 638
- setEmbeddedUrl
 - flxengine_com::ISwfOptions, 631
- setEmbeddedUrlTarget
 - flxengine_com::ISwfOptions, 631
- setEmbeddedUrlType
 - flxengine_com::ISwfOptions, 632
- setEnablePreloader
 - flxengine_com::ISwfOptions, 632
- setExportAudio
 - flxengine_com::IFlix, 602
- setExportVideo
 - flxengine_com::IFlix, 602
- setExportVideoType
 - flxengine_com::IFlix, 602
- setInputFile
 - flxengine_com::IFlix, 602
- setInsertBlankFrameOnStart
 - flxengine_com::ISwfOptions, 632
- setLoadMovieOnEndUrl
 - flxengine_com::ISwfOptions, 632
- setLogLevel
 - flxengine_com::IFlix, 602
- setLogPath
 - flxengine_com::IFlix, 602
- setLoopCount
 - flxengine_com::ISwfOptions, 632
- setMovieOnEndOptions
 - flxengine_com::ISwfOptions, 633
- setMovieOnStartOptions
 - flxengine_com::ISwfOptions, 633
- setOutputFile
 - flxengine_com::IFlix, 602
- setOverwriteExistingFiles
 - flxengine_com::IFlix, 602
- setParam
 - flxengine_com::IFlixPlgn, 624
- setParamAsStr
 - flxengine_com::IFlixPlgn, 625
- setPercentToPreload
 - flxengine_com::ISwfOptions, 633
- setPreloaderType
 - flxengine_com::ISwfOptions, 633
- setSwfFramerateAsDouble
 - flxengine_com::ISwfOptions, 633
- setSwfHeight
 - flxengine_com::IVideoOptions, 638
- setSwfWidth
 - flxengine_com::IVideoOptions, 638
- setUseCustomSwfDimensions
 - flxengine_com::IVideoOptions, 638
- setUseSourceFramerate
 - flxengine_com::IVideoOptions, 639
- setVideoFramerateAsDouble
 - flxengine_com::IVideoOptions, 639
- setWaitTimeToStart
 - flxengine_com::ISwfOptions, 634
- Sharpen, 475
- startTime
 - flxengine_com::IEncodingStatus, 584
- stopEncoding
 - flxengine_com::IFlix, 602
- SWF, 521
 - swf_options_AddVariable
 - muxer_swf, 531
 - swf_options_DeleteVariable
 - muxer_swf, 532
 - swf_options_GetAdaptivePreloaderBufferFactor
 - muxer_swf, 532
 - swf_options_GetEmbeddedUrl
 - muxer_swf, 533
 - swf_options_GetEmbeddedUrlTarget
 - muxer_swf, 534
 - swf_options_GetEmbeddedUrlType
 - muxer_swf, 534
 - swf_options_GetEnablePreloader
 - muxer_swf, 535
 - swf_options_GetInsertBlankFrameOnStart
 - muxer_swf, 535
 - swf_options_GetLoadMovieOnEndUrl
 - muxer_swf, 536
 - swf_options_GetLoopCount
 - muxer_swf, 536
 - swf_options_GetMovieOnEndOptions
 - muxer_swf, 537

- swf_options_GetMovieOnStartOptions
 - muxer_swf, [537](#)
- swf_options_GetPercentToPreload
 - muxer_swf, [538](#)
- swf_options_GetPreloaderType
 - muxer_swf, [538](#)
- swf_options_GetSwfFramerate
 - muxer_swf, [539](#)
- swf_options_GetSwfFramerateAsDouble
 - muxer_swf, [539](#)
- swf_options_GetVariableCount
 - muxer_swf, [540](#)
- swf_options_GetWaitTimeToStart
 - muxer_swf, [540](#)
- swf_options_Reset
 - muxer_swf, [541](#)
- swf_options_SetAdaptivePreloaderBufferFactor
 - muxer_swf, [541](#)
- swf_options_SetEmbeddedUrl
 - muxer_swf, [542](#)
- swf_options_SetEmbeddedUrlTarget
 - muxer_swf, [543](#)
- swf_options_SetEmbeddedUrlType
 - muxer_swf, [544](#)
- swf_options_SetEnablePreloader
 - muxer_swf, [545](#)
- swf_options_SetInsertBlankFrameOnStart
 - muxer_swf, [545](#)
- swf_options_SetLoadMovieOnEndUrl
 - muxer_swf, [546](#)
- swf_options_SetLoopCount
 - muxer_swf, [546](#)
- swf_options_SetMovieOnEndOptions
 - muxer_swf, [547](#)
- swf_options_SetMovieOnStartOptions
 - muxer_swf, [547](#)
- swf_options_SetPercentToPreload
 - muxer_swf, [548](#)
- swf_options_SetPreloaderType
 - muxer_swf, [549](#)
- swf_options_SetSwfFramerate
