

# FFmpeg Codecs Documentation

## Table of Contents

- 1 Description
- 2 Codec Options
- 3 Decoders
- 4 Video Decoders
  - 4.1 hevc
  - 4.2 rawvideo
    - 4.2.1 Options
- 5 Audio Decoders
  - 5.1 ac3
    - 5.1.1 AC-3 Decoder Options
  - 5.2 flac
    - 5.2.1 FLAC Decoder options
  - 5.3 ffwavesynth
  - 5.4 libcelt
  - 5.5 libgsm
  - 5.6 libilbc
    - 5.6.1 Options
  - 5.7 libopencore-amrnb
  - 5.8 libopencore-amrwb
  - 5.9 libopus
- 6 Subtitles Decoders
  - 6.1 dvbsub
    - 6.1.1 Options
  - 6.2 dvdsup
    - 6.2.1 Options
  - 6.3 libzvbi-teletext
    - 6.3.1 Options
- 7 Encoders
- 8 Audio Encoders
  - 8.1 aac
    - 8.1.1 Options
  - 8.2 ac3 and ac3\_fixed
    - 8.2.1 AC-3 Metadata
      - 8.2.1.1 Metadata Control Options
      - 8.2.1.2 Downmix Levels
      - 8.2.1.3 Audio Production Information
      - 8.2.1.4 Other Metadata Options
    - 8.2.2 Extended Bitstream Information
      - 8.2.2.1 Extended Bitstream Information - Part 1

- 8.2.2.2 Extended Bitstream Information - Part 2
  - 8.2.3 Other AC-3 Encoding Options
  - 8.2.4 Floating-Point-Only AC-3 Encoding Options
- 8.3 flac
  - 8.3.1 Options
- 8.4 opus
  - 8.4.1 Options
- 8.5 libfdk\_aac
  - 8.5.1 Options
  - 8.5.2 Examples
- 8.6 libmp3lame
  - 8.6.1 Options
- 8.7 libopencore-amrnb
  - 8.7.1 Options
- 8.8 libopus
  - 8.8.1 Option Mapping
- 8.9 libshine
  - 8.9.1 Options
- 8.10 libtwolame
  - 8.10.1 Options
- 8.11 libvo-amrwbenc
  - 8.11.1 Options
- 8.12 libvorbis
  - 8.12.1 Options
- 8.13 libwavpack
  - 8.13.1 Options
- 8.14 mjpeg
  - 8.14.1 Options
- 8.15 wavpack
  - 8.15.1 Options
    - 8.15.1.1 Shared options
    - 8.15.1.2 Private options
- 9 Video Encoders
  - 9.1 Hap
    - 9.1.1 Options
  - 9.2 jpeg2000
    - 9.2.1 Options
  - 9.3 libkvazaar
    - 9.3.1 Options
  - 9.4 libopenh264
    - 9.4.1 Options
  - 9.5 libtheora
    - 9.5.1 Options

- 9.5.2 Examples
- 9.6 libvpx
  - 9.6.1 Options
- 9.7 libwebp
  - 9.7.1 Pixel Format
  - 9.7.2 Options
- 9.8 libx264, libx264rgb
  - 9.8.1 Supported Pixel Formats
  - 9.8.2 Options
- 9.9 libx265
  - 9.9.1 Options
- 9.10 libxvid
  - 9.10.1 Options
- 9.11 mpeg2
  - 9.11.1 Options
- 9.12 png
  - 9.12.1 Private options
- 9.13 ProRes
  - 9.13.1 Private Options for prores-ks
  - 9.13.2 Speed considerations
- 9.14 QSV encoders
- 9.15 snow
  - 9.15.1 Options
- 9.16 VA-API encoders
- 9.17 vc2
  - 9.17.1 Options
- 10 Subtitles Encoders
  - 10.1 dvdsub
    - 10.1.1 Options
- 11 See Also
- 12 Authors

## 1 Description# TOC

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

## 2 Codec Options# TOC

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 nonsensical 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.

`'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'`

`'ildct'`

Use interlaced DCT.

`'low_delay'`

Force low delay.

