FFmpeg Codecs Documentation

Table of Contents

1. Description

This document describes the codecs (decoders and encoders) provided by the libavcodec library.

2. Codec Options

libavcodec provides some generic global options, which can be set on all the encoders and decoders. In addition each codec may support so-called private options, which are specific for a given codec.

Sometimes, a global option may only affect a specific kind of codec, and may be unsensical or ignored by another, so you need to be aware of the meaning of the specified options. Also some options are meant only for decoding or encoding.

Options may be set by specifying -*option value* in the FFmpeg tools, or by setting the value explicitly in the `AVCodecContext` options or using the ‘`libavutil/opt.h`’ API for programmatic use.

The list of supported options follow:

‘`b` *`integer`* `(`*`encoding,audio,video`*`)`’

Set bitrate in bits/s. Default value is 200K.

‘`ab` *`integer`* `(`*`encoding,audio`*`)`’

Set audio bitrate (in bits/s). Default value is 128K.

‘`bt` *`integer`* `(`*`encoding,video`*`)`’

Set video bitrate tolerance (in bits/s). In 1-pass mode, bitrate tolerance specifies how far ratecontrol is willing to deviate from the target average bitrate value. This is not related to min/max bitrate. Lowering tolerance too much has an adverse effect on quality.

‘`flags` *`flags`* `(`*`decoding/encoding,audio,video,subtitles`*`)`’

Set generic flags.

Possible values:

‘`mv4`’

Use four motion vector by macroblock (mpeg4).

‘`qpel`’

Use 1/4 pel motion compensation.

‘`loop`’

Use loop filter.

‘`qscale`’

Use fixed qscale.

‘`gmc`’

Use gmc.

‘`mv0`’

Always try a mb with mv=<0,0>.

‘`input_preserved`’
‘`pass1`’

Use internal 2pass ratecontrol in first pass mode.

‘`pass2`’

Use internal 2pass ratecontrol in second pass mode.

‘`gray`’

Only decode/encode grayscale.

‘`emu_edge`’

Do not draw edges.

‘`psnr`’

Set error[?] variables during encoding.

‘`truncated`’
‘`naq`’

Normalize adaptive quantization.

‘`ildct`’

Use interlaced DCT.

‘`low_delay`’

Force low delay.

‘`global_header`’

Place global headers in extradata instead of every keyframe.

‘`bitexact`’

Use only bitexact stuff (except (I)DCT).

‘`aic`’

Apply H263 advanced intra coding / mpeg4 ac prediction.

‘`cbp`’

Deprecated, use mpegvideo private options instead.

‘`qprd`’

Deprecated, use mpegvideo private options instead.

‘`ilme`’

Apply interlaced motion estimation.

‘`cgop`’

Use closed gop.

‘`sub_id` *`integer`*’

Deprecated, currently unused.

‘`me_method` *`integer`* `(`*`encoding,video`*`)`’

Set motion estimation method.

Possible values:

‘`zero`’

zero motion estimation (fastest)

‘`full`’

full motion estimation (slowest)

‘`epzs`’

EPZS motion estimation (default)

‘`esa`’

esa motion estimation (alias for full)

‘`tesa`’

tesa motion estimation

‘`dia`’

dia motion estimation (alias for epzs)

‘`log`’

log motion estimation

‘`phods`’

phods motion estimation

‘`x1`’

X1 motion estimation

‘`hex`’

hex motion estimation

‘`umh`’

umh motion estimation

‘`iter`’

iter motion estimation

‘`extradata_size` *`integer`*’

Set extradata size.

‘`time_base` *`rational number`*’

Set codec time base.

It is the fundamental unit of time (in seconds) in terms of which frame timestamps are represented. For fixed-fps content, timebase should be `1 / frame_rate` and timestamp increments should be identically 1.

‘`g` *`integer`* (*`encoding,video`*)’

Set the group of picture size. Default value is 12.

‘`ar` *`integer`* (*`decoding/encoding,audio`*)’

Set audio sampling rate (in Hz).

‘`ac` *`integer`* (*`decoding/encoding,audio`*)’

Set number of audio channels.

‘`cutoff` *`integer`* (*`encoding,audio`*)’

Set cutoff bandwidth.

‘`frame_size` *`integer`* (*`encoding,audio`*)’

Set audio frame size.

Each submitted frame except the last must contain exactly frame_size samples per channel. May be 0 when the codec has CODEC_CAP_VARIABLE_FRAME_SIZE set, in that case the frame size is not restricted. It is set by some decoders to indicate constant frame size.

‘`frame_number` *integer*’

Set the frame number.

‘`delay` *integer*’
‘`qcomp` *float* (*encoding,video*)’

Set video quantizer scale compression (VBR). It is used as a constant in the ratecontrol equation. Recommended range for default rc_eq: 0.0-1.0.

‘`qblur` *float* (*encoding,video*)’

Set video quantizer scale blur (VBR).

‘`qmin` *integer* (*encoding,video*)’

Set min video quantizer scale (VBR). Must be included between -1 and 69, default value is 2.

‘`qmax` *integer* (*encoding,video*)’

Set max video quantizer scale (VBR). Must be included between -1 and 1024, default value is 31.

‘`qdiff` *integer* (*encoding,video*)’

Set max difference between the quantizer scale (VBR).

‘`bf` *integer* (*encoding,video*)’

Set max number of B frames.

‘`b_qfactor` *float* (*encoding,video*)’

Set qp factor between P and B frames.

‘`rc_strategy` *integer* (*encoding,video*)’

Set ratecontrol method.

‘`b_strategy` *integer* (*encoding,video*)’

Set strategy to choose between I/P/B-frames.

‘`ps` *integer* (*encoding,video*)’

Set RTP payload size in bytes.

‘`mv_bits` *integer*’
‘`header_bits` *integer*’

‘`i_tex_bits` *integer*’
‘`p_tex_bits` *integer*’
‘`i_count` *integer*’
‘`p_count` *integer*’
‘`skip_count` *integer*’
‘`misc_bits` *integer*’
‘`frame_bits` *integer*’
‘`codec_tag` *integer*’
‘`bug` *flags* (*decoding,video*)’

Workaround not auto detected encoder bugs.

Possible values:

‘`autodetect`’
‘`old_msmpeg4`’

some old lavc generated msmpeg4v3 files (no autodetection)

‘`xvid_ilace`’

Xvid interlacing bug (autodetected if fourcc==XVIX)

‘`ump4`’

(autodetected if fourcc==UMP4)

‘`no_padding`’

padding bug (autodetected)

‘`amv`’
‘`ac_vlc`’

illegal vlc bug (autodetected per fourcc)

‘`qpel_chroma`’
‘`std_qpel`’

old standard qpel (autodetected per fourcc/version)

‘`qpel_chroma2`’
‘`direct_blocksize`’

direct-qpel-blocksize bug (autodetected per fourcc/version)

‘`edge`’

edge padding bug (autodetected per fourcc/version)

‘`hpel_chroma`’
‘`dc_clip`’
‘`ms`’

Workaround various bugs in microsoft broken decoders.

‘`trunc`’

trancated frames

‘`lelim *integer* (*encoding,video*)`’

Set single coefficient elimination threshold for luminance (negative values also consider DC coefficient).

‘`celim *integer* (*encoding,video*)`’

Set single coefficient elimination threshold for chrominance (negative values also consider dc coefficient)

‘`strict *integer* (*decoding/encoding,audio,video*)`’

Specify how strictly to follow the standards.

Possible values:

‘`very`’

strictly conform to a older more strict version of the spec or reference software

‘`strict`’

strictly conform to all the things in the spec no matter what consequences

‘`normal`’
‘`unofficial`’

allow unofficial extensions

‘`experimental`’

allow non standardized experimental things, experimental (unfinished/work in progress/not well tested) decoders and encoders. Note: experimental decoders can pose a security risk, do not use this for decoding untrusted input.

‘`b_qoffset *float* (*encoding,video*)`’

Set QP offset between P and B frames.

‘`err_detect` *`flags`* `(`*`decoding,audio,video`*`)`’

Set error detection flags.

Possible values:

‘`crccheck`’

verify embedded CRCs

‘`bitstream`’

detect bitstream specification deviations

‘`buffer`’

detect improper bitstream length

‘`explode`’

abort decoding on minor error detection

‘`careful`’

consider things that violate the spec and have not been seen in the wild as errors

‘`compliant`’

consider all spec non compliancies as errors

‘`aggressive`’

consider things that a sane encoder should not do as an error

‘`has_b_frames` *`integer`*’
‘`block_align` *`integer`*’
‘`mpeg_quant` *`integer`* `(`*`encoding,video`*`)`’

Use MPEG quantizers instead of H.263.

‘`qsquish` *`float`* `(`*`encoding,video`*`)`’

How to keep quantizer between qmin and qmax (0 = clip, 1 = use differentiable function).

‘`rc_qmod_amp` *`float`* `(`*`encoding,video`*`)`’

Set experimental quantizer modulation.

‘`rc_qmod_freq` *integer* (*encoding,video*)’

Set experimental quantizer modulation.

‘`rc_override_count` *integer*’
‘`rc_eq` *string* (*encoding,video*)’

Set rate control equation. When computing the expression, besides the standard functions defined in the section ’Expression Evaluation’, the following functions are available: bits2qp(bits), qp2bits(qp). Also the following constants are available: iTex pTex tex mv fCode iCount mcVar var isI isP isB avgQP qComp avgIITex avgPITex avgPPTex avgBPTex avgTex.

‘`maxrate` *integer* (*encoding,audio,video*)’

Set max bitrate tolerance (in bits/s). Requires bufsize to be set.

‘`minrate` *integer* (*encoding,audio,video*)’

Set min bitrate tolerance (in bits/s). Most useful in setting up a CBR encode. It is of little use elsewise.

‘`bufsize` *integer* (*encoding,audio,video*)’

Set ratecontrol buffer size (in bits).

‘`rc_buf_aggressivity` *float* (*encoding,video*)’

Currently useless.

‘`i_qfactor` *float* (*encoding,video*)’

Set QP factor between P and I frames.

‘`i_qoffset` *float* (*encoding,video*)’

Set QP offset between P and I frames.

‘`rc_init_cplx` *float* (*encoding,video*)’

Set initial complexity for 1-pass encoding.

‘`dct` *integer* (*encoding,video*)’

Set DCT algorithm.

Possible values:

‘`auto`’

autoselect a good one (default)

‘`fastint`’

fast integer

‘`int`’

accurate integer

‘`mmx`’
‘`altivec`’
‘`faan`’

floating point AAN DCT

‘`lumi_mask` *float* (*encoding,video*)’

Compress bright areas stronger than medium ones.

‘`tcplx_mask` *float* (*encoding,video*)’

Set temporal complexity masking.

‘`scplx_mask` *float* (*encoding,video*)’

Set spatial complexity masking.

‘`p_mask` *float* (*encoding,video*)’

Set inter masking.

‘`dark_mask` *float* (*encoding,video*)’

Compress dark areas stronger than medium ones.

‘`idct` *integer* (*decoding/encoding,video*)’

Select IDCT implementation.

Possible values:

‘`auto`’
‘`int`’
‘`simple`’
‘`simplemmx`’

‘`libmpeg2mmx`’
‘`mmi`’
‘`arm`’
‘`altivec`’
‘`sh4`’
‘`simplearm`’
‘`simplearmv5te`’
‘`simplearmv6`’
‘`simpleneon`’
‘`simplealpha`’
‘`h264`’
‘`vp3`’
‘`ipp`’
‘`xvidmmx`’
‘`faani`’

floating point AAN IDCT

‘`slice_count` *`integer`*’
‘`ec` *`flags`* `(`*`decoding,video`*`)`’

Set error concealment strategy.

Possible values:

‘`guess_mvs`’

iterative motion vector (MV) search (slow)

‘`deblock`’

use strong deblock filter for damaged MBs

‘`bits_per_coded_sample` *`integer`*’
‘`pred` *`integer`* `(`*`encoding,video`*`)`’

Set prediction method.

Possible values:

‘`left`’
‘`plane`’
‘`median`’
‘`aspect` *`rational number`* `(`*`encoding,video`*`)`’

Set sample aspect ratio.

‘`debug` *`flags`* `(`*`decoding/encoding,audio,video,subtitles`*`)`’

Print specific debug info.

Possible values:

‘`pict`’

picture info

‘`rc`’

rate control

‘`bitstream`’
‘`mb_type`’

macroblock (MB) type

‘`qp`’

per-block quantization parameter (QP)

‘`mv`’

motion vector

‘`dct_coeff`’
‘`skip`’
‘`startcode`’
‘`pts`’
‘`er`’

error recognition

‘`mmco`’

memory management control operations (H.264)

‘`bugs`’
‘`vis_qp`’

visualize quantization parameter (QP), lower QP are tinted greener

‘`vis_mb_type`’

visualize block types

‘`buffers`’

picture buffer allocations

‘`thread_ops`’

threading operations

‘`vismv` *`integer`* `(`*`decoding,video`*`)`’

Visualize motion vectors (MVs).

Possible values:

‘`pf`’

forward predicted MVs of P-frames

‘`bf`’

forward predicted MVs of B-frames

‘`bb`’

backward predicted MVs of B-frames

‘`cmp` *`integer`* `(`*`encoding,video`*`)`’

Set full pel me compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’
‘`subcmp` *`integer`* `(`*`encoding,video`*`)`’

Set sub pel me compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘satd’

sum of absolute Hadamard transformed differences

‘dct’

sum of absolute DCT transformed differences

‘psnr’

sum of squared quantization errors (avoid, low quality)

‘bit’

number of bits needed for the block

‘rd’

rate distortion optimal, slow

‘zero’

0

‘vsad’

sum of absolute vertical differences

‘vsse’

sum of squared vertical differences

‘nsse’

noise preserving sum of squared differences

‘w53’

5/3 wavelet, only used in snow

‘w97’

9/7 wavelet, only used in snow

‘dctmax’
‘chroma’

‘`mbcmp` *`integer`* `(`*`encoding,video`*`)`’

Set macroblock compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘w53’

5/3 wavelet, only used in snow

‘w97’

9/7 wavelet, only used in snow

‘dctmax’
‘chroma’

‘ildctcmp *integer* (*encoding,video*)’

Set interlaced dct compare function.