 - muxer_swf, [549](#)
- swf_options_SetSwfFramerateAsDouble
 - muxer_swf, [550](#)
- swf_options_SetWaitTimeToStart
 - muxer_swf, [550](#)
- swf_options_UpdateVariable
 - muxer_swf, [551](#)
- SwfAdaptivePreloader
 - muxer_swf, [531](#)
- SwfFixedPreloader
 - muxer_swf, [531](#)
- SwfOnMovieEndLoadMovie
 - muxer_swf, [531](#)
- SwfOnMovieEndLoop
 - muxer_swf, [530](#)
- SwfOnMovieEndNothing
 - muxer_swf, [530](#)
- SwfOnMovieEndSTOP
 - muxer_swf, [530](#)
- SwfOnMovieEndUnload
 - muxer_swf, [530](#)
- SwfOnMovieStartAutomatically
 - muxer_swf, [531](#)
- SwfOnMovieStartEmbedSTOP
 - muxer_swf, [531](#)
- SwfOnMovieStartOnClick
 - muxer_swf, [531](#)
- SwfOnMovieStartWait
 - muxer_swf, [531](#)
- swfOptions
 - flxengine_com::IFlix, [602](#)
- SwfPreloaderNone
 - muxer_swf, [531](#)
- syserrno
 - flxengine_com::IFlix, [603](#)
- totalFrames
 - flxengine_com::IEncodingStatus, [585](#)
- uint16_t
 - basetypes, [557](#)
- uint32_t
 - basetypes, [557](#)
- UINT64_MAX
 - basetypes, [556](#)
- uint64_t
 - basetypes, [557](#)
- uint8_t
 - basetypes, [557](#)
- updateVariable
 - flxengine_com::ISwfOptions, [634](#)
- validate
 - flxengine_com::IFlix, [603](#)
 - flxengine_com::IVideoOptions, [639](#)
- VBR_1PASSControl
 - codeccparam_common, [334](#)
- VBR_2PASSControl
 - codeccparam_common, [334](#)
- version
 - flxengine_com::IFlix, [603](#)
- Video Codecs, [335](#)
- Video Encoding Options, [559](#)
- Video Filters, [387](#)
- video_options_AddFLVCuePoint
 - vidopts_deprecated, [566](#)
- video_options_AddFLVCuePointParameter

- vidopts_deprecated, [567](#)
- video_options_GetAlphaPercentage
 - vidopts_deprecated, [568](#)
- video_options_GetCompressMode
 - vidopts_deprecated, [568](#)
- video_options_GetDecimateValue
 - filter_framerate, [429](#)
 - filter_scale, [466](#)
- video_options_GetDeinterlace
 - filter_adaptive_deinterlace, [390](#)
 - filter_scale, [466](#)
- video_options_GetImageHeight
 - filter_scale, [467](#)
- video_options_GetImageQuality
 - vidopts_deprecated, [568](#)
- video_options_GetImageWidth
 - filter_scale, [467](#)
- video_options_GetKeyframeInterval
 - vidopts_deprecated, [569](#)
- video_options_GetKeyframeIntervalType
 - vidopts_deprecated, [569](#)
- video_options_GetMaximumBitrate
 - vidopts_deprecated, [570](#)
- video_options_GetRateControlType
 - vidopts_deprecated, [570](#)
- video_options_GetSourceHeight
 - vidopts, [559](#)
- video_options_GetSourceWidth
 - vidopts, [560](#)
- video_options_GetSwfFramerate
 - vidopts_deprecated, [571](#)
- video_options_GetSwfHeight
 - vidopts, [560](#)
- video_options_GetSwfWidth
 - vidopts, [560](#)
- video_options_GetUseCustomSwfDimensions
 - vidopts, [561](#)
- video_options_GetUseMaximumBitrate
 - vidopts_deprecated, [571](#)
- video_options_GetUseSourceDimensions
 - filter_scale, [468](#)
- video_options_GetUseSourceFramerate
 - filter_framerate, [430](#)
 - filter_scale, [468](#)
- video_options_GetVideoCodec
 - vidopts_deprecated, [571](#)
- video_options_GetVideoFramerate
 - filter_framerate, [430](#)
 - filter_scale, [469](#)
- video_options_GetVideoFramerateAsDouble
 - filter_framerate, [431](#)
 - filter_scale, [469](#)
- video_options_Reset
 - vidopts, [561](#)
- video_options_SetAlphaPercentage
 - vidopts_deprecated, [572](#)
- video_options_SetCompressMode
 - vidopts_deprecated, [572](#)
- video_options_SetDecimateValue