`'global_header'`

Place global headers in extradata instead of every keyframe.

`'bitexact'`

Only write platform-, build- and time-independent data. (except (I)DCT). This ensures that file and data checksums are reproducible and match between platforms. Its primary use is for regression testing.

`'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.

`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  $1 / \text{frame\_rate}$  and timestamp increments should be identically 1.

*g integer (encoding, video)*

Set the group of picture (GOP) 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. (Supported only by selected encoders, see their respective documentation sections.)

*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 between non-B-frames.

Must be an integer between -1 and 16. 0 means that B-frames are disabled. If a value of -1 is used, it will choose an automatic value depending on the encoder.

Default value is 0.

*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'`

truncated 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 an 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

‘ignore\_err’

ignore decoding errors, and continue decoding. This is useful if you want to analyze the content of a video and thus want everything to be decoded no matter what. This option will not result in a video that is pleasing to watch in case of errors.

‘careful’

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

‘compliant’

consider all spec non compliances 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 otherwise.

`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'  
'simpleauto'

Automatically pick a IDCT compatible with the simple one

'arm'  
'altivec'  
'sh4'  
'simplearm'  
'simplearmv5te'  
'simplearmv6'  
'simpleneon'  
'simplealpha'  
'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

'favor\_inter'

favor predicting from the previous frame instead of the current

*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.

*sar rational number (encoding, video)*

Set sample aspect ratio. Alias to *aspect*.

*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'

'green\_metadata'

display complexity metadata for the upcoming frame, GoP or for a given duration.

'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

`'nomc'`

skip motion compensation

`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*

*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.

*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.

`'ignorecrop'`

Ignore cropping information from sps.

`'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.

`'export_mvs'`

Export motion vectors into frame side-data (see `AV_FRAME_DATA_MOTION_VECTORS`) for codecs that support it. See also `doc/examples/export_mvs.c`.

`error integer (encoding, video)`

`qns integer (encoding, video)`

Deprecated, use mpegvideo private options instead.

`threads integer (decoding/encoding, video)`

Set the number of threads to be used, in case the selected codec implementation supports multi-threading.

Possible values:

‘auto, 0’

automatically select the number of threads to set

Default value is ‘auto’.

`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’

‘mpeg2\_aac\_low’

‘mpeg2\_aac\_he’

‘mpeg4\_sp’

‘mpeg4\_core’  
‘mpeg4\_main’  
‘mpeg4\_asp’  
‘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. Negative values behave identical to the corresponding positive ones, except that the score is normalized. Positive values exist primarily for compatibility reasons and are not so useful.

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)*

Possible values:

`'bt709'`

**BT.709**

`'bt470m'`

**BT.470 M**

`'bt470bg'`

**BT.470 BG**

`'smpte170m'`

**SMPTE 170 M**

`'smpte240m'`

**SMPTE 240 M**

`'film'`

**Film**

`'bt2020'`

**BT.2020**

`'smpte428'`

`'smpte428_1'`

**SMPTE ST 428-1**

`'smpte431'`

**SMPTE 431-2**

`'smpte432'`

**SMPTE 432-1**

`'jedec-p22'`

**JEDEC P22**

`color_trc integer (decoding/encoding, video)`

Possible values:

`'bt709'`

**BT.709**

`'gamma22'`

**BT.470 M**

`'gamma28'`

**BT.470 BG**

`'smpte170m'`

**SMPTE 170 M**

`'smpte240m'`

**SMPTE 240 M**

`'linear'`

**Linear**

`'log'`

`'log100'`

**Log**

`'log_sqrt'`

`'log316'`

**Log square root**

`'iec61966_2_4'`

`'iec61966-2-4'`



IEC 61966-2-4

'bt1361'  
'bt1361e'

BT.1361

'iec61966\_2\_1'  
'iec61966-2-1'

IEC 61966-2-1

'bt2020\_10'  
'bt2020\_10bit'

BT.2020 - 10 bit

'bt2020\_12'  
'bt2020\_12bit'

BT.2020 - 12 bit

'smpte2084'