Possible values:

‘sad’

sum of absolute differences, fast (default)

‘sse’

sum of squared errors

‘satd’

sum of absolute Hadamard transformed differences

‘dct’

sum of absolute DCT transformed differences

‘psnr’

sum of squared quantization errors (avoid, low quality)

‘bit’

number of bits needed for the block

‘rd’

rate distortion optimal, slow

‘zero’

0

‘vsad’

sum of absolute vertical differences

‘vsse’

sum of squared vertical differences

‘nsse’

noise preserving sum of squared differences

‘w53’

5/3 wavelet, only used in snow

‘w97’

9/7 wavelet, only used in snow

‘dctmax’
‘chroma’
‘dia_size *integer* (*encoding,video*)’

Set diamond type & size for motion estimation.

‘last_pred *integer* (*encoding,video*)’

Set amount of motion predictors from the previous frame.

‘preme *integer* (*encoding,video*)’

Set pre motion estimation.

‘precmp *integer* (*encoding,video*)’

Set pre motion estimation compare function.

Possible values:

‘sad’

sum of absolute differences, fast (default)

‘sse’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’

‘`pre_dia_size` *integer* (*encoding,video*)’

Set diamond type & size for motion estimation pre-pass.

‘`subq` *integer* (*encoding,video*)’

Set sub pel motion estimation quality.

‘`dtg_active_format` *integer*’
‘`me_range` *integer* (*encoding,video*)’

Set limit motion vectors range (1023 for DivX player).

‘`ibias` *integer* (*encoding,video*)’

Set intra quant bias.

‘`pbias` *integer* (*encoding,video*)’

Set inter quant bias.

‘`color_table_id` *integer*’
‘`global_quality` *integer* (*encoding,audio,video*)’
‘`coder` *integer* (*encoding,video*)’

Possible values:

‘`vlc`’

variable length coder / huffman coder

‘`ac`’

arithmetic coder

‘`raw`’

raw (no encoding)

‘`rle`’

run-length coder

‘`deflate`’

deflate-based coder

‘`context` *integer* (*encoding,video*)’

Set context model.

'`slice_flags` *integer*'
'`xvmc_acceleration` *integer*'
'`mbd` *integer* (*encoding,video*)'

Set macroblock decision algorithm (high quality mode).

Possible values:

'`simple`'

use mbcmp (default)

'`bits`'

use fewest bits

'`rd`'

use best rate distortion

'`stream_codec_tag` *integer*'
'`sc_threshold` *integer* (*encoding,video*)'

Set scene change threshold.

'`lmin` *integer* (*encoding,video*)'

Set min lagrange factor (VBR).

'`lmax` *integer* (*encoding,video*)'

Set max lagrange factor (VBR).

'`nr` *integer* (*encoding,video*)'

Set noise reduction.

'`rc_init_occupancy` *integer* (*encoding,video*)'

Set number of bits which should be loaded into the rc buffer before decoding starts.

'`inter_threshold` *integer* (*encoding,video*)'
'`flags2` *flags* (*decoding/encoding,audio,video*)'

Possible values:

‘`fast`’

allow non spec compliant speedup tricks

‘`sgop`’

Deprecated, use mpegvideo private options instead

‘`noout`’

skip bitstream encoding

‘`local_header`’

place global headers at every keyframe instead of in extradata

‘`chunks`’

Frame data might be split into multiple chunks

‘`showall`’

Show all frames before the first keyframe

‘`skiprd`’

Deprecated, use mpegvideo private options instead

‘`error` *integer* (*encoding,video*)’
‘`qns` *integer* (*encoding,video*)’

Deprecated, use mpegvideo private options instead.

‘`threads` *integer* (*decoding/encoding,video*)’

Possible values:

‘`auto`’

detect a good number of threads

‘`me_threshold` *integer* (*encoding,video*)’

Set motion estimation threshold.

‘`mb_threshold` *integer* (*encoding,video*)’

Set macroblock threshold.

‘dc *integer* (*encoding,video*)’

Set intra_dc_precision.

‘nssew *integer* (*encoding,video*)’

Set nsse weight.

‘skip_top *integer* (*decoding,video*)’

Set number of macroblock rows at the top which are skipped.

‘skip_bottom *integer* (*decoding,video*)’

Set number of macroblock rows at the bottom which are skipped.

‘profile *integer* (*encoding,audio,video*)’

Possible values:

‘unknown’

‘aac_main’

‘aac_low’

‘aac_ssr’

‘aac_ltp’

‘aac_he’

‘aac_he_v2’

‘aac_ld’

‘aac_eld’

‘dts’

‘dts_es’

‘dts_96_24’

‘dts_hd_hra’

‘dts_hd_ma’

‘level *integer* (*encoding,audio,video*)’

Possible values:

‘unknown’

‘lowres *integer* (*decoding,audio,video*)’

Decode at 1= 1/2, 2=1/4, 3=1/8 resolutions.

‘skip_threshold *integer* (*encoding,video*)’

Set frame skip threshold.

‘`skip_factor` *integer* (*encoding,video*)’

Set frame skip factor.

‘`skip_exp` *integer* (*encoding,video*)’

Set frame skip exponent.

‘`skipcmp` *integer* (*encoding,video*)’

Set frame skip compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’

‘`border_mask` *float* (*encoding,video*)’

Increase the quantizer for macroblocks close to borders.

‘`mblmin` *integer* (*encoding,video*)’

Set min macroblock lagrange factor (VBR).

‘`mblmax` *integer* (*encoding,video*)’

Set max macroblock lagrange factor (VBR).

‘`mepc` *integer* (*encoding,video*)’

Set motion estimation bitrate penalty compensation (1.0 = 256).

‘`skip_loop_filter` *integer* (*decoding,video*)’
‘`skip_idct` *integer* (*decoding,video*)’
‘`skip_frame` *integer* (*decoding,video*)’

Make decoder discard processing depending on the frame type selected by the option value.

‘`skip_loop_filter`’ skips frame loop filtering, ‘`skip_idct`’ skips frame IDCT/dequantization, ‘`skip_frame`’ skips decoding.

Possible values:

‘`none`’

Discard no frame.

‘`default`’

Discard useless frames like 0-sized frames.

‘`noref`’

Discard all non-reference frames.

‘`bidir`’

Discard all bidirectional frames.

‘`nokey`’

Discard all frames excepts keyframes.

‘`all`’

Discard all frames.

Default value is ‘`default`’.

‘`bidir_refine` *`integer`* (*`encoding,video`*)’

Refine the two motion vectors used in bidirectional macroblocks.

‘`brd_scale` *`integer`* (*`encoding,video`*)’

Downscale frames for dynamic B-frame decision.

‘`keyint_min` *`integer`* (*`encoding,video`*)’

Set minimum interval between IDR-frames.

‘`refs` *`integer`* (*`encoding,video`*)’

Set reference frames to consider for motion compensation.

‘`chromaoffset` *`integer`* (*`encoding,video`*)’

Set chroma qp offset from luma.

‘`trellis` *`integer`* (*`encoding,audio,video`*)’

Set rate-distortion optimal quantization.

‘`sc_factor` *`integer`* (*`encoding,video`*)’

Set value multiplied by qscale for each frame and added to scene_change_score.

‘`mv0_threshold` *integer* (*encoding,video*)’
‘`b_sensitivity` *integer* (*encoding,video*)’

Adjust sensitivity of b_frame_strategy 1.

‘`compression_level` *integer* (*encoding,audio,video*)’
‘`min_prediction_order` *integer* (*encoding,audio*)’
‘`max_prediction_order` *integer* (*encoding,audio*)’
‘`timecode_frame_start` *integer* (*encoding,video*)’

Set GOP timecode frame start number, in non drop frame format.

‘`request_channels` *integer* (*decoding,audio*)’

Set desired number of audio channels.

‘`bits_per_raw_sample` *integer*’
‘`channel_layout` *integer* (*decoding/encoding,audio*)’

Possible values:

‘`request_channel_layout` *integer* (*decoding,audio*)’

Possible values:

‘`rc_max_vbv_use` *float* (*encoding,video*)’
‘`rc_min_vbv_use` *float* (*encoding,video*)’
‘`ticks_per_frame` *integer* (*decoding/encoding,audio,video*)’
‘`color_primaries` *integer* (*decoding/encoding,video*)’
‘`color_trc` *integer* (*decoding/encoding,video*)’
‘`colorspace` *integer* (*decoding/encoding,video*)’
‘`color_range` *integer* (*decoding/encoding,video*)’
‘`chroma_sample_location` *integer* (*decoding/encoding,video*)’
‘`log_level_offset` *integer*’

Set the log level offset.

‘`slices` *integer* (*encoding,video*)’

Number of slices, used in parallelized encoding.

‘`thread_type` *flags* (*decoding/encoding,video*)’

Select multithreading type.

Possible values:

‘slice’
‘frame’

‘audio_service_type *integer* (*encoding,audio*)’

Set audio service type.

Possible values:

‘ma’

Main Audio Service

‘ef’

Effects

‘vi’

Visually Impaired

‘hi’

Hearing Impaired

‘di’

Dialogue

‘co’

Commentary

‘em’

Emergency

‘vo’

Voice Over

‘ka’

Karaoke

‘request_sample_fmt *sample_fmt* (*decoding,audio*)’

Set sample format audio decoders should prefer. Default value is `none`.

‘`pkt_timebase` *`rational number`*’
‘`sub_charenc` *`encoding`* `(`*`decoding,subtitles`*`)`’

Set the input subtitles character encoding.

3. Decoders

Decoders are configured elements in FFmpeg which allow the decoding of multimedia streams.

When you configure your FFmpeg build, all the supported native decoders are enabled by default. Decoders requiring an external library must be enabled manually via the corresponding `--enable-lib` option. You can list all available decoders using the configure option `--list-decoders`.

You can disable all the decoders with the configure option `--disable-decoders` and selectively enable / disable single decoders with the options `--enable-decoder=`*`DECODER`* / `--disable-decoder=`*`DECODER`*.

The option `-codecs` of the ff* tools will display the list of enabled decoders.

4. Video Decoders

A description of some of the currently available video decoders follows.

4.1 rawvideo

Raw video decoder.

This decoder decodes rawvideo streams.

4.1.1 Options

‘`top` *`top_field_first`*’

Specify the assumed field type of the input video.

‘`-1`’

the video is assumed to be progressive (default)

‘`0`’

bottom-field-first is assumed

‘`1`’

top-field-first is assumed

5. Audio Decoders

5.1 ffwavesynth

Internal wave synthetizer.

This decoder generates wave patterns according to predefined sequences. Its use is purely internal and the format of the data it accepts is not publicly documented.

5.2 libcelt

libcelt decoder wrapper.

libcelt allows libavcodec to decode the Xiph CELT ultra-low delay audio codec. Requires the presence of the libcelt headers and library during configuration. You need to explicitly configure the build with `--enable-libcelt`.

5.3 libgsm

libgsm decoder wrapper.

libgsm allows libavcodec to decode the GSM full rate audio codec. Requires the presence of the libgsm headers and library during configuration. You need to explicitly configure the build with `--enable-libgsm`.

This decoder supports both the ordinary GSM and the Microsoft variant.

5.4 libilbc

libilbc decoder wrapper.

libilbc allows libavcodec to decode the Internet Low Bitrate Codec (iLBC) audio codec. Requires the presence of the libilbc headers and library during configuration. You need to explicitly configure the build with `--enable-libilbc`.

5.4.1 Options

The following option is supported by the libilbc wrapper.

‘`enhance`’

Enable the enhancement of the decoded audio when set to 1. The default value is 0 (disabled).

5.5 libopencore-amrnb

libopencore-amrnb decoder wrapper.

libopencore-amrnb allows libavcodec to decode the Adaptive Multi-Rate Narrowband audio codec. Using it requires the presence of the libopencore-amrnb headers and library during configuration. You need to explicitly configure the build with `--enable-libopencore-amrnb`.

An FFmpeg native decoder for AMR-NB exists, so users can decode AMR-NB without this library.

5.6 libopencore-amrwb

libopencore-amrwb decoder wrapper.

libopencore-amrwb allows libavcodec to decode the Adaptive Multi-Rate Wideband audio codec. Using it requires the presence of the libopencore-amrwb headers and library during configuration. You need to explicitly configure the build with `--enable-libopencore-amrwb`.

An FFmpeg native decoder for AMR-WB exists, so users can decode AMR-WB without this library.

5.7 libopus

libopus decoder wrapper.

libopus allows libavcodec to decode the Opus Interactive Audio Codec. Requires the presence of the libopus headers and library during configuration. You need to explicitly configure the build with `--enable-libopus`.

6. Subtitles Decoders

6.1 dvdsub

This codec decodes the bitmap subtitles used in DVDs; the same subtitles can also be found in VobSub file pairs and in some Matroska files.

6.1.1 Options

‘`palette`’

Specify the global palette used by the bitmaps. When stored in VobSub, the palette is normally specified in the index file; in Matroska, the palette is stored in the codec extra-data in the same format as in VobSub. In DVDs, the palette is stored in the IFO file, and therefore not available when reading from dumped VOB files.

The format for this option is a string containing 16 24-bits hexadecimal numbers (without 0x prefix) separated by comas, for example `0d00ee, ee450d, 101010, eaeaea, 0ce60b, ec14ed, ebff0b, 0d617a, 7b7b7b, d1d1d1, 7b2a0e, 0d950c, 0f007b, cf0dec, cfa80c, 7c127b`.

7. Encoders

Encoders are configured elements in FFmpeg which allow the encoding of multimedia streams.

When you configure your FFmpeg build, all the supported native encoders are enabled by default. Encoders requiring an external library must be enabled manually via the corresponding `--enable-lib` option. You can list all available encoders using the configure option `--list-encoders`.

You can disable all the encoders with the configure option `--disable-encoders` and selectively enable / disable single encoders with the options `--enable-encoder=`*`ENCODER`* / `--disable-encoder=`*`ENCODER`*.

The option `-codecs` of the ff* tools will display the list of enabled encoders.

8. Audio Encoders

A description of some of the currently available audio encoders follows.

8.1 ac3 and ac3_fixed

AC-3 audio encoders.

These encoders implement part of ATSC A/52:2010 and ETSI TS 102 366, as well as the undocumented RealAudio 3 (a.k.a. dnet).

The *ac3* encoder uses floating-point math, while the *ac3_fixed* encoder only uses fixed-point integer math. This does not mean that one is always faster, just that one or the other may be better suited to a particular system. The floating-point encoder will generally produce better quality audio for a given bitrate. The *ac3_fixed* encoder is not the default codec for any of the output formats, so it must be specified explicitly using the option `-acodec ac3_fixed` in order to use it.