 - filter_framerate, [431](#)
 - filter_scale, [470](#)
- video_options_SetDeinterlace
 - filter_adaptive_deinterlace, [391](#)
 - filter_scale, [470](#)
- video_options_SetImageHeight
 - filter_scale, [470](#)
- video_options_SetImageQuality
 - vidopts_deprecated, [573](#)
- video_options_SetImageWidth
 - filter_scale, [471](#)
- video_options_SetKeyframeInterval
 - vidopts_deprecated, [573](#)
- video_options_SetKeyframeIntervalType
 - vidopts_deprecated, [574](#)
- video_options_SetMaximumBitrate
 - vidopts_deprecated, [574](#)
- video_options_SetRateControlType
 - vidopts_deprecated, [575](#)
- video_options_SetSwfFramerate
 - vidopts_deprecated, [575](#)
- video_options_SetSwfHeight
 - vidopts, [561](#)
- video_options_SetSwfWidth
 - vidopts, [562](#)
- video_options_SetUseCustomSwfDimensions
 - vidopts, [562](#)
- video_options_SetUseMaximumBitrate
 - vidopts_deprecated, [576](#)
- video_options_SetUseSourceDimensions
 - filter_scale, [472](#)
- video_options_SetUseSourceFramerate
 - filter_framerate, [432](#)
 - filter_scale, [472](#)
- video_options_SetVideoCodec
 - vidopts_deprecated, [576](#)
- video_options_SetVideoFramerate
 - filter_framerate, [432](#)
 - filter_scale, [473](#)
- video_options_SetVideoFramerateAsDouble
 - filter_framerate, [432](#)
 - filter_scale, [473](#)
- video_options_Validate
 - vidopts, [562](#)
- videoOptions
 - flixengine_com::IFlix, [603](#)
- vidopts
 - video_options_GetSourceHeight, [559](#)
 - video_options_GetSourceWidth, [560](#)

- video_options_GetSwfHeight, [560](#)
- video_options_GetSwfWidth, [560](#)
- video_options_
 - GetUseCustomSwfDimensions, [561](#)
- video_options_Reset, [561](#)
- video_options_SetSwfHeight, [561](#)
- video_options_SetSwfWidth, [562](#)
- video_options_
 - SetUseCustomSwfDimensions, [562](#)
- video_options_Validate, [562](#)
- vidopts_deprecated
 - CODEC_H263, [566](#)
 - CODEC_NULL, [566](#)
 - CODEC_SCREENVIDEO, [566](#)
 - CODEC_VP6, [566](#)
 - CODEC_VP6ALPHA, [566](#)
 - CUE_EVENT, [566](#)
 - CUE_NAVIGATION, [566](#)
- vidopts_deprecated
 - FE2_CuePointType, [566](#)
 - FE2_VideoCodec, [566](#)
 - video_options_AddFLVCuePoint, [566](#)
 - video_options_AddFLVCuePointParameter, [567](#)
 - video_options_GetAlphaPercentage, [568](#)
 - video_options_GetCompressMode, [568](#)
 - video_options_GetImageQuality, [568](#)
 - video_options_GetKeyframeInterval, [569](#)
 - video_options_GetKeyframeIntervalType, [569](#)
 - video_options_GetMaximumBitrate, [570](#)
 - video_options_GetRateControlType, [570](#)
 - video_options_GetSwfFramerate, [571](#)
 - video_options_GetUseMaximumBitrate, [571](#)
 - video_options_GetVideoCodec, [571](#)
 - video_options_SetAlphaPercentage, [572](#)
 - video_options_SetCompressMode, [572](#)
 - video_options_SetImageQuality, [573](#)
 - video_options_SetKeyframeInterval, [573](#)
 - video_options_SetKeyframeIntervalType, [574](#)
 - video_options_SetMaximumBitrate, [574](#)
 - video_options_SetRateControlType, [575](#)
 - video_options_SetSwfFramerate, [575](#)
 - video_options_SetUseMaximumBitrate, [576](#)
 - video_options_SetVideoCodec, [576](#)
- VP6, [354](#)
- VP6 with Alpha, [364](#)
- VP6_E
 - codec_vp6, [363](#)
- VP6_S
 - codec_vp6, [363](#)
- vp6profile
 - codec_vp6, [363](#)
- vp6profile_t
 - codec_vp6, [362](#)
- VP8, [372](#)
- warning
 - flixengine_com.idl, [709](#)
- WebM - FFmpeg, [553](#)