SMPTE ST 2084

'smpte428'  
'smpte428\_1'

SMPTE ST 428-1

'arib-std-b67'

ARIB STD-B67

colorspace *integer (decoding/encoding, video)*

Possible values:

'rgb'

RGB

'bt709'

BT.709

'fcc'

FCC

`'bt470bg'`

BT.470 BG

`'smpte170m'`

SMPTE 170 M

`'smpte240m'`

SMPTE 240 M

`'ycocg'`

YCOCG

`'bt2020nc'`

`'bt2020_ncl'`

BT.2020 NCL

`'bt2020c'`

`'bt2020_cl'`

BT.2020 CL

`'smpte2085'`

SMPTE 2085

`color_range integer (decoding/encoding, video)`

If used as input parameter, it serves as a hint to the decoder, which color\_range the input has.  
Possible values:

`'tv'`

`'mpeg'`

MPEG (219\*2<sup>(n-8)</sup>)

`'pc'`

`'jpeg'`

JPEG (2<sup>n-1</sup>)

`chroma_sample_location integer (decoding/encoding, video)`

Possible values:

'left'  
'center'  
'topleft'  
'top'  
'bottomleft'  
'bottom'

`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 which multithreading methods to use.

Use of 'frame' will increase decoding delay by one frame per thread, so clients which cannot provide future frames should not use it.

Possible values:

'slice'

Decode more than one part of a single frame at once.

Multithreading using slices works only when the video was encoded with slices.

'frame'

Decode more than one frame at once.

Default value is '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.

`field_order field_order (video)`

Set/override the field order of the video. Possible values:

‘progressive’

Progressive video

‘tt’

Interlaced video, top field coded and displayed first

`'bb'`

Interlaced video, bottom field coded and displayed first

`'tb'`

Interlaced video, top coded first, bottom displayed first

`'bt'`

Interlaced video, bottom coded first, top displayed first

`skip_alpha bool (decoding, video)`

Set to 1 to disable processing alpha (transparency). This works like the `'gray'` flag in the `flags` option which skips chroma information instead of alpha. Default is 0.

`codec_whitelist list (input)`

"," separated list of allowed decoders. By default all are allowed.

`dump_separator string (input)`

Separator used to separate the fields printed on the command line about the Stream parameters. For example to separate the fields with newlines and indentation:

```
ffprobe -dump_separator "
                        " -i ~/videos/matrixbench_mpeg2.mpg
```

`max_pixels integer (decoding/encoding, video)`

Maximum number of pixels per image. This value can be used to avoid out of memory failures due to large images.

`apply_cropping bool (decoding, video)`

Enable cropping if cropping parameters are multiples of the required alignment for the left and top parameters. If the alignment is not met the cropping will be partially applied to maintain alignment. Default is 1 (enabled). Note: The required alignment depends on if `AV_CODEC_FLAG_UNALIGNED` is set and the CPU. `AV_CODEC_FLAG_UNALIGNED` cannot be changed from the command line. Also hardware decoders will not apply left/top Cropping.

## 3 Decoders# TOC

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

## 4 Video Decoders# TOC

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

### 4.1 hevc# TOC

HEVC / H.265 decoder.

Note: the `skip_loop_filter` option has effect only at level `all`.

### 4.2 rawvideo# TOC

Raw video decoder.

This decoder decodes rawvideo streams.

#### 4.2.1 Options# TOC

`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# TOC

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

### 5.1 ac3# TOC

AC-3 audio decoder.

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

#### 5.1.1 AC-3 Decoder Options# TOC

`-drc_scale value`

Dynamic Range Scale Factor. The factor to apply to dynamic range values from the AC-3 stream. This factor is applied exponentially. There are 3 notable scale factor ranges:

`drc_scale == 0`

DRC disabled. Produces full range audio.

`0 < drc_scale <= 1`

DRC enabled. Applies a fraction of the stream DRC value. Audio reproduction is between full range and full compression.

`drc_scale > 1`

DRC enabled. Applies `drc_scale` asymmetrically. Loud sounds are fully compressed. Soft sounds are enhanced.