8.1.1 AC-3 Metadata

The AC-3 metadata options are used to set parameters that describe the audio, but in most cases do not affect the audio encoding itself. Some of the options do directly affect or influence the decoding and playback of the resulting bitstream, while others are just for informational purposes. A few of the options will add bits to the output stream that could otherwise be used for audio data, and will thus affect the quality of the output. Those will be indicated accordingly with a note in the option list below.

These parameters are described in detail in several publicly-available documents.

- A/52:2010 - Digital Audio Compression (AC-3) (E-AC-3) Standard
- A/54 - Guide to the Use of the ATSC Digital Television Standard
- Dolby Metadata Guide
- Dolby Digital Professional Encoding Guidelines

8.1.1.1 Metadata Control Options

‘`-per_frame_metadata` *`boolean`*’

Allow Per-Frame Metadata. Specifies if the encoder should check for changing metadata for each frame.

‘`0`’

The metadata values set at initialization will be used for every frame in the stream. (default)

‘`1`’

Metadata values can be changed before encoding each frame.

8.1.1.2 Downmix Levels

‘`-center_mixlev` *`level`*’

Center Mix Level. The amount of gain the decoder should apply to the center channel when downmixing to stereo. This field will only be written to the bitstream if a center channel is present. The value is specified as a scale factor. There are 3 valid values:

‘`0.707`’

Apply -3dB gain

‘`0.595`’

Apply -4.5dB gain (default)

‘`0.500`’

Apply -6dB gain

‘`-surround_mixlev` *`level`*’

Surround Mix Level. The amount of gain the decoder should apply to the surround channel(s) when downmixing to stereo. This field will only be written to the bitstream if one or more surround channels are present. The value is specified as a scale factor. There are 3 valid values:

‘`0.707`’

Apply -3dB gain

‘`0.500`’

Apply -6dB gain (default)

‘`0.000`’

Silence Surround Channel(s)

8.1.1.3 Audio Production Information

Audio Production Information is optional information describing the mixing environment. Either none or both of the fields are written to the bitstream.

‘`-mixing_level` *`number`*’

Mixing Level. Specifies peak sound pressure level (SPL) in the production environment when the mix was mastered. Valid values are 80 to 111, or -1 for unknown or not indicated. The default value is -1, but that value cannot be used if the Audio Production Information is written to the bitstream. Therefore, if the `room_type` option is not the default value, the `mixing_level` option must not be -1.

‘`-room_type` *`type`*’

Room Type. Describes the equalization used during the final mixing session at the studio or on the dubbing stage. A large room is a dubbing stage with the industry standard X-curve equalization; a small room has flat equalization. This field will not be written to the bitstream if both the `mixing_level` option and the `room_type` option have the default values.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`large`’

Large Room

'`2`'
'`small`'

Small Room

8.1.1.4 Other Metadata Options

'`-copyright` *`boolean`*'

Copyright Indicator. Specifies whether a copyright exists for this audio.

'`0`'
'`off`'

No Copyright Exists (default)

'`1`'
'`on`'

Copyright Exists

'`-dialnorm` *`value`*'

Dialogue Normalization. Indicates how far the average dialogue level of the program is below digital 100% full scale (0 dBFS). This parameter determines a level shift during audio reproduction that sets the average volume of the dialogue to a preset level. The goal is to match volume level between program sources. A value of -31dB will result in no volume level change, relative to the source volume, during audio reproduction. Valid values are whole numbers in the range -31 to -1, with -31 being the default.

'`-dsur_mode` *`mode`*'

Dolby Surround Mode. Specifies whether the stereo signal uses Dolby Surround (Pro Logic). This field will only be written to the bitstream if the audio stream is stereo. Using this option does **NOT** mean the encoder will actually apply Dolby Surround processing.

'`0`'
'`notindicated`'

Not Indicated (default)

'`1`'
'`off`'

Not Dolby Surround Encoded

‘`2`’
‘`on`’

Dolby Surround Encoded

‘`-original` *`boolean`*’

Original Bit Stream Indicator. Specifies whether this audio is from the original source and not a copy.

‘`0`’
‘`off`’

Not Original Source

‘`1`’
‘`on`’

Original Source (default)

8.1.2 Extended Bitstream Information

The extended bitstream options are part of the Alternate Bit Stream Syntax as specified in Annex D of the A/52:2010 standard. It is grouped into 2 parts. If any one parameter in a group is specified, all values in that group will be written to the bitstream. Default values are used for those that are written but have not been specified. If the mixing levels are written, the decoder will use these values instead of the ones specified in the `center_mixlev` and `surround_mixlev` options if it supports the Alternate Bit Stream Syntax.

8.1.2.1 Extended Bitstream Information - Part 1

‘`-dmix_mode` *`mode`*’

Preferred Stereo Downmix Mode. Allows the user to select either Lt/Rt (Dolby Surround) or Lo/Ro (normal stereo) as the preferred stereo downmix mode.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`ltrt`’

Lt/Rt Downmix Preferred

‘`2`’
‘`loro`’

Lo/Ro Downmix Preferred

‘`-ltrt_cmixlev` *`level`*’

Lt/Rt Center Mix Level. The amount of gain the decoder should apply to the center channel when downmixing to stereo in Lt/Rt mode.

‘`1.414`’

Apply +3dB gain

‘`1.189`’

Apply +1.5dB gain

‘`1.000`’

Apply 0dB gain

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain (default)

‘`0.500`’

Apply -6.0dB gain

‘`0.000`’

Silence Center Channel

‘`-ltrt_surmixlev` *`level`*’

Lt/Rt Surround Mix Level. The amount of gain the decoder should apply to the surround channel(s) when downmixing to stereo in Lt/Rt mode.

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain

‘`0.500`’

Apply -6.0dB gain (default)

‘`0.000`’

Silence Surround Channel(s)

‘`-loro_cmixlev` *`level`*’

Lo/Ro Center Mix Level. The amount of gain the decoder should apply to the center channel when downmixing to stereo in Lo/Ro mode.

‘`1.414`’

Apply +3dB gain

‘`1.189`’

Apply +1.5dB gain

‘`1.000`’

Apply 0dB gain

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain (default)

‘`0.500`’

Apply -6.0dB gain

‘`0.000`’

Silence Center Channel

‘`-loro_surmixlev *level*`’

Lo/Ro Surround Mix Level. The amount of gain the decoder should apply to the surround channel(s) when downmixing to stereo in Lo/Ro mode.

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain

‘`0.500`’

Apply -6.0dB gain (default)

‘`0.000`’

Silence Surround Channel(s)

8.1.2.2 Extended Bitstream Information - Part 2

‘`-dsurex_mode *mode*`’

Dolby Surround EX Mode. Indicates whether the stream uses Dolby Surround EX (7.1 matrixed to 5.1). Using this option does **NOT** mean the encoder will actually apply Dolby Surround EX processing.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`on`’

Dolby Surround EX Off

‘`2`’

‘`off`’

Dolby Surround EX On

‘`-dheadphone_mode` *`mode`*’

Dolby Headphone Mode. Indicates whether the stream uses Dolby Headphone encoding (multi-channel matrixed to 2.0 for use with headphones). Using this option does **NOT** mean the encoder will actually apply Dolby Headphone processing.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`on`’

Dolby Headphone Off

‘`2`’
‘`off`’

Dolby Headphone On

‘`-ad_conv_type` *`type`*’

A/D Converter Type. Indicates whether the audio has passed through HDCD A/D conversion.

‘`0`’
‘`standard`’

Standard A/D Converter (default)

‘`1`’
‘`hdcd`’

HDCD A/D Converter

8.1.3 Other AC-3 Encoding Options

‘`-stereo_rematrixing` *`boolean`*’

Stereo Rematrixing. Enables/Disables use of rematrixing for stereo input. This is an optional AC-3 feature that increases quality by selectively encoding the left/right channels as mid/side. This option is enabled by default, and it is highly recommended that it be left as enabled except for testing purposes.

8.1.4 Floating-Point-Only AC-3 Encoding Options

These options are only valid for the floating-point encoder and do not exist for the fixed-point encoder due to the corresponding features not being implemented in fixed-point.

‘`-channel_coupling` *`boolean`*’

Enables/Disables use of channel coupling, which is an optional AC-3 feature that increases quality by combining high frequency information from multiple channels into a single channel. The per-channel high frequency information is sent with less accuracy in both the frequency and time domains. This allows more bits to be used for lower frequencies while preserving enough information to reconstruct the high frequencies. This option is enabled by default for the floating-point encoder and should generally be left as enabled except for testing purposes or to increase encoding speed.

‘`-1`’
‘`auto`’

Selected by Encoder (default)

‘`0`’
‘`off`’

Disable Channel Coupling

‘`1`’
‘`on`’

Enable Channel Coupling

‘`-cpl_start_band` *`number`*’

Coupling Start Band. Sets the channel coupling start band, from 1 to 15. If a value higher than the bandwidth is used, it will be reduced to 1 less than the coupling end band. If *auto* is used, the start band will be determined by the encoder based on the bit rate, sample rate, and channel layout. This option has no effect if channel coupling is disabled.

‘`-1`’
‘`auto`’

Selected by Encoder (default)

8.2 libmp3lame

LAME (Lame Ain’t an MP3 Encoder) MP3 encoder wrapper.

Requires the presence of the libmp3lame headers and library during configuration. You need to explicitly configure the build with `--enable-libmp3lame`.

8.2.1 Option Mapping

The following options are supported by the libmp3lame wrapper, the LAME-equivalent options follow the FFmpeg ones.

FFmpeg	LAME
b	b Set bitrate expressed in bits/s, LAME `bitrate` is expressed in kilobits/s.
q	V Set quality setting for VBR.
compression_level	q Set algorithm quality. Valid arguments are integers in the 0-9 range.
reservoir	N.A. Enable use of bit reservoir. LAME has this enabled by default.
joint_stereo	-m j Enable the encoder to use (on a frame by frame basis) either L/R stereo or mid/side stereo.

8.3 libopencore-amrnb

OpenCORE Adaptive Multi-Rate Narrowband encoder.

Requires the presence of the libopencore-amrnb headers and library during configuration. You need to explicitly configure the build with `--enable-libopencore-amrnb --enable-version3`.

This is a mono-only encoder. Officially it only supports 8000Hz sample rate, but you can override it by setting ‘`strict`’ to ‘`unofficial`’ or lower.

8.3.1 Options

‘`b`’

Set bitrate in bits per second. Only the following bitrates are supported, otherwise libavcodec will round to the nearest valid bitrate.

‘`4750`’
‘`5150`’
‘`5900`’
‘`6700`’
‘`7400`’
‘`7950`’
‘`10200`’
‘`12200`’

‘`dtx`’

Allow discontinuous transmission (generate comfort noise) when set to 1. The default value is 0 (disabled).

8.4 libtwolame

TwoLAME MP2 encoder wrapper.

Requires the presence of the libtwolame headers and library during configuration. You need to explicitly configure the build with `--enable-libtwolame`.

8.4.1 Options Mapping

The following options are supported by the libtwolame wrapper. The TwoLAME-equivalent options follow the FFmpeg ones and are in parentheses.

‘`b`’

(b) Set bitrate in bits/s. Note that FFmpeg `b` option is expressed in bits/s, twolame `b` in kilobits/s. The default value is 128k.

‘`q`’

(V) Set quality for experimental VBR support. Maximum value range is from -50 to 50, useful range is from -10 to 10.

‘`mode`’

(mode) Set MPEG mode. Possible values:

‘`auto`’

Choose mode automatically based on the input. This is the default.

‘`stereo`’

Stereo

‘`joint_stereo`’

Joint stereo

‘`dual_channel`’

Dual channel

‘`mono`’

Mono

‘`psymodel`’

(psyc-mode) Set psychoacoustic model to use in encoding. The argument must be an integer between -1 and 4, inclusive. The higher the value, the better the quality. The default value is 3.

‘`energy_levels`’

(energy) Enable energy levels extensions when set to 1. The default value is 0 (disabled).

‘`error_protection`’

(protect) Enable CRC error protection when set to 1. The default value is 0 (disabled).

‘`copyright`’

(copyright) Set MPEG audio copyright flag when set to 1. The default value is 0 (disabled).

‘`original`’

(original) Set MPEG audio original flag when set to 1. The default value is 0 (disabled).

8.5 libvo-aacenc

VisualOn AAC encoder.

Requires the presence of the libvo-aacenc headers and library during configuration. You need to explicitly configure the build with `--enable-libvo-aacenc --enable-version3`.

8.5.1 Options

The VisualOn AAC encoder only support encoding AAC-LC and up to 2 channels. It is also CBR-only. It is considered to be worse than the native experimental FFmpeg AAC encoder.

‘`b`’

Bitrate.

8.6 libvo-amrwbenc

VisualOn Adaptive Multi-Rate Wideband encoder.

Requires the presence of the libvo-amrwbenc headers and library during configuration. You need to explicitly configure the build with `--enable-libvo-amrwbenc --enable-version3`.

This is a mono-only encoder. Officially it only supports 16000Hz sample rate, but you can override it by setting ‘`strict`’ to ‘`unofficial`’ or lower.

8.6.1 Options

‘`b`’

Set bitrate in bits/s. Only the following bitrates are supported, otherwise libavcodec will round to the nearest valid bitrate.

‘`6600`’
‘`8850`’
‘`12650`’
‘`14250`’
‘`15850`’
‘`18250`’
‘`19850`’
‘`23050`’
‘`23850`’

‘`dtx`’

Allow discontinuous transmission (generate comfort noise) when set to 1. The default value is 0 (disabled).

8.7 libopus

libopus Opus Interactive Audio Codec encoder wrapper.

Requires the presence of the libopus headers and library during configuration. You need to explicitly configure the build with `--enable-libopus`.

8.7.1 Option Mapping

Most libopus options are modeled after the `opusenc` utility from opus-tools. The following is an option mapping chart describing options supported by the libopus wrapper, and their `opusenc`-equivalent in parentheses.

‘`b (`*`bitrate`*`)`’

Set the bit rate in bits/s. FFmpeg’s ‘`b`’ option is expressed in bits/s, while `opusenc`’s ‘`bitrate`’ in kilobits/s.

‘`vbr (`*`vbr`*`, `*`hard-cbr`*`, and `*`cvbr`*`)`’

Set VBR mode. The FFmpeg ‘`vbr`’ option has the following valid arguments, with the their `opusenc` equivalent options in parentheses:

‘`off (`*`hard-cbr`*`)`’

Use constant bit rate encoding.