### 5.2 flac# TOC

FLAC audio decoder.

This decoder aims to implement the complete FLAC specification from Xiph.

#### 5.2.1 FLAC Decoder options# TOC

`-use_buggy_lpc`

The lavc FLAC encoder used to produce buggy streams with high lpc values (like the default value). This option makes it possible to decode such streams correctly by using lavc's old buggy lpc logic for decoding.

## 5.3 ffwavesynth# TOC

Internal wave synthesizer.

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.

## 5.4 libcelt# TOC

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`.

## 5.5 libgsm# TOC

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.

## 5.6 libilbc# TOC

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`.

### 5.6.1 Options# TOC

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).

## 5.7 libopencore-amrnb# TOC

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.

## 5.8 libopencore-amrwb# TOC

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.

## 5.9 libopus# TOC

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`.

An FFmpeg native decoder for Opus exists, so users can decode Opus without this library.

## 6 Subtitles Decoders# TOC

### 6.1 dvbsub# TOC

#### 6.1.1 Options# TOC

`compute_clut`

-1

Compute clut if no matching CLUT is in the stream.

0

Never compute CLUT

1

Always compute CLUT and override the one provided in the stream.

`dvb_substream`

Selects the dvb substream, or all substreams if -1 which is default.

## 6.2 dvbsub# TOC

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.

### 6.2.1 Options# TOC

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.

ifo\_palette

Specify the IFO file from which the global palette is obtained. (experimental)

forced\_subs\_only

Only decode subtitle entries marked as forced. Some titles have forced and non-forced subtitles in the same track. Setting this flag to 1 will only keep the forced subtitles. Default value is 0.

## 6.3 libzvbi-teletext# TOC

Libzvbi allows libavcodec to decode DVB teletext pages and DVB teletext subtitles. Requires the presence of the libzvbi headers and library during configuration. You need to explicitly configure the build with `--enable-libzvbi`.

### 6.3.1 Options# TOC

txt\_page

List of teletext page numbers to decode. You may use the special \* string to match all pages. Pages that do not match the specified list are dropped. Default value is \*.

txt\_chop\_top

Discards the top teletext line. Default value is 1.

`txt_format`

Specifies the format of the decoded subtitles. The teletext decoder is capable of decoding the teletext pages to bitmaps or to simple text, you should use "bitmap" for teletext pages, because certain graphics and colors cannot be expressed in simple text. You might use "text" for teletext based subtitles if your application can handle simple text based subtitles. Default value is bitmap.

`txt_left`

X offset of generated bitmaps, default is 0.

`txt_top`

Y offset of generated bitmaps, default is 0.

`txt_chop_spaces`

Chops leading and trailing spaces and removes empty lines from the generated text. This option is useful for teletext based subtitles where empty spaces may be present at the start or at the end of the lines or empty lines may be present between the subtitle lines because of double-sized teletext characters. Default value is 1.

`txt_duration`

Sets the display duration of the decoded teletext pages or subtitles in milliseconds. Default value is 30000 which is 30 seconds.

`txt_transparent`

Force transparent background of the generated teletext bitmaps. Default value is 0 which means an opaque background.

`txt_opacity`

Sets the opacity (0-255) of the teletext background. If `txt_transparent` is not set, it only affects characters between a start box and an end box, typically subtitles. Default value is 0 if `txt_transparent` is set, 255 otherwise.

## 7 Encoders# TOC

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 `-encoders` of the ff\* tools will display the list of enabled encoders.

## 8 Audio Encoders# TOC

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

### 8.1 aac# TOC

Advanced Audio Coding (AAC) encoder.

This encoder is the default AAC encoder, natively implemented into FFmpeg. Its quality is on par or better than libfdk\_aac at the default bitrate of 128kbps. This encoder also implements more options, profiles and samplersates than other encoders (with only the AAC-HE profile pending to be implemented) so this encoder has become the default and is the recommended choice.

#### 8.1.1 Options# TOC

`b`

Set bit rate in bits/s. Setting this automatically activates constant bit rate (CBR) mode. If this option is unspecified it is set to 128kbps.

`q`

Set quality for variable bit rate (VBR) mode. This option is valid only using the `ffmpeg` command-line tool. For library interface users, use `global_quality`.

`cutoff`

Set cutoff frequency. If unspecified will allow the encoder to dynamically adjust the cutoff to improve clarity on low bitrates.

`aac_coder`

Set AAC encoder coding method. Possible values:

`'twoloop'`

Two loop searching (TLS) method.

This method first sets quantizers depending on band thresholds and then tries to find an optimal combination by adding or subtracting a specific value from all quantizers and adjusting some individual quantizer a little. Will tune itself based on whether `aac_is`, `aac_ms` and `aac_pns` are enabled. This is the default choice for a coder.

`'anmr'`

Average noise to mask ratio (ANMR) trellis-based solution.

This is an experimental coder which currently produces a lower quality, is more unstable and is slower than the default twoloop coder but has potential. Currently has no support for the `aac_is` or `aac_pns` options. Not currently recommended.

`'fast'`

Constant quantizer method.

This method sets a constant quantizer for all bands. This is the fastest of all the methods and has no rate control or support for `aac_is` or `aac_pns`. Not recommended.

`aac_ms`

Sets mid/side coding mode. The default value of "auto" will automatically use M/S with bands which will benefit from such coding. Can be forced for all bands using the value "enable", which is mainly useful for debugging or disabled using "disable".

`aac_is`

Sets intensity stereo coding tool usage. By default, it's enabled and will automatically toggle IS for similar pairs of stereo bands if it's beneficial. Can be disabled for debugging by setting the value to "disable".

`aac_pns`

Uses perceptual noise substitution to replace low entropy high frequency bands with imperceptible white noise during the decoding process. By default, it's enabled, but can be disabled for debugging purposes by using "disable".

`aac_tns`

Enables the use of a multitap FIR filter which spans through the high frequency bands to hide quantization noise during the encoding process and is reverted by the decoder. As well as decreasing unpleasant artifacts in the high range this also reduces the entropy in the high bands and allows for more bits to be used by the mid-low bands. By default it's enabled but can be disabled for debugging by setting the option to "disable".

`aac_ltp`

Enables the use of the long term prediction extension which increases coding efficiency in very low bandwidth situations such as encoding of voice or solo piano music by extending constant harmonic peaks in bands throughout frames. This option is implied by profile:a `aac_low` and is incompatible with `aac_pred`. Use in conjunction with `-ar` to decrease the samplerate.

`aac_pred`

Enables the use of a more traditional style of prediction where the spectral coefficients transmitted are replaced by the difference of the current coefficients minus the previous "predicted" coefficients. In theory and sometimes in practice this can improve quality for low to mid bitrate audio. This option implies the `aac_main` profile and is incompatible with `aac_ltp`.

`profile`

Sets the encoding profile, possible values:

`'aac_low'`

The default, AAC "Low-complexity" profile. Is the most compatible and produces decent quality.

`'mpeg2_aac_low'`

Equivalent to `-profile:a aac_low -aac_pns 0`. PNS was introduced with the MPEG4 specifications.

`'aac_ltp'`

Long term prediction profile, is enabled by and will enable the `aac_ltp` option. Introduced in MPEG4.

`'aac_main'`

Main-type prediction profile, is enabled by and will enable the `aac_pred` option. Introduced in MPEG2.

If this option is unspecified it is set to `'aac_low'`.

## 8.2 ac3 and ac3\_fixed# TOC

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.

## 8.2.1 AC-3 Metadata# TOC

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

### 8.2.1.1 Metadata Control Options# TOC

`-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.

### 8.2.1.2 Downmix Levels# TOC

`-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)

### 8.2.1.3 Audio Production Information# TOC

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

#### 8.2.1.4 Other Metadata Options# TOC

`-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)

## 8.2.2 Extended Bitstream Information# TOC

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.

### 8.2.2.1 Extended Bitstream Information - Part 1# TOC

`-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

`-lttrt_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

`-lttrt_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)

### 8.2.2.2 Extended Bitstream Information - Part 2# TOC

`-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

## 8.2.3 Other AC-3 Encoding Options# TOC

`-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.