‘`on (`*`vbr`*`)`’

Use variable bit rate encoding (the default).

‘`constrained (`*`cvbr`*`)`’

Use constrained variable bit rate encoding.

‘`compression_level (`*`comp`*`)`’

Set encoding algorithm complexity. Valid options are integers in the 0-10 range. 0 gives the fastest encodes but lower quality, while 10 gives the highest quality but slowest encoding. The default is 10.

‘`frame_duration (`*`framesize`*`)`’

Set maximum frame size, or duration of a frame in milliseconds. The argument must be exactly the following: 2.5, 5, 10, 20, 40, 60. Smaller frame sizes achieve lower latency but less quality at a given bitrate. Sizes greater than 20ms are only interesting at fairly low bitrates. The default of FFmpeg is 10ms, but is 20ms in `opusenc`.

‘`packet_loss (`*`expect-loss`*`)`’

Set expected packet loss percentage. The default is 0.

‘`application (N.A.)`’

Set intended application type. Valid options are listed below:

‘`voip`’

Favor improved speech intelligibility.

‘`audio`’

Favor faithfulness to the input (the default).

‘`lowdelay`’

Restrict to only the lowest delay modes.

‘`cutoff (N.A.)`’

Set cutoff bandwidth in Hz. The argument must be exactly one of the following: 4000, 6000, 8000, 12000, or 20000, corresponding to narrowband, mediumband, wideband, super wideband, and fullband respectively. The default is 0 (cutoff disabled).

8.8 libwavpack

A wrapper providing WavPack encoding through libwavpack.

Only lossless mode using 32-bit integer samples is supported currently. The ‘`compression_level`’ option can be used to control speed vs. compression tradeoff, with the values mapped to libwavpack as follows:

‘`0`’

Fast mode - corresponding to the wavpack ‘`-f`’ option.

‘`1`’

Normal (default) settings.

‘`2`’

High quality - corresponding to the wavpack ‘`-h`’ option.

‘`3`’

Very high quality - corresponding to the wavpack ‘`-hh`’ option.

‘`4-8`’

Same as 3, but with extra processing enabled - corresponding to the wavpack ‘`-x`’ option. I.e. 4 is the same as ‘`-x2`’ and 8 is the same as ‘`-x6`’.

9. Video Encoders

A description of some of the currently available video encoders follows.

9.1 libtheora

Theora format supported through libtheora.

Requires the presence of the libtheora headers and library during configuration. You need to explicitly configure the build with `--enable-libtheora`.

9.1.1 Options

The following global options are mapped to internal libtheora options which affect the quality and the bitrate of the encoded stream.

‘b’

Set the video bitrate, only works if the qscale flag in ‘flags’ is not enabled.

‘flags’

Used to enable constant quality mode encoding through the ‘qscale’ flag, and to enable the pass1 and pass2 modes.

‘g’

Set the GOP size.

‘global_quality’

Set the global quality in lambda units, only works if the qscale flag in ‘flags’ is enabled. The value is clipped in the [0 - 10*FF_QP2LAMBDA] range, and then multiplied for 6.3 to get a value in the native libtheora range [0-63]. A higher value corresponds to a higher quality.

For example, to set maximum constant quality encoding with ffmpeg:

```
ffmpeg -i INPUT -flags:v qscale -global_quality:v "10*QP2LAMBDA" -codec:v libtheora OUTPUT.ogg
```

9.2 libvpx

VP8 format supported through libvpx.

Requires the presence of the libvpx headers and library during configuration. You need to explicitly configure the build with --enable-libvpx.

9.2.1 Options

Mapping from FFmpeg to libvpx options with conversion notes in parentheses.

‘threads’

g_threads

‘profile’

g_profile

‘vb’

rc_target_bitrate

‘`g`’

kf_max_dist

‘`keyint_min`’

kf_min_dist

‘`qmin`’

rc_min_quantizer

‘`qmax`’

rc_max_quantizer

‘`bufsize, vb`’

rc_buf_sz (`bufsize * 1000 / vb`)

rc_buf_optimal_sz (`bufsize * 1000 / vb * 5 / 6`)

‘`rc_init_occupancy, vb`’

rc_buf_initial_sz (`rc_init_occupancy * 1000 / vb`)

‘`rc_buffer_aggressivity`’

rc_undershoot_pct

‘`skip_threshold`’

rc_dropframe_thresh

‘`qcomp`’

rc_2pass_vbr_bias_pct

‘`maxrate, vb`’

rc_2pass_vbr_maxsection_pct (`maxrate * 100 / vb`)

‘`minrate, vb`’

rc_2pass_vbr_minsection_pct (`minrate * 100 / vb`)

‘`minrate, maxrate, vb`’

VPX_CBR (minrate == maxrate == vb)

‘crf’

VPX_CQ, VP8E_SET_CQ_LEVEL

‘quality’

- ‘*best*’

  VPX_DL_BEST_QUALITY

- ‘*good*’

  VPX_DL_GOOD_QUALITY

- ‘*realtime*’

  VPX_DL_REALTIME

‘speed’

VP8E_SET_CPUUSED

‘nr’

VP8E_SET_NOISE_SENSITIVITY

‘mb_threshold’

VP8E_SET_STATIC_THRESHOLD

‘slices’

VP8E_SET_TOKEN_PARTITIONS

‘max-intra-rate’

VP8E_SET_MAX_INTRA_BITRATE_PCT

‘force_key_frames’

VPX_EFLAG_FORCE_KF

‘Alternate reference frame related’

- ‘vp8flags altref’

  VP8E_SET_ENABLEAUTOALTREF

‘*arnr_max_frames*’

VP8E_SET_ARNR_MAXFRAMES

‘*arnr_type*’

VP8E_SET_ARNR_TYPE

‘*arnr_strength*’

VP8E_SET_ARNR_STRENGTH

‘*rc_lookahead*’

g_lag_in_frames

‘vp8flags error_resilient’

g_error_resilient

For more information about libvpx see: http://www.webmproject.org/

9.3 libx264

x264 H.264/MPEG-4 AVC encoder wrapper.

Requires the presence of the libx264 headers and library during configuration. You need to explicitly configure the build with `--enable-libx264`.

x264 supports an impressive number of features, including 8x8 and 4x4 adaptive spatial transform, adaptive B-frame placement, CAVLC/CABAC entropy coding, interlacing (MBAFF), lossless mode, psy optimizations for detail retention (adaptive quantization, psy-RD, psy-trellis).

The FFmpeg wrapper provides a mapping for most of them using global options that match those of the encoders and provides private options for the unique encoder options. Additionally an expert override is provided to directly pass a list of key=value tuples as accepted by x264_param_parse.

9.3.1 Option Mapping

The following options are supported by the x264 wrapper, the x264-equivalent options follow the FFmpeg ones.

b	bitrate FFmpeg `b` option is expressed in bits/s, x264 `bitrate` in kilobits/s.
bf	bframes Maximum number of B-frames.
g	keyint Maximum GOP size.
qmin	qpmin
qmax	qpmax
qdiff	qpstep
qblur	qblur
qcomp	qcomp
refs	ref
sc_threshold	scenecut
trellis	trellis
nr	nr Noise reduction.
me_range	merange
me_method	me
subq	subme
b_strategy	b-adapt
keyint_min	keyint-min
coder	cabac Set coder to `ac` to use CABAC.
cmp	chroma-me Set to `chroma` to use chroma motion estimation.
threads	threads
thread_type	sliced_threads Set to `slice` to use sliced threading instead of frame threading.
flags -cgop	open-gop Set `-cgop` to use recovery points to close GOPs.
rc_init_occupancy	vbv-init Initial buffer occupancy.

9.3.2 Private Options

‘`-preset` *`string`*’

Set the encoding preset (cf. x264 –fullhelp).

‘`-tune` *`string`*’

Tune the encoding params (cf. x264 –fullhelp).

‘`-profile` *`string`*’

Set profile restrictions (cf. x264 –fullhelp).

‘`-fastfirstpass` *`integer`*’

Use fast settings when encoding first pass.

‘`-crf` *`float`*’

Select the quality for constant quality mode.

‘`-crf_max` *`float`*’

In CRF mode, prevents VBV from lowering quality beyond this point.

‘`-qp` *`integer`*’

Constant quantization parameter rate control method.

‘`-aq-mode` *`integer`*’

AQ method

Possible values:

‘`none`’
‘`variance`’

Variance AQ (complexity mask).

‘`autovariance`’

Auto-variance AQ (experimental).

‘`-aq-strength` *`float`*’

AQ strength, reduces blocking and blurring in flat and textured areas.

‘`-psy` *`integer`*’

Use psychovisual optimizations.

‘`-psy-rd` *`string`*’

Strength of psychovisual optimization, in <psy-rd>:<psy-trellis> format.

‘`-rc-lookahead` *integer*’

Number of frames to look ahead for frametype and ratecontrol.

‘`-weightb` *integer*’

Weighted prediction for B-frames.

‘`-weightp` *integer*’

Weighted prediction analysis method.

Possible values:

‘`none`’
‘`simple`’
‘`smart`’

‘`-ssim` *integer*’

Calculate and print SSIM stats.

‘`-intra-refresh` *integer*’

Use Periodic Intra Refresh instead of IDR frames.

‘`-b-bias` *integer*’

Influences how often B-frames are used.

‘`-b-pyramid` *integer*’

Keep some B-frames as references.

Possible values:

‘`none`’
‘`strict`’

Strictly hierarchical pyramid.

‘`normal`’

Non-strict (not Blu-ray compatible).

‘`-mixed-refs` *integer*’

One reference per partition, as opposed to one reference per macroblock.

‘-8x8dct *integer*’

High profile 8x8 transform.

‘-fast-pskip *integer*’
‘-aud *integer*’

Use access unit delimiters.

‘-mbtree *integer*’

Use macroblock tree ratecontrol.

‘-deblock *string*’

Loop filter parameters, in <alpha:beta> form.

‘-cplxblur *float*’

Reduce fluctuations in QP (before curve compression).

‘-partitions *string*’

A comma-separated list of partitions to consider, possible values: p8x8, p4x4, b8x8, i8x8, i4x4, none, all.

‘-direct-pred *integer*’

Direct MV prediction mode

Possible values:

‘none’
‘spatial’
‘temporal’
‘auto’

‘-slice-max-size *integer*’

Limit the size of each slice in bytes.

‘-stats *string*’

Filename for 2 pass stats.

‘-nal-hrd *integer*’

Signal HRD information (requires vbv-bufsize; cbr not allowed in .mp4).

Possible values:

‘`none`’
‘`vbr`’
‘`cbr`’

‘`x264opts` *`options`*’

Allow to set any x264 option, see `x264 --fullhelp` for a list.

options is a list of *key*=*value* couples separated by ":". In *filter* and *psy-rd* options that use ":" as a separator themselves, use "," instead. They accept it as well since long ago but this is kept undocumented for some reason.

For example to specify libx264 encoding options with `ffmpeg`:

```
ffmpeg -i foo.mpg -vcodec libx264 -x264opts keyint=123:min-keyint=20 -an out.mkv
```

For more information about libx264 and the supported options see:
http://www.videolan.org/developers/x264.html

‘`-x264-params` *`string`*’

Override the x264 configuration using a :-separated list of key=value parameters.

```
-x264-params level=30:bframes=0:weightp=0:cabac=0:ref=1:vbv-maxrate=768:vbv-bufsize=2000:analyse=all:me=umh:no-fast-pskip=1:subq=6:8x8dct=0:trellis=0
```

Encoding avpresets for common usages are provided so they can be used with the general presets system (e.g. passing the `-pre` option).

9.4 png

PNG image encoder.

9.4.1 Private options

‘`dpi` *`integer`*’

Set physical density of pixels, in dots per inch, unset by default

‘`dpm` *`integer`*’

Set physical density of pixels, in dots per meter, unset by default

9.5 ProRes

Apple ProRes encoder.

FFmpeg contains 2 ProRes encoders, the prores-aw and prores-ks encoder. The used encoder can be choosen with the `-vcodec` option.

9.5.1 Private Options for prores-ks

‘`profile` *`integer`*’

Select the ProRes profile to encode

‘`proxy`’
‘`lt`’
‘`standard`’
‘`hq`’
‘`4444`’

‘`quant_mat` *`integer`*’

Select quantization matrix.

‘`auto`’
‘`default`’
‘`proxy`’
‘`lt`’
‘`standard`’
‘`hq`’

If set to *auto*, the matrix matching the profile will be picked. If not set, the matrix providing the highest quality, *default*, will be picked.

‘`bits_per_mb` *`integer`*’

How many bits to allot for coding one macroblock. Different profiles use between 200 and 2400 bits per macroblock, the maximum is 8000.

‘`mbs_per_slice` *`integer`*’

Number of macroblocks in each slice (1-8); the default value (8) should be good in almost all situations.

‘`vendor` *`string`*’

Override the 4-byte vendor ID. A custom vendor ID like *apl0* would claim the stream was produced by the Apple encoder.

‘`alpha_bits` *`integer`*’

Specify number of bits for alpha component. Possible values are *0*, *8* and *16*. Use *0* to disable alpha plane coding.

9.5.2 Speed considerations

In the default mode of operation the encoder has to honor frame constraints (i.e. not produc frames with size bigger than requested) while still making output picture as good as possible. A frame containing a lot of small details is harder to compress and the encoder would spend more time searching for appropriate quantizers for each slice.

Setting a higher ‘`bits_per_mb`’ limit will improve the speed.

For the fastest encoding speed set the ‘`qscale`’ parameter (4 is the recommended value) and do not set a size constraint.

10. See Also

ffmpeg, ffplay, ffprobe, ffserver, libavcodec

11. Authors

The FFmpeg developers.

For details about the authorship, see the Git history of the project (git://source.ffmpeg.org/ffmpeg), e.g. by typing the command `git log` in the FFmpeg source directory, or browsing the online repository at http://source.ffmpeg.org.

Maintainers for the specific components are listed in the file ‘`MAINTAINERS`’ in the source code tree.

This document was generated by *john* on *September 24, 2013* using *texi2html 1.82*.

FFmpeg Codecs Documentation

Table of Contents

1. Description

This document describes the codecs (decoders and encoders) provided by the libavcodec library.