*cutoff frequency*

Set lowpass cutoff frequency. If unspecified, the encoder selects a default determined by various other encoding parameters.

## 8.2.4 Floating-Point-Only AC-3 Encoding Options# TOC

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)

## 8.3 flac# TOC

FLAC (Free Lossless Audio Codec) Encoder

### 8.3.1 Options# TOC

The following options are supported by FFmpeg's flac encoder.

`compression_level`

Sets the compression level, which chooses defaults for many other options if they are not set explicitly. Valid values are from 0 to 12, 5 is the default.

`frame_size`

Sets the size of the frames in samples per channel.

`lpc_coeff_precision`

Sets the LPC coefficient precision, valid values are from 1 to 15, 15 is the default.

`lpc_type`

Sets the first stage LPC algorithm

`'none'`

LPC is not used

`'fixed'`

fixed LPC coefficients

`'levinson'`

`'cholesky'`

`lpc_passes`

Number of passes to use for Cholesky factorization during LPC analysis

`min_partition_order`

The minimum partition order

`max_partition_order`

The maximum partition order

`prediction_order_method`



`'estimation'`  
`'2level'`  
`'4level'`  
`'8level'`  
`'search'`

Bruteforce search

`'log'`  
`ch_mode`

Channel mode

`'auto'`

The mode is chosen automatically for each frame

`'indep'`

Channels are independently coded

`'left_side'`  
`'right_side'`  
`'mid_side'`  
`exact_rice_parameters`

Chooses if rice parameters are calculated exactly or approximately. if set to 1 then they are chosen exactly, which slows the code down slightly and improves compression slightly.

`multi_dim_quant`

Multi Dimensional Quantization. If set to 1 then a 2nd stage LPC algorithm is applied after the first stage to finetune the coefficients. This is quite slow and slightly improves compression.

## 8.4 opus# TOC

Opus encoder.

This is a native FFmpeg encoder for the Opus format. Currently its in development and only implements the CELT part of the codec. Its quality is usually worse and at best is equal to the libopus encoder.

### 8.4.1 Options# TOC

b

Set bit rate in bits/s. If unspecified it uses the number of channels and the layout to make a good guess.

`opus_delay`

Sets the maximum delay in milliseconds. Lower delays than 20ms will very quickly decrease quality.

## 8.5 libfdk\_aac# TOC

libfdk-aac AAC (Advanced Audio Coding) encoder wrapper.

The libfdk-aac library is based on the Fraunhofer FDK AAC code from the Android project.

Requires the presence of the libfdk-aac headers and library during configuration. You need to explicitly configure the build with `--enable-libfdk-aac`. The library is also incompatible with GPL, so if you allow the use of GPL, you should configure with `--enable-gpl --enable-nonfree --enable-libfdk-aac`.

This encoder is considered to produce output on par or worse at 128kbps to the the native FFmpeg AAC encoder but can often produce better sounding audio at identical or lower bitrates and has support for the AAC-HE profiles.

VBR encoding, enabled through the `vbr` or `flags +qscale` options, is experimental and only works with some combinations of parameters.

Support for encoding 7.1 audio is only available with libfdk-aac 0.1.3 or higher.

For more information see the fdk-aac project at <http://sourceforge.net/p/opencore-amr/fdk-aac/>.

### 8.5.1 Options# TOC

The following options are mapped on the shared FFmpeg codec options.

`b`

Set bit rate in bits/s. If the bitrate is not explicitly specified, it is automatically set to a suitable value depending on the selected profile.

In case VBR mode is enabled the option is ignored.

`ar`

Set audio sampling rate (in Hz).

`channels`

Set the number of audio channels.

`flags +qscale`

Enable fixed quality, VBR (Variable Bit Rate) mode. Note that VBR is implicitly enabled when the vbr value is positive.

#### cutoff

Set cutoff frequency. If not specified (or explicitly set to 0) it will use a value automatically computed by the library. Default value is 0.

#### profile

Set audio profile.