2. Codec Options

libavcodec provides some generic global options, which can be set on all the encoders and decoders. In addition each codec may support so-called private options, which are specific for a given codec.

Sometimes, a global option may only affect a specific kind of codec, and may be unsensical or ignored by another, so you need to be aware of the meaning of the specified options. Also some options are meant only for decoding or encoding.

Options may be set by specifying -*option value* in the FFmpeg tools, or by setting the value explicitly in the `AVCodecContext` options or using the ‘`libavutil/opt.h`’ API for programmatic use.

The list of supported options follow:

‘`b` *`integer`* `(`*`encoding,audio,video`*`)`’

Set bitrate in bits/s. Default value is 200K.

‘`ab` *`integer`* (*`encoding,audio`*)’

Set audio bitrate (in bits/s). Default value is 128K.

‘`bt` *`integer`* (*`encoding,video`*)’

Set video bitrate tolerance (in bits/s). In 1-pass mode, bitrate tolerance specifies how far ratecontrol is willing to deviate from the target average bitrate value. This is not related to min/max bitrate. Lowering tolerance too much has an adverse effect on quality.

‘`flags` *`flags`* (*`decoding/encoding,audio,video,subtitles`*)’

Set generic flags.

Possible values:

‘`mv4`’

Use four motion vector by macroblock (mpeg4).

‘`qpel`’

Use 1/4 pel motion compensation.

‘`loop`’

Use loop filter.

‘`qscale`’

Use fixed qscale.

‘`gmc`’

Use gmc.

‘`mv0`’

Always try a mb with mv=<0,0>.

‘`input_preserved`’
‘`pass1`’

Use internal 2pass ratecontrol in first pass mode.

‘`pass2`’

Use internal 2pass ratecontrol in second pass mode.

‘`gray`’

Only decode/encode grayscale.

‘`emu_edge`’

Do not draw edges.

‘`psnr`’

Set error[?] variables during encoding.

‘`truncated`’
‘`naq`’

Normalize adaptive quantization.

‘`ildct`’

Use interlaced DCT.

‘`low_delay`’

Force low delay.

‘`global_header`’

Place global headers in extradata instead of every keyframe.

‘`bitexact`’

Use only bitexact stuff (except (I)DCT).

‘`aic`’

Apply H263 advanced intra coding / mpeg4 ac prediction.

‘`cbp`’

Deprecated, use mpegvideo private options instead.

‘`qprd`’

Deprecated, use mpegvideo private options instead.

‘`ilme`’

Apply interlaced motion estimation.

‘`cgop`’

Use closed gop.

‘`sub_id` *`integer`*’

Deprecated, currently unused.

‘`me_method` *`integer`* `(`*`encoding,video`*`)`’

Set motion estimation method.

Possible values:

‘`zero`’

zero motion estimation (fastest)

‘`full`’

full motion estimation (slowest)

‘`epzs`’

EPZS motion estimation (default)

‘`esa`’

esa motion estimation (alias for full)

‘`tesa`’

tesa motion estimation

‘`dia`’

dia motion estimation (alias for epzs)

‘`log`’

log motion estimation

‘`phods`’

phods motion estimation

‘`x1`’

X1 motion estimation

‘`hex`’

hex motion estimation

‘`umh`’

umh motion estimation

‘`iter`’

iter motion estimation

‘`extradata_size` *`integer`*’

Set extradata size.

‘`time_base` *`rational number`*’

Set codec time base.

It is the fundamental unit of time (in seconds) in terms of which frame timestamps are represented. For fixed-fps content, timebase should be `1 / frame_rate` and timestamp increments should be identically 1.

‘`g` *`integer`* (*`encoding,video`*)’

Set the group of picture size. Default value is 12.

‘`ar` *`integer`* (*`decoding/encoding,audio`*)’

Set audio sampling rate (in Hz).

‘`ac` *`integer`* (*`decoding/encoding,audio`*)’

Set number of audio channels.

‘`cutoff` *`integer`* (*`encoding,audio`*)’

Set cutoff bandwidth.

‘`frame_size` *`integer`* (*`encoding,audio`*)’

Set audio frame size.

Each submitted frame except the last must contain exactly frame_size samples per channel. May be 0 when the codec has CODEC_CAP_VARIABLE_FRAME_SIZE set, in that case the frame size is not restricted. It is set by some decoders to indicate constant frame size.

‘`frame_number` *`integer`*’

Set the frame number.

‘`delay` *`integer`*’
‘`qcomp` *`float`* `(`*`encoding,video`*`)`’

Set video quantizer scale compression (VBR). It is used as a constant in the ratecontrol equation. Recommended range for default rc_eq: 0.0-1.0.

‘`qblur` *`float`* `(`*`encoding,video`*`)`’

Set video quantizer scale blur (VBR).

‘`qmin` *`integer`* `(`*`encoding,video`*`)`’

Set min video quantizer scale (VBR). Must be included between -1 and 69, default value is 2.

‘`qmax` *`integer`* `(`*`encoding,video`*`)`’

Set max video quantizer scale (VBR). Must be included between -1 and 1024, default value is 31.

‘`qdiff` *`integer`* `(`*`encoding,video`*`)`’

Set max difference between the quantizer scale (VBR).

‘`bf` *`integer`* `(`*`encoding,video`*`)`’

Set max number of B frames.

‘`b_qfactor` *`float`* `(`*`encoding,video`*`)`’

Set qp factor between P and B frames.

‘`rc_strategy` *`integer`* `(`*`encoding,video`*`)`’

Set ratecontrol method.

‘`b_strategy` *`integer`* `(`*`encoding,video`*`)`’

Set strategy to choose between I/P/B-frames.

‘`ps` *`integer`* `(`*`encoding,video`*`)`’

Set RTP payload size in bytes.

‘`mv_bits` *`integer`*’
‘`header_bits` *`integer`*’

‘i_tex_bits *integer*’
‘p_tex_bits *integer*’
‘i_count *integer*’
‘p_count *integer*’
‘skip_count *integer*’
‘misc_bits *integer*’
‘frame_bits *integer*’
‘codec_tag *integer*’
‘bug *flags* (*decoding,video*)’

Workaround not auto detected encoder bugs.

Possible values:

‘autodetect’
‘old_msmpeg4’

some old lavc generated msmpeg4v3 files (no autodetection)

‘xvid_ilace’

Xvid interlacing bug (autodetected if fourcc==XVIX)

‘ump4’

(autodetected if fourcc==UMP4)

‘no_padding’

padding bug (autodetected)

‘amv’
‘ac_vlc’

illegal vlc bug (autodetected per fourcc)

‘qpel_chroma’
‘std_qpel’

old standard qpel (autodetected per fourcc/version)

‘qpel_chroma2’
‘direct_blocksize’

direct-qpel-blocksize bug (autodetected per fourcc/version)

‘edge’

edge padding bug (autodetected per fourcc/version)

‘`hpel_chroma`’
‘`dc_clip`’
‘`ms`’

Workaround various bugs in microsoft broken decoders.

‘`trunc`’

trancated frames

‘`lelim` *`integer`* `(`*`encoding,video`*`)`’

Set single coefficient elimination threshold for luminance (negative values also consider DC coefficient).

‘`celim` *`integer`* `(`*`encoding,video`*`)`’

Set single coefficient elimination threshold for chrominance (negative values also consider dc coefficient)

‘`strict` *`integer`* `(`*`decoding/encoding,audio,video`*`)`’

Specify how strictly to follow the standards.

Possible values:

‘`very`’

strictly conform to a older more strict version of the spec or reference software

‘`strict`’

strictly conform to all the things in the spec no matter what consequences

‘`normal`’
‘`unofficial`’

allow unofficial extensions

‘`experimental`’

allow non standardized experimental things, experimental (unfinished/work in progress/not well tested) decoders and encoders. Note: experimental decoders can pose a security risk, do not use this for decoding untrusted input.

‘`b_qoffset` *`float`* `(`*`encoding,video`*`)`’

Set QP offset between P and B frames.

‘`err_detect` *`flags`* (*`decoding,audio,video`*)’

Set error detection flags.

Possible values:

‘`crccheck`’

verify embedded CRCs

‘`bitstream`’

detect bitstream specification deviations

‘`buffer`’

detect improper bitstream length

‘`explode`’

abort decoding on minor error detection

‘`careful`’

consider things that violate the spec and have not been seen in the wild as errors

‘`compliant`’

consider all spec non compliancies as errors

‘`aggressive`’

consider things that a sane encoder should not do as an error

‘`has_b_frames` *`integer`*’
‘`block_align` *`integer`*’
‘`mpeg_quant` *`integer`* (*`encoding,video`*)’

Use MPEG quantizers instead of H.263.

‘`qsquish` *`float`* (*`encoding,video`*)’

How to keep quantizer between qmin and qmax (0 = clip, 1 = use differentiable function).

‘`rc_qmod_amp` *`float`* (*`encoding,video`*)’

Set experimental quantizer modulation.

‘`rc_qmod_freq` *integer* (*encoding,video*)’

Set experimental quantizer modulation.

‘`rc_override_count` *integer*’
‘`rc_eq` *string* (*encoding,video*)’

Set rate control equation. When computing the expression, besides the standard functions defined in the section ’Expression Evaluation’, the following functions are available: bits2qp(bits), qp2bits(qp). Also the following constants are available: iTex pTex tex mv fCode iCount mcVar var isI isP isB avgQP qComp avgIITex avgPITex avgPPTex avgBPTex avgTex.

‘`maxrate` *integer* (*encoding,audio,video*)’

Set max bitrate tolerance (in bits/s). Requires bufsize to be set.

‘`minrate` *integer* (*encoding,audio,video*)’

Set min bitrate tolerance (in bits/s). Most useful in setting up a CBR encode. It is of little use elsewise.

‘`bufsize` *integer* (*encoding,audio,video*)’

Set ratecontrol buffer size (in bits).

‘`rc_buf_aggressivity` *float* (*encoding,video*)’

Currently useless.

‘`i_qfactor` *float* (*encoding,video*)’

Set QP factor between P and I frames.

‘`i_qoffset` *float* (*encoding,video*)’

Set QP offset between P and I frames.

‘`rc_init_cplx` *float* (*encoding,video*)’

Set initial complexity for 1-pass encoding.

‘`dct` *integer* (*encoding,video*)’

Set DCT algorithm.

Possible values:

‘`auto`’

autoselect a good one (default)

‘`fastint`’

fast integer

‘`int`’

accurate integer

‘`mmx`’
‘`altivec`’
‘`faan`’

floating point AAN DCT

‘`lumi_mask` *`float`* (*`encoding,video`*)’

Compress bright areas stronger than medium ones.

‘`tcplx_mask` *`float`* (*`encoding,video`*)’

Set temporal complexity masking.

‘`scplx_mask` *`float`* (*`encoding,video`*)’

Set spatial complexity masking.

‘`p_mask` *`float`* (*`encoding,video`*)’

Set inter masking.

‘`dark_mask` *`float`* (*`encoding,video`*)’

Compress dark areas stronger than medium ones.

‘`idct` *`integer`* (*`decoding/encoding,video`*)’

Select IDCT implementation.

Possible values:

‘`auto`’
‘`int`’
‘`simple`’
‘`simplemmx`’

‘`libmpeg2mmx`’
‘`mmi`’
‘`arm`’
‘`altivec`’
‘`sh4`’
‘`simplearm`’
‘`simplearmv5te`’
‘`simplearmv6`’
‘`simpleneon`’
‘`simplealpha`’
‘`h264`’
‘`vp3`’
‘`ipp`’
‘`xvidmmx`’
‘`faani`’

floating point AAN IDCT

‘`slice_count` *`integer`*’
‘`ec` *`flags`* `(`*`decoding,video`*`)`’

Set error concealment strategy.

Possible values:

‘`guess_mvs`’

iterative motion vector (MV) search (slow)

‘`deblock`’

use strong deblock filter for damaged MBs

‘`bits_per_coded_sample` *`integer`*’
‘`pred` *`integer`* `(`*`encoding,video`*`)`’

Set prediction method.

Possible values:

‘`left`’
‘`plane`’
‘`median`’
‘`aspect` *`rational number`* `(`*`encoding,video`*`)`’

Set sample aspect ratio.

‘`debug` *flags* (*decoding/encoding,audio,video,subtitles*)’

Print specific debug info.

Possible values:

‘`pict`’

picture info

‘`rc`’

rate control

‘`bitstream`’
‘`mb_type`’

macroblock (MB) type

‘`qp`’

per-block quantization parameter (QP)

‘`mv`’

motion vector

‘`dct_coeff`’
‘`skip`’
‘`startcode`’
‘`pts`’
‘`er`’

error recognition

‘`mmco`’

memory management control operations (H.264)

‘`bugs`’
‘`vis_qp`’

visualize quantization parameter (QP), lower QP are tinted greener

‘`vis_mb_type`’

visualize block types

‘`buffers`’

picture buffer allocations

‘`thread_ops`’

threading operations

‘`vismv` *`integer`* `(`*`decoding,video`*`)`’

Visualize motion vectors (MVs).

Possible values:

‘`pf`’

forward predicted MVs of P-frames

‘`bf`’

forward predicted MVs of B-frames

‘`bb`’

backward predicted MVs of B-frames

‘`cmp` *`integer`* `(`*`encoding,video`*`)`’

Set full pel me compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’
‘`subcmp` *`integer`* (*`encoding,video`*)’

Set sub pel me compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’

‘`mbcmp` *`integer`* `(`*`encoding,video`*`)`’

Set macroblock compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’
‘`ildctcmp` *`integer`* `(`*`encoding,video`*`)`’

Set interlaced dct compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’

‘`dia_size` *`integer`* `(`*`encoding,video`*`)`’

Set diamond type & size for motion estimation.

‘`last_pred` *`integer`* `(`*`encoding,video`*`)`’

Set amount of motion predictors from the previous frame.

‘`preme` *`integer`* `(`*`encoding,video`*`)`’

Set pre motion estimation.

‘`precmp` *`integer`* `(`*`encoding,video`*`)`’

Set pre motion estimation compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’

‘`pre_dia_size` *integer* (*encoding,video*)’

Set diamond type & size for motion estimation pre-pass.

‘`subq` *integer* (*encoding,video*)’

Set sub pel motion estimation quality.

‘`dtg_active_format` *integer*’
‘`me_range` *integer* (*encoding,video*)’

Set limit motion vectors range (1023 for DivX player).

‘`ibias` *integer* (*encoding,video*)’

Set intra quant bias.