The following profiles are recognized:

‘aac\_low’

Low Complexity AAC (LC)

‘aac\_he’

High Efficiency AAC (HE-AAC)

‘aac\_he\_v2’

High Efficiency AAC version 2 (HE-AACv2)

‘aac\_ld’

Low Delay AAC (LD)

‘aac\_eld’

Enhanced Low Delay AAC (ELD)

If not specified it is set to ‘aac\_low’.

The following are private options of the libfdk\_aac encoder.

#### afterburner

Enable afterburner feature if set to 1, disabled if set to 0. This improves the quality but also the required processing power.

Default value is 1.

#### eld\_sbr

Enable SBR (Spectral Band Replication) for ELD if set to 1, disabled if set to 0.

Default value is 0.

#### signaling

Set SBR/PS signaling style.

It can assume one of the following values:

‘default’

choose signaling implicitly (explicit hierarchical by default, implicit if global header is disabled)

‘implicit’

implicit backwards compatible signaling

‘explicit\_sbr’

explicit SBR, implicit PS signaling

‘explicit\_hierarchical’

explicit hierarchical signaling

Default value is ‘default’.

#### latm

Output LATM/LOAS encapsulated data if set to 1, disabled if set to 0.

Default value is 0.

#### header\_period

Set StreamMuxConfig and PCE repetition period (in frames) for sending in-band configuration buffers within LATM/LOAS transport layer.

Must be a 16-bits non-negative integer.

Default value is 0.

#### vbr

Set VBR mode, from 1 to 5. 1 is lowest quality (though still pretty good) and 5 is highest quality. A value of 0 will disable VBR, and CBR (Constant Bit Rate) is enabled.

Currently only the ‘aac\_low’ profile supports VBR encoding.

VBR modes 1-5 correspond to roughly the following average bit rates:

‘1’

32 kbps/channel

‘2’

40 kbps/channel

‘3’

48-56 kbps/channel

‘4’

64 kbps/channel

‘5’

about 80-96 kbps/channel

Default value is 0.

### 8.5.2 Examples# TOC

- Use `ffmpeg` to convert an audio file to VBR AAC in an M4A (MP4) container:

```
ffmpeg -i input.wav -codec:a libfdk_aac -vbr 3 output.m4a
```

- Use `ffmpeg` to convert an audio file to CBR 64k kbps AAC, using the High-Efficiency AAC profile:

```
ffmpeg -i input.wav -c:a libfdk_aac -profile:a aac_he -b:a 64k output.m4a
```

## 8.6 libmp3lame# TOC

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`.

See `libshine` for a fixed-point MP3 encoder, although with a lower quality.

### 8.6.1 Options# TOC

The following options are supported by the `libmp3lame` wrapper. The `lame`-equivalent of the options are listed in parentheses.

`b (-b)`

Set bitrate expressed in bits/s for CBR or ABR. LAME `bitrate` is expressed in kilobits/s.

`q (-V)`

Set constant quality setting for VBR. This option is valid only using the `ffmpeg` command-line tool. For library interface users, use `global_quality`.

`compression_level (-q)`

Set algorithm quality. Valid arguments are integers in the 0-9 range, with 0 meaning highest quality but slowest, and 9 meaning fastest while producing the worst quality.

`cutoff (--lowpass)`

Set lowpass cutoff frequency. If unspecified, the encoder dynamically adjusts the cutoff.

`reservoir`

Enable use of bit reservoir when set to 1. Default value is 1. LAME has this enabled by default, but can be overridden by use `--nores` option.

`joint_stereo (-m j)`

Enable the encoder to use (on a frame by frame basis) either L/R stereo or mid/side stereo. Default value is 1.

`abr (--abr)`

Enable the encoder to use ABR when set to 1. The `lame --abr` sets the target bitrate, while this options only tells FFmpeg to use ABR still relies on `b` to set bitrate.

## 8.7 libopencore-amrnb# TOC

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.

### 8.7.1 Options# TOC

`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).

## 8.8 libopus# TOC

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`.

### 8.8.1 Option Mapping# TOC

Most libopus options are modelled 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 `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 is 20ms.