‘`pbias` *integer* (*encoding,video*)’

Set inter quant bias.

‘`color_table_id` *integer*’
‘`global_quality` *integer* (*encoding,audio,video*)’
‘`coder` *integer* (*encoding,video*)’

Possible values:

‘`vlc`’

variable length coder / huffman coder

‘`ac`’

arithmetic coder

‘`raw`’

raw (no encoding)

‘`rle`’

run-length coder

‘`deflate`’

deflate-based coder

‘`context` *integer* (*encoding,video*)’

Set context model.

‘`slice_flags` *`integer`*’
‘`xvmc_acceleration` *`integer`*’
‘`mbd` *`integer`* (*`encoding,video`*)’

Set macroblock decision algorithm (high quality mode).

Possible values:

‘`simple`’

use mbcmp (default)

‘`bits`’

use fewest bits

‘`rd`’

use best rate distortion

‘`stream_codec_tag` *`integer`*’
‘`sc_threshold` *`integer`* (*`encoding,video`*)’

Set scene change threshold.

‘`lmin` *`integer`* (*`encoding,video`*)’

Set min lagrange factor (VBR).

‘`lmax` *`integer`* (*`encoding,video`*)’

Set max lagrange factor (VBR).

‘`nr` *`integer`* (*`encoding,video`*)’

Set noise reduction.

‘`rc_init_occupancy` *`integer`* (*`encoding,video`*)’

Set number of bits which should be loaded into the rc buffer before decoding starts.

‘`inter_threshold` *`integer`* (*`encoding,video`*)’
‘`flags2` *`flags`* (*`decoding/encoding,audio,video`*)’

Possible values:

‘`fast`’

allow non spec compliant speedup tricks

‘`sgop`’

Deprecated, use mpegvideo private options instead

‘`noout`’

skip bitstream encoding

‘`local_header`’

place global headers at every keyframe instead of in extradata

‘`chunks`’

Frame data might be split into multiple chunks

‘`showall`’

Show all frames before the first keyframe

‘`skiprd`’

Deprecated, use mpegvideo private options instead

‘`error` *`integer`* `(`*`encoding,video`*`)`’
‘`qns` *`integer`* `(`*`encoding,video`*`)`’

Deprecated, use mpegvideo private options instead.

‘`threads` *`integer`* `(`*`decoding/encoding,video`*`)`’

Possible values:

‘`auto`’

detect a good number of threads

‘`me_threshold` *`integer`* `(`*`encoding,video`*`)`’

Set motion estimation threshold.

‘`mb_threshold` *`integer`* `(`*`encoding,video`*`)`’

Set macroblock threshold.

‘`dc` *`integer`* `(`*`encoding,video`*`)`’

Set intra_dc_precision.

‘`nssew` *`integer`* `(`*`encoding,video`*`)`’

Set nsse weight.

‘`skip_top` *`integer`* `(`*`decoding,video`*`)`’

Set number of macroblock rows at the top which are skipped.

‘`skip_bottom` *`integer`* `(`*`decoding,video`*`)`’

Set number of macroblock rows at the bottom which are skipped.

‘`profile` *`integer`* `(`*`encoding,audio,video`*`)`’

Possible values:

‘`unknown`’
‘`aac_main`’
‘`aac_low`’
‘`aac_ssr`’
‘`aac_ltp`’
‘`aac_he`’
‘`aac_he_v2`’
‘`aac_ld`’
‘`aac_eld`’
‘`dts`’
‘`dts_es`’
‘`dts_96_24`’
‘`dts_hd_hra`’
‘`dts_hd_ma`’

‘`level` *`integer`* `(`*`encoding,audio,video`*`)`’

Possible values:

‘`unknown`’

‘`lowres` *`integer`* `(`*`decoding,audio,video`*`)`’

Decode at 1= 1/2, 2=1/4, 3=1/8 resolutions.

‘`skip_threshold` *`integer`* `(`*`encoding,video`*`)`’

Set frame skip threshold.

‘`skip_factor` *integer* (*encoding,video*)’

Set frame skip factor.

‘`skip_exp` *integer* (*encoding,video*)’

Set frame skip exponent.

‘`skipcmp` *integer* (*encoding,video*)’

Set frame skip compare function.

Possible values:

‘`sad`’

sum of absolute differences, fast (default)

‘`sse`’

sum of squared errors

‘`satd`’

sum of absolute Hadamard transformed differences

‘`dct`’

sum of absolute DCT transformed differences

‘`psnr`’

sum of squared quantization errors (avoid, low quality)

‘`bit`’

number of bits needed for the block

‘`rd`’

rate distortion optimal, slow

‘`zero`’

0

‘`vsad`’

sum of absolute vertical differences

‘`vsse`’

sum of squared vertical differences

‘`nsse`’

noise preserving sum of squared differences

‘`w53`’

5/3 wavelet, only used in snow

‘`w97`’

9/7 wavelet, only used in snow

‘`dctmax`’
‘`chroma`’

‘`border_mask` *float* (*encoding,video*)’

Increase the quantizer for macroblocks close to borders.

‘`mblmin` *integer* (*encoding,video*)’

Set min macroblock lagrange factor (VBR).

‘`mblmax` *integer* (*encoding,video*)’

Set max macroblock lagrange factor (VBR).

‘`mepc` *integer* (*encoding,video*)’

Set motion estimation bitrate penalty compensation (1.0 = 256).

‘`skip_loop_filter` *integer* (*decoding,video*)’
‘`skip_idct` *integer* (*decoding,video*)’
‘`skip_frame` *integer* (*decoding,video*)’

Make decoder discard processing depending on the frame type selected by the option value.

‘`skip_loop_filter`’ skips frame loop filtering, ‘`skip_idct`’ skips frame IDCT/dequantization, ‘`skip_frame`’ skips decoding.

Possible values:

‘`none`’

Discard no frame.

‘`default`’

Discard useless frames like 0-sized frames.

‘`noref`’

Discard all non-reference frames.

‘`bidir`’

Discard all bidirectional frames.

‘`nokey`’

Discard all frames excepts keyframes.

‘`all`’

Discard all frames.

Default value is ‘`default`’.

‘`bidir_refine` *`integer`* (*`encoding,video`*)’

Refine the two motion vectors used in bidirectional macroblocks.

‘`brd_scale` *`integer`* (*`encoding,video`*)’

Downscale frames for dynamic B-frame decision.

‘`keyint_min` *`integer`* (*`encoding,video`*)’

Set minimum interval between IDR-frames.

‘`refs` *`integer`* (*`encoding,video`*)’

Set reference frames to consider for motion compensation.

‘`chromaoffset` *`integer`* (*`encoding,video`*)’

Set chroma qp offset from luma.

‘`trellis` *`integer`* (*`encoding,audio,video`*)’

Set rate-distortion optimal quantization.

‘`sc_factor` *`integer`* (*`encoding,video`*)’

Set value multiplied by qscale for each frame and added to scene_change_score.

‘`mv0_threshold` *integer* (*encoding,video*)’
‘`b_sensitivity` *integer* (*encoding,video*)’

Adjust sensitivity of b_frame_strategy 1.

‘`compression_level` *integer* (*encoding,audio,video*)’
‘`min_prediction_order` *integer* (*encoding,audio*)’
‘`max_prediction_order` *integer* (*encoding,audio*)’
‘`timecode_frame_start` *integer* (*encoding,video*)’

Set GOP timecode frame start number, in non drop frame format.

‘`request_channels` *integer* (*decoding,audio*)’

Set desired number of audio channels.

‘`bits_per_raw_sample` *integer*’
‘`channel_layout` *integer* (*decoding/encoding,audio*)’

Possible values:

‘`request_channel_layout` *integer* (*decoding,audio*)’

Possible values:

‘`rc_max_vbv_use` *float* (*encoding,video*)’
‘`rc_min_vbv_use` *float* (*encoding,video*)’
‘`ticks_per_frame` *integer* (*decoding/encoding,audio,video*)’
‘`color_primaries` *integer* (*decoding/encoding,video*)’
‘`color_trc` *integer* (*decoding/encoding,video*)’
‘`colorspace` *integer* (*decoding/encoding,video*)’
‘`color_range` *integer* (*decoding/encoding,video*)’
‘`chroma_sample_location` *integer* (*decoding/encoding,video*)’
‘`log_level_offset` *integer*’

Set the log level offset.

‘`slices` *integer* (*encoding,video*)’

Number of slices, used in parallelized encoding.

‘`thread_type` *flags* (*decoding/encoding,video*)’

Select multithreading type.

Possible values:

‘`slice`’
‘`frame`’

‘`audio_service_type` *`integer`* `(`*`encoding,audio`*`)`’

Set audio service type.

Possible values:

‘`ma`’

Main Audio Service

‘`ef`’

Effects

‘`vi`’

Visually Impaired

‘`hi`’

Hearing Impaired

‘`di`’

Dialogue

‘`co`’

Commentary

‘`em`’

Emergency

‘`vo`’

Voice Over

‘`ka`’

Karaoke

‘`request_sample_fmt` *`sample_fmt`* `(`*`decoding,audio`*`)`’

Set sample format audio decoders should prefer. Default value is `none`.

‘`pkt_timebase` *`rational number`*’
‘`sub_charenc` *`encoding`* `(`*`decoding,subtitles`*`)`’

Set the input subtitles character encoding.

3. Decoders

Decoders are configured elements in FFmpeg which allow the decoding of multimedia streams.

When you configure your FFmpeg build, all the supported native decoders are enabled by default. Decoders requiring an external library must be enabled manually via the corresponding `--enable-lib` option. You can list all available decoders using the configure option `--list-decoders`.

You can disable all the decoders with the configure option `--disable-decoders` and selectively enable / disable single decoders with the options `--enable-decoder=`*`DECODER`* / `--disable-decoder=`*`DECODER`*.

The option `-codecs` of the ff* tools will display the list of enabled decoders.

4. Video Decoders

A description of some of the currently available video decoders follows.

4.1 rawvideo

Raw video decoder.

This decoder decodes rawvideo streams.

4.1.1 Options

‘`top` *`top_field_first`*’

Specify the assumed field type of the input video.

‘`-1`’

the video is assumed to be progressive (default)

‘`0`’

bottom-field-first is assumed

‘1’

top-field-first is assumed

5. Audio Decoders

5.1 ffwavesynth

Internal wave synthetizer.

This decoder generates wave patterns according to predefined sequences. Its use is purely internal and the format of the data it accepts is not publicly documented.

5.2 libcelt

libcelt decoder wrapper.

libcelt allows libavcodec to decode the Xiph CELT ultra-low delay audio codec. Requires the presence of the libcelt headers and library during configuration. You need to explicitly configure the build with `--enable-libcelt`.

5.3 libgsm

libgsm decoder wrapper.

libgsm allows libavcodec to decode the GSM full rate audio codec. Requires the presence of the libgsm headers and library during configuration. You need to explicitly configure the build with `--enable-libgsm`.

This decoder supports both the ordinary GSM and the Microsoft variant.

5.4 libilbc

libilbc decoder wrapper.

libilbc allows libavcodec to decode the Internet Low Bitrate Codec (iLBC) audio codec. Requires the presence of the libilbc headers and library during configuration. You need to explicitly configure the build with `--enable-libilbc`.

5.4.1 Options

The following option is supported by the libilbc wrapper.

‘`enhance`’

Enable the enhancement of the decoded audio when set to 1. The default value is 0 (disabled).

5.5 libopencore-amrnb

libopencore-amrnb decoder wrapper.

libopencore-amrnb allows libavcodec to decode the Adaptive Multi-Rate Narrowband audio codec. Using it requires the presence of the libopencore-amrnb headers and library during configuration. You need to explicitly configure the build with `--enable-libopencore-amrnb`.

An FFmpeg native decoder for AMR-NB exists, so users can decode AMR-NB without this library.

5.6 libopencore-amrwb

libopencore-amrwb decoder wrapper.

libopencore-amrwb allows libavcodec to decode the Adaptive Multi-Rate Wideband audio codec. Using it requires the presence of the libopencore-amrwb headers and library during configuration. You need to explicitly configure the build with `--enable-libopencore-amrwb`.

An FFmpeg native decoder for AMR-WB exists, so users can decode AMR-WB without this library.

5.7 libopus

libopus decoder wrapper.

libopus allows libavcodec to decode the Opus Interactive Audio Codec. Requires the presence of the libopus headers and library during configuration. You need to explicitly configure the build with `--enable-libopus`.

6. Subtitles Decoders

6.1 dvdsub

This codec decodes the bitmap subtitles used in DVDs; the same subtitles can also be found in VobSub file pairs and in some Matroska files.

6.1.1 Options

‘`palette`’

Specify the global palette used by the bitmaps. When stored in VobSub, the palette is normally specified in the index file; in Matroska, the palette is stored in the codec extra-data in the same format as in VobSub. In DVDs, the palette is stored in the IFO file, and therefore not available when reading from dumped VOB files.

The format for this option is a string containing 16 24-bits hexadecimal numbers (without 0x prefix) separated by comas, for example `0d00ee, ee450d, 101010, eaeaea, 0ce60b, ec14ed, ebff0b, 0d617a, 7b7b7b, d1d1d1, 7b2a0e, 0d950c, 0f007b, cf0dec, cfa80c, 7c127b`.

7. Encoders

Encoders are configured elements in FFmpeg which allow the encoding of multimedia streams.

When you configure your FFmpeg build, all the supported native encoders are enabled by default. Encoders requiring an external library must be enabled manually via the corresponding `--enable-lib` option. You can list all available encoders using the configure option `--list-encoders`.

You can disable all the encoders with the configure option `--disable-encoders` and selectively enable / disable single encoders with the options `--enable-encoder=`*`ENCODER`* / `--disable-encoder=`*`ENCODER`*.

The option `-codecs` of the ff* tools will display the list of enabled encoders.

8. Audio Encoders

A description of some of the currently available audio encoders follows.

8.1 ac3 and ac3_fixed

AC-3 audio encoders.

These encoders implement part of ATSC A/52:2010 and ETSI TS 102 366, as well as the undocumented RealAudio 3 (a.k.a. dnet).

The *ac3* encoder uses floating-point math, while the *ac3_fixed* encoder only uses fixed-point integer math. This does not mean that one is always faster, just that one or the other may be better suited to a particular system. The floating-point encoder will generally produce better quality audio for a given bitrate. The *ac3_fixed* encoder is not the default codec for any of the output formats, so it must be specified explicitly using the option `-acodec ac3_fixed` in order to use it.

8.1.1 AC-3 Metadata

The AC-3 metadata options are used to set parameters that describe the audio, but in most cases do not affect the audio encoding itself. Some of the options do directly affect or influence the decoding and playback of the resulting bitstream, while others are just for informational purposes. A few of the options will add bits to the output stream that could otherwise be used for audio data, and will thus affect the quality of the output. Those will be indicated accordingly with a note in the option list below.

These parameters are described in detail in several publicly-available documents.

- A/52:2010 - Digital Audio Compression (AC-3) (E-AC-3) Standard
- A/54 - Guide to the Use of the ATSC Digital Television Standard
- Dolby Metadata Guide
- Dolby Digital Professional Encoding Guidelines

8.1.1.1 Metadata Control Options

‘`-per_frame_metadata` *`boolean`*’

Allow Per-Frame Metadata. Specifies if the encoder should check for changing metadata for each frame.

‘`0`’

The metadata values set at initialization will be used for every frame in the stream. (default)

‘`1`’

Metadata values can be changed before encoding each frame.

8.1.1.2 Downmix Levels

‘`-center_mixlev` *`level`*’

Center Mix Level. The amount of gain the decoder should apply to the center channel when downmixing to stereo. This field will only be written to the bitstream if a center channel is present. The value is specified as a scale factor. There are 3 valid values:

‘`0.707`’

Apply -3dB gain

‘`0.595`’

Apply -4.5dB gain (default)

‘`0.500`’

Apply -6dB gain

‘`-surround_mixlev` *`level`*’

Surround Mix Level. The amount of gain the decoder should apply to the surround channel(s) when downmixing to stereo. This field will only be written to the bitstream if one or more surround channels are present. The value is specified as a scale factor. There are 3 valid values:

‘`0.707`’

Apply -3dB gain

‘`0.500`’

Apply -6dB gain (default)

‘`0.000`’

Silence Surround Channel(s)

8.1.1.3 Audio Production Information

Audio Production Information is optional information describing the mixing environment. Either none or both of the fields are written to the bitstream.

‘`-mixing_level` *`number`*’

Mixing Level. Specifies peak sound pressure level (SPL) in the production environment when the mix was mastered. Valid values are 80 to 111, or -1 for unknown or not indicated. The default value is -1, but that value cannot be used if the Audio Production Information is written to the bitstream. Therefore, if the `room_type` option is not the default value, the `mixing_level` option must not be -1.

‘`-room_type` *`type`*’

Room Type. Describes the equalization used during the final mixing session at the studio or on the dubbing stage. A large room is a dubbing stage with the industry standard X-curve equalization; a small room has flat equalization. This field will not be written to the bitstream if both the `mixing_level` option and the `room_type` option have the default values.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`large`’

Large Room

‘`2`’
‘`small`’

Small Room

8.1.1.4 Other Metadata Options

‘`-copyright` *`boolean`*’

Copyright Indicator. Specifies whether a copyright exists for this audio.

‘`0`’
‘`off`’

No Copyright Exists (default)

‘`1`’
‘`on`’

Copyright Exists

‘`-dialnorm` *`value`*’

Dialogue Normalization. Indicates how far the average dialogue level of the program is below digital 100% full scale (0 dBFS). This parameter determines a level shift during audio reproduction that sets the average volume of the dialogue to a preset level. The goal is to match volume level between program sources. A value of -31dB will result in no volume level change, relative to the source volume, during audio reproduction. Valid values are whole numbers in the range -31 to -1, with -31 being the default.

‘`-dsur_mode` *`mode`*’

Dolby Surround Mode. Specifies whether the stereo signal uses Dolby Surround (Pro Logic). This field will only be written to the bitstream if the audio stream is stereo. Using this option does **NOT** mean the encoder will actually apply Dolby Surround processing.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`off`’

Not Dolby Surround Encoded

‘2’
‘on’

Dolby Surround Encoded

‘-original *boolean*’

Original Bit Stream Indicator. Specifies whether this audio is from the original source and not a copy.

‘0’
‘off’

Not Original Source

‘1’
‘on’

Original Source (default)

8.1.2 Extended Bitstream Information

The extended bitstream options are part of the Alternate Bit Stream Syntax as specified in Annex D of the A/52:2010 standard. It is grouped into 2 parts. If any one parameter in a group is specified, all values in that group will be written to the bitstream. Default values are used for those that are written but have not been specified. If the mixing levels are written, the decoder will use these values instead of the ones specified in the center_mixlev and surround_mixlev options if it supports the Alternate Bit Stream Syntax.

8.1.2.1 Extended Bitstream Information - Part 1

‘-dmix_mode *mode*’

Preferred Stereo Downmix Mode. Allows the user to select either Lt/Rt (Dolby Surround) or Lo/Ro (normal stereo) as the preferred stereo downmix mode.

‘0’
‘notindicated’

Not Indicated (default)

‘1’
‘ltrt’

Lt/Rt Downmix Preferred

‘2’
‘loro’

Lo/Ro Downmix Preferred

‘`-ltrt_cmixlev` *`level`*’

Lt/Rt Center Mix Level. The amount of gain the decoder should apply to the center channel when downmixing to stereo in Lt/Rt mode.

‘`1.414`’

Apply +3dB gain

‘`1.189`’

Apply +1.5dB gain

‘`1.000`’

Apply 0dB gain

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain (default)

‘`0.500`’

Apply -6.0dB gain

‘`0.000`’

Silence Center Channel

‘`-ltrt_surmixlev` *`level`*’

Lt/Rt Surround Mix Level. The amount of gain the decoder should apply to the surround channel(s) when downmixing to stereo in Lt/Rt mode.

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain

‘`0.500`’

Apply -6.0dB gain (default)

‘`0.000`’

Silence Surround Channel(s)

‘`-loro_cmixlev` *`level`*’

Lo/Ro Center Mix Level. The amount of gain the decoder should apply to the center channel when downmixing to stereo in Lo/Ro mode.

‘`1.414`’

Apply +3dB gain

‘`1.189`’

Apply +1.5dB gain

‘`1.000`’

Apply 0dB gain

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain (default)

‘`0.500`’

Apply -6.0dB gain

‘`0.000`’

Silence Center Channel

‘`-loro_surmixlev` *`level`*’

Lo/Ro Surround Mix Level. The amount of gain the decoder should apply to the surround channel(s) when downmixing to stereo in Lo/Ro mode.

‘`0.841`’

Apply -1.5dB gain

‘`0.707`’

Apply -3.0dB gain

‘`0.595`’

Apply -4.5dB gain

‘`0.500`’

Apply -6.0dB gain (default)

‘`0.000`’

Silence Surround Channel(s)

8.1.2.2 Extended Bitstream Information - Part 2

‘`-dsurex_mode` *`mode`*’

Dolby Surround EX Mode. Indicates whether the stream uses Dolby Surround EX (7.1 matrixed to 5.1). Using this option does **NOT** mean the encoder will actually apply Dolby Surround EX processing.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`on`’

Dolby Surround EX Off

‘`2`’

‘`off`’

Dolby Surround EX On

‘`-dheadphone_mode` *`mode`*’

Dolby Headphone Mode. Indicates whether the stream uses Dolby Headphone encoding (multi-channel matrixed to 2.0 for use with headphones). Using this option does **NOT** mean the encoder will actually apply Dolby Headphone processing.

‘`0`’
‘`notindicated`’

Not Indicated (default)

‘`1`’
‘`on`’

Dolby Headphone Off

‘`2`’
‘`off`’

Dolby Headphone On

‘`-ad_conv_type` *`type`*’

A/D Converter Type. Indicates whether the audio has passed through HDCD A/D conversion.

‘`0`’
‘`standard`’

Standard A/D Converter (default)

‘`1`’
‘`hdcd`’

HDCD A/D Converter

8.1.3 Other AC-3 Encoding Options

‘`-stereo_rematrixing` *`boolean`*’

Stereo Rematrixing. Enables/Disables use of rematrixing for stereo input. This is an optional AC-3 feature that increases quality by selectively encoding the left/right channels as mid/side. This option is enabled by default, and it is highly recommended that it be left as enabled except for testing purposes.

8.1.4 Floating-Point-Only AC-3 Encoding Options

These options are only valid for the floating-point encoder and do not exist for the fixed-point encoder due to the corresponding features not being implemented in fixed-point.

'`-channel_coupling` *`boolean`*'

> Enables/Disables use of channel coupling, which is an optional AC-3 feature that increases quality by combining high frequency information from multiple channels into a single channel. The per-channel high frequency information is sent with less accuracy in both the frequency and time domains. This allows more bits to be used for lower frequencies while preserving enough information to reconstruct the high frequencies. This option is enabled by default for the floating-point encoder and should generally be left as enabled except for testing purposes or to increase encoding speed.
>
> '`-1`'
> '`auto`'
>
> > Selected by Encoder (default)
>
> '`0`'
> '`off`'
>
> > Disable Channel Coupling
>
> '`1`'
> '`on`'
>
> > Enable Channel Coupling

'`-cpl_start_band` *`number`*'

> Coupling Start Band. Sets the channel coupling start band, from 1 to 15. If a value higher than the bandwidth is used, it will be reduced to 1 less than the coupling end band. If *auto* is used, the start band will be determined by the encoder based on the bit rate, sample rate, and channel layout. This option has no effect if channel coupling is disabled.
>
> '`-1`'
> '`auto`'
>
> > Selected by Encoder (default)

8.2 libmp3lame

LAME (Lame Ain't an MP3 Encoder) MP3 encoder wrapper.

Requires the presence of the libmp3lame headers and library during configuration. You need to explicitly configure the build with `--enable-libmp3lame`.

8.2.1 Option Mapping

The following options are supported by the libmp3lame wrapper, the LAME-equivalent options follow the FFmpeg ones.

FFmpeg	LAME
b	b Set bitrate expressed in bits/s, LAME `bitrate` is expressed in kilobits/s.
q	V Set quality setting for VBR.
compression_level	q Set algorithm quality. Valid arguments are integers in the 0-9 range.
reservoir	N.A. Enable use of bit reservoir. LAME has this enabled by default.
joint_stereo	-m j Enable the encoder to use (on a frame by frame basis) either L/R stereo or mid/side stereo.

8.3 libopencore-amrnb

OpenCORE Adaptive Multi-Rate Narrowband encoder.

Requires the presence of the libopencore-amrnb headers and library during configuration. You need to explicitly configure the build with `--enable-libopencore-amrnb --enable-version3`.

This is a mono-only encoder. Officially it only supports 8000Hz sample rate, but you can override it by setting ‘`strict`’ to ‘`unofficial`’ or lower.

8.3.1 Options

‘`b`’

Set bitrate in bits per second. Only the following bitrates are supported, otherwise libavcodec will round to the nearest valid bitrate.

‘`4750`’
‘`5150`’
‘`5900`’
‘`6700`’
‘`7400`’
‘`7950`’
‘`10200`’
‘`12200`’

‘`dtx`’

Allow discontinuous transmission (generate comfort noise) when set to 1. The default value is 0 (disabled).

8.4 libtwolame

TwoLAME MP2 encoder wrapper.

Requires the presence of the libtwolame headers and library during configuration. You need to explicitly configure the build with `--enable-libtwolame`.

8.4.1 Options Mapping

The following options are supported by the libtwolame wrapper. The TwoLAME-equivalent options follow the FFmpeg ones and are in parentheses.

‘`b`’

> (b) Set bitrate in bits/s. Note that FFmpeg `b` option is expressed in bits/s, twolame `b` in kilobits/s. The default value is 128k.

‘`q`’

> (V) Set quality for experimental VBR support. Maximum value range is from -50 to 50, useful range is from -10 to 10.

‘`mode`’

> (mode) Set MPEG mode. Possible values:
>
> ‘`auto`’
>
> > Choose mode automatically based on the input. This is the default.
>
> ‘`stereo`’
>
> > Stereo
>
> ‘`joint_stereo`’
>
> > Joint stereo
>
> ‘`dual_channel`’
>
> > Dual channel
>
> ‘`mono`’

Mono

‘`psymodel`’

(psyc-mode) Set psychoacoustic model to use in encoding. The argument must be an integer between -1 and 4, inclusive. The higher the value, the better the quality. The default value is 3.

‘`energy_levels`’

(energy) Enable energy levels extensions when set to 1. The default value is 0 (disabled).

‘`error_protection`’

(protect) Enable CRC error protection when set to 1. The default value is 0 (disabled).

‘`copyright`’

(copyright) Set MPEG audio copyright flag when set to 1. The default value is 0 (disabled).

‘`original`’

(original) Set MPEG audio original flag when set to 1. The default value is 0 (disabled).

8.5 libvo-aacenc

VisualOn AAC encoder.

Requires the presence of the libvo-aacenc headers and library during configuration. You need to explicitly configure the build with `--enable-libvo-aacenc --enable-version3`.

8.5.1 Options

The VisualOn AAC encoder only support encoding AAC-LC and up to 2 channels. It is also CBR-only. It is considered to be worse than the native experimental FFmpeg AAC encoder.

‘`b`’

Bitrate.

8.6 libvo-amrwbenc

VisualOn Adaptive Multi-Rate Wideband encoder.

Requires the presence of the libvo-amrwbenc headers and library during configuration. You need to explicitly configure the build with `--enable-libvo-amrwbenc --enable-version3`.

This is a mono-only encoder. Officially it only supports 16000Hz sample rate, but you can override it by setting ‘`strict`’ to ‘`unofficial`’ or lower.

8.6.1 Options

‘`b`’

Set bitrate in bits/s. Only the following bitrates are supported, otherwise libavcodec will round to the nearest valid bitrate.

‘`6600`’
‘`8850`’
‘`12650`’
‘`14250`’
‘`15850`’
‘`18250`’
‘`19850`’
‘`23050`’
‘`23850`’

‘`dtx`’

Allow discontinuous transmission (generate comfort noise) when set to 1. The default value is 0 (disabled).

8.7 libopus

libopus Opus Interactive Audio Codec encoder wrapper.

Requires the presence of the libopus headers and library during configuration. You need to explicitly configure the build with `--enable-libopus`.

8.7.1 Option Mapping

Most libopus options are modeled after the `opusenc` utility from opus-tools. The following is an option mapping chart describing options supported by the libopus wrapper, and their `opusenc`-equivalent in parentheses.

‘`b (`*`bitrate`*`)`’

Set the bit rate in bits/s. FFmpeg’s ‘`b`’ option is expressed in bits/s, while `opusenc`’s ‘`bitrate`’ in kilobits/s.

‘`vbr (`*`vbr`*`, `*`hard-cbr`*`, and `*`cvbr`*`)`’

Set VBR mode. The FFmpeg ‘`vbr`’ option has the following valid arguments, with the their `opusenc` equivalent options in parentheses:

‘`off (`*hard-cbr*`)`’

Use constant bit rate encoding.

‘`on (`*vbr*`)`’

Use variable bit rate encoding (the default).

‘`constrained (`*cvbr*`)`’

Use constrained variable bit rate encoding.

‘`compression_level (`*comp*`)`’

Set encoding algorithm complexity. Valid options are integers in the 0-10 range. 0 gives the fastest encodes but lower quality, while 10 gives the highest quality but slowest encoding. The default is 10.

‘`frame_duration (`*framesize*`)`’

Set maximum frame size, or duration of a frame in milliseconds. The argument must be exactly the following: 2.5, 5, 10, 20, 40, 60. Smaller frame sizes achieve lower latency but less quality at a given bitrate. Sizes greater than 20ms are only interesting at fairly low bitrates. The default of FFmpeg is 10ms, but is 20ms in `opusenc`.

‘`packet_loss (`*expect-loss*`)`’

Set expected packet loss percentage. The default is 0.

‘`application (N.A.)`’

Set intended application type. Valid options are listed below:

‘`voip`’

Favor improved speech intelligibility.

‘`audio`’

Favor faithfulness to the input (the default).

‘`lowdelay`’

Restrict to only the lowest delay modes.

‘`cutoff (N.A.)`’

Set cutoff bandwidth in Hz. The argument must be exactly one of the following: 4000, 6000, 8000, 12000, or 20000, corresponding to narrowband, mediumband, wideband, super wideband, and fullband respectively. The default is 0 (cutoff disabled).

8.8 libwavpack

A wrapper providing WavPack encoding through libwavpack.

Only lossless mode using 32-bit integer samples is supported currently. The ‘`compression_level`’ option can be used to control speed vs. compression tradeoff, with the values mapped to libwavpack as follows:

‘`0`’

Fast mode - corresponding to the wavpack ‘`-f`’ option.

‘`1`’

Normal (default) settings.

‘`2`’

High quality - corresponding to the wavpack ‘`-h`’ option.

‘`3`’

Very high quality - corresponding to the wavpack ‘`-hh`’ option.

‘`4-8`’

Same as 3, but with extra processing enabled - corresponding to the wavpack ‘`-x`’ option. I.e. 4 is the same as ‘`-x2`’ and 8 is the same as ‘`-x6`’.

9. Video Encoders

A description of some of the currently available video encoders follows.

9.1 libtheora

Theora format supported through libtheora.

Requires the presence of the libtheora headers and library during configuration. You need to explicitly configure the build with `--enable-libtheora`.

9.1.1 Options

The following global options are mapped to internal libtheora options which affect the quality and the bitrate of the encoded stream.

‘b’

Set the video bitrate, only works if the qscale flag in ‘flags’ is not enabled.

‘flags’

Used to enable constant quality mode encoding through the ‘qscale’ flag, and to enable the pass1 and pass2 modes.

‘g’

Set the GOP size.

‘global_quality’

Set the global quality in lambda units, only works if the qscale flag in ‘flags’ is enabled. The value is clipped in the [0 - 10*FF_QP2LAMBDA] range, and then multiplied for 6.3 to get a value in the native libtheora range [0-63]. A higher value corresponds to a higher quality.

For example, to set maximum constant quality encoding with ffmpeg:

```
ffmpeg -i INPUT -flags:v qscale -global_quality:v "10*QP2LAMBDA" -codec:v libtheora OUTPUT.ogg
```

9.2 libvpx

VP8 format supported through libvpx.

Requires the presence of the libvpx headers and library during configuration. You need to explicitly configure the build with --enable-libvpx.

9.2.1 Options

Mapping from FFmpeg to libvpx options with conversion notes in parentheses.

‘threads’

g_threads

‘profile’

g_profile

‘vb’

rc_target_bitrate

‘g’

kf_max_dist

‘keyint_min’

kf_min_dist

‘qmin’

rc_min_quantizer

‘qmax’

rc_max_quantizer

‘bufsize, vb’

rc_buf_sz (bufsize * 1000 / vb)

rc_buf_optimal_sz (bufsize * 1000 / vb * 5 / 6)

‘rc_init_occupancy, vb’

rc_buf_initial_sz (rc_init_occupancy * 1000 / vb)

‘rc_buffer_aggressivity’

rc_undershoot_pct

‘skip_threshold’

rc_dropframe_thresh

‘qcomp’

rc_2pass_vbr_bias_pct

‘maxrate, vb’

rc_2pass_vbr_maxsection_pct (maxrate * 100 / vb)

‘minrate, vb’

rc_2pass_vbr_minsection_pct (minrate * 100 / vb)

‘minrate, maxrate, vb’

VPX_CBR (minrate == maxrate == vb)

‘crf’

VPX_CQ, VP8E_SET_CQ_LEVEL

‘quality’
‘*best*’

VPX_DL_BEST_QUALITY

‘*good*’

VPX_DL_GOOD_QUALITY

‘*realtime*’

VPX_DL_REALTIME

‘speed’

VP8E_SET_CPUUSED

‘nr’

VP8E_SET_NOISE_SENSITIVITY

‘mb_threshold’

VP8E_SET_STATIC_THRESHOLD

‘slices’

VP8E_SET_TOKEN_PARTITIONS

‘max-intra-rate’

VP8E_SET_MAX_INTRA_BITRATE_PCT

‘force_key_frames’

VPX_EFLAG_FORCE_KF

‘Alternate reference frame related’
‘vp8flags altref’

VP8E_SET_ENABLEAUTOALTREF

‘*arnr_max_frames*’

VP8E_SET_ARNR_MAXFRAMES

‘*arnr_type*’

VP8E_SET_ARNR_TYPE

‘*arnr_strength*’

VP8E_SET_ARNR_STRENGTH

‘*rc_lookahead*’

g_lag_in_frames

‘vp8flags error_resilient’

g_error_resilient

For more information about libvpx see: http://www.webmproject.org/

9.3 libx264

x264 H.264/MPEG-4 AVC encoder wrapper.

Requires the presence of the libx264 headers and library during configuration. You need to explicitly configure the build with `--enable-libx264`.

x264 supports an impressive number of features, including 8x8 and 4x4 adaptive spatial transform, adaptive B-frame placement, CAVLC/CABAC entropy coding, interlacing (MBAFF), lossless mode, psy optimizations for detail retention (adaptive quantization, psy-RD, psy-trellis).

The FFmpeg wrapper provides a mapping for most of them using global options that match those of the encoders and provides private options for the unique encoder options. Additionally an expert override is provided to directly pass a list of key=value tuples as accepted by x264_param_parse.

9.3.1 Option Mapping

The following options are supported by the x264 wrapper, the x264-equivalent options follow the FFmpeg ones.

b	bitrate FFmpeg `b` option is expressed in bits/s, x264 `bitrate` in kilobits/s.
bf	bframes Maximum number of B-frames.
g	keyint Maximum GOP size.
qmin	qpmin
qmax	qpmax
qdiff	qpstep
qblur	qblur
qcomp	qcomp
refs	ref
sc_threshold	scenecut
trellis	trellis
nr	nr Noise reduction.
me_range	merange
me_method	me
subq	subme
b_strategy	b-adapt
keyint_min	keyint-min
coder	cabac Set coder to `ac` to use CABAC.
cmp	chroma-me Set to `chroma` to use chroma motion estimation.
threads	threads
thread_type	sliced_threads Set to `slice` to use sliced threading instead of frame threading.
flags -cgop	open-gop Set `-cgop` to use recovery points to close GOPs.
rc_init_occupancy	vbv-init Initial buffer occupancy.

9.3.2 Private Options

‘`-preset` *`string`*’

Set the encoding preset (cf. x264 –fullhelp).

‘`-tune` *string*’

Tune the encoding params (cf. x264 –fullhelp).

‘`-profile` *string*’

Set profile restrictions (cf. x264 –fullhelp).

‘`-fastfirstpass` *integer*’

Use fast settings when encoding first pass.

‘`-crf` *float*’

Select the quality for constant quality mode.

‘`-crf_max` *float*’

In CRF mode, prevents VBV from lowering quality beyond this point.

‘`-qp` *integer*’

Constant quantization parameter rate control method.

‘`-aq-mode` *integer*’

AQ method

Possible values:

‘`none`’
‘`variance`’

Variance AQ (complexity mask).

‘`autovariance`’

Auto-variance AQ (experimental).

‘`-aq-strength` *float*’

AQ strength, reduces blocking and blurring in flat and textured areas.

‘`-psy` *integer*’

Use psychovisual optimizations.

‘`-psy-rd` *string*’

Strength of psychovisual optimization, in <psy-rd>:<psy-trellis> format.

‘`-rc-lookahead` *integer*’

Number of frames to look ahead for frametype and ratecontrol.

‘`-weightb` *integer*’

Weighted prediction for B-frames.

‘`-weightp` *integer*’

Weighted prediction analysis method.

Possible values:

‘`none`’
‘`simple`’
‘`smart`’
‘`-ssim` *integer*’

Calculate and print SSIM stats.

‘`-intra-refresh` *integer*’

Use Periodic Intra Refresh instead of IDR frames.

‘`-b-bias` *integer*’

Influences how often B-frames are used.

‘`-b-pyramid` *integer*’

Keep some B-frames as references.

Possible values:

‘`none`’
‘`strict`’

Strictly hierarchical pyramid.

‘`normal`’

Non-strict (not Blu-ray compatible).

‘`-mixed-refs` *integer*’

One reference per partition, as opposed to one reference per macroblock.

‘-8x8dct *integer*’

High profile 8x8 transform.

‘-fast-pskip *integer*’
‘-aud *integer*’

Use access unit delimiters.

‘-mbtree *integer*’

Use macroblock tree ratecontrol.

‘-deblock *string*’

Loop filter parameters, in <alpha:beta> form.

‘-cplxblur *float*’

Reduce fluctuations in QP (before curve compression).

‘-partitions *string*’

A comma-separated list of partitions to consider, possible values: p8x8, p4x4, b8x8, i8x8, i4x4, none, all.

‘-direct-pred *integer*’

Direct MV prediction mode

Possible values:

‘none’
‘spatial’
‘temporal’
‘auto’

‘-slice-max-size *integer*’

Limit the size of each slice in bytes.

‘-stats *string*’

Filename for 2 pass stats.

‘-nal-hrd *integer*’

Signal HRD information (requires vbv-bufsize; cbr not allowed in .mp4).

Possible values:

‘`none`’
‘`vbr`’
‘`cbr`’

‘`x264opts` *`options`*’

Allow to set any x264 option, see `x264 --fullhelp` for a list.

options is a list of *key*=*value* couples separated by ":". In *filter* and *psy-rd* options that use ":" as a separator themselves, use "," instead. They accept it as well since long ago but this is kept undocumented for some reason.

For example to specify libx264 encoding options with `ffmpeg`:

```
ffmpeg -i foo.mpg -vcodec libx264 -x264opts keyint=123:min-keyint=20 -an out.mkv
```

For more information about libx264 and the supported options see:
http://www.videolan.org/developers/x264.html

‘`-x264-params` *`string`*’

Override the x264 configuration using a :-separated list of key=value parameters.

```
-x264-params level=30:bframes=0:weightp=0:cabac=0:ref=1:vbv-maxrate=768:vbv-bufsize=2000:analyse=all:me=umh:no-fast-pskip=1:subq=6:8x8dct=0:trellis=0
```

Encoding avpresets for common usages are provided so they can be used with the general presets system (e.g. passing the `-pre` option).

9.4 png

PNG image encoder.

9.4.1 Private options

‘`dpi` *`integer`*’

Set physical density of pixels, in dots per inch, unset by default

‘`dpm` *`integer`*’

Set physical density of pixels, in dots per meter, unset by default

9.5 ProRes

Apple ProRes encoder.

FFmpeg contains 2 ProRes encoders, the prores-aw and prores-ks encoder. The used encoder can be choosen with the `-vcodec` option.

9.5.1 Private Options for prores-ks

‘`profile` *`integer`*’

> Select the ProRes profile to encode
>
> ‘`proxy`’
> ‘`lt`’
> ‘`standard`’
> ‘`hq`’
> ‘`4444`’

‘`quant_mat` *`integer`*’

> Select quantization matrix.
>
> ‘`auto`’
> ‘`default`’
> ‘`proxy`’
> ‘`lt`’
> ‘`standard`’
> ‘`hq`’
>
> If set to *auto*, the matrix matching the profile will be picked. If not set, the matrix providing the highest quality, *default*, will be picked.

‘`bits_per_mb` *`integer`*’

> How many bits to allot for coding one macroblock. Different profiles use between 200 and 2400 bits per macroblock, the maximum is 8000.

‘`mbs_per_slice` *`integer`*’

> Number of macroblocks in each slice (1-8); the default value (8) should be good in almost all situations.

‘`vendor` *`string`*’

> Override the 4-byte vendor ID. A custom vendor ID like *apl0* would claim the stream was produced by the Apple encoder.

‘`alpha_bits` *integer*’

Specify number of bits for alpha component. Possible values are *0*, *8* and *16*. Use *0* to disable alpha plane coding.

9.5.2 Speed considerations

In the default mode of operation the encoder has to honor frame constraints (i.e. not produc frames with size bigger than requested) while still making output picture as good as possible. A frame containing a lot of small details is harder to compress and the encoder would spend more time searching for appropriate quantizers for each slice.

Setting a higher ‘`bits_per_mb`’ limit will improve the speed.

For the fastest encoding speed set the ‘`qscale`’ parameter (4 is the recommended value) and do not set a size constraint.

10. See Also

ffmpeg, ffplay, ffprobe, ffserver, libavcodec

11. Authors

The FFmpeg developers.

For details about the authorship, see the Git history of the project (git://source.ffmpeg.org/ffmpeg), e.g. by typing the command `git log` in the FFmpeg source directory, or browsing the online repository at http://source.ffmpeg.org.

Maintainers for the specific components are listed in the file ‘`MAINTAINERS`’ in the source code tree.

This document was generated by *john* on *September 24, 2013* using *texi2html 1.82*.