`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).

`mapping_family (mapping_family)`

Set channel mapping family to be used by the encoder. The default value of -1 uses mapping family 0 for mono and stereo inputs, and mapping family 1 otherwise. The default also disables the surround masking and LFE bandwidth optimizations in libopus, and requires that the input contains 8 channels or fewer.



Other values include 0 for mono and stereo, 1 for surround sound with masking and LFE bandwidth optimizations, and 255 for independent streams with an unspecified channel layout.

## 8.9 libshine# TOC

Shine Fixed-Point MP3 encoder wrapper.

Shine is a fixed-point MP3 encoder. It has a far better performance on platforms without an FPU, e.g. armel CPUs, and some phones and tablets. However, as it is more targeted on performance than quality, it is not on par with LAME and other production-grade encoders quality-wise. Also, according to the project's homepage, this encoder may not be free of bugs as the code was written a long time ago and the project was dead for at least 5 years.

This encoder only supports stereo and mono input. This is also CBR-only.

The original project (last updated in early 2007) is at <http://sourceforge.net/projects/libshine-fxp/>. We only support the updated fork by the Savonet/Liquidsoap project at <https://github.com/savonet/shine>.

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

See also libmp3lame.

### 8.9.1 Options# TOC

The following options are supported by the libshine wrapper. The `shineenc`-equivalent of the options are listed in parentheses.

`b` (`-b`)

Set bitrate expressed in bits/s for CBR. `shineenc -b` option is expressed in kilobits/s.

## 8.10 libtwolame# TOC

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`.

### 8.10.1 Options# TOC

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 expressed in bits/s for CBR. `twolame b` option is expressed in kilobits/s. 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. The higher the value, the better the quality. This option is valid only using the `ffmpeg` command-line tool. For library interface users, use `global_quality`.

`mode (--mode)`

Set the mode of the resulting audio. 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).

## 8.11 libvo-amrwbenc# TOC

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.

### 8.11.1 Options# TOC

`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).

## 8.12 libvorbis# TOC

libvorbis encoder wrapper.

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

## 8.12.1 Options# TOC

The following options are supported by the libvorbis wrapper. The `oggenc`-equivalent of the options are listed in parentheses.

To get a more accurate and extensive documentation of the libvorbis options, consult the libvorbisenc's and `oggenc`'s documentations. See <http://xiph.org/vorbis/>, <http://wiki.xiph.org/Vorbis-tools>, and `oggenc(1)`.

`b` (`-b`)

Set bitrate expressed in bits/s for ABR. `oggenc -b` is expressed in kilobits/s.

`q` (`-q`)

Set constant quality setting for VBR. The value should be a float number in the range of -1.0 to 10.0. The higher the value, the better the quality. The default value is '3.0'.

This option is valid only using the `ffmpeg` command-line tool. For library interface users, use `global_quality`.

`cutoff` (`--advanced-encode-option lowpass_frequency=N`)

Set cutoff bandwidth in Hz, a value of 0 disables cutoff. `oggenc`'s related option is expressed in kHz. The default value is '0' (cutoff disabled).

`minrate` (`-m`)

Set minimum bitrate expressed in bits/s. `oggenc -m` is expressed in kilobits/s.

`maxrate` (`-M`)

Set maximum bitrate expressed in bits/s. `oggenc -M` is expressed in kilobits/s. This only has effect on ABR mode.

`iblock` (`--advanced-encode-option impulse_noisetune=N`)

Set noise floor bias for impulse blocks. The value is a float number from -15.0 to 0.0. A negative bias instructs the encoder to pay special attention to the crispness of transients in the encoded audio. The tradeoff for better transient response is a higher bitrate.

## 8.13 libwavpack# TOC

A wrapper providing WavPack encoding through libwavpack.

Only lossless mode using 32-bit integer samples is supported currently.

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

Note that a libavcodec-native encoder for the WavPack codec exists so users can encode audios with this codec without using this encoder. See `wavpackenc`.

### 8.13.1 Options# TOC

`wavpack` command line utility's corresponding options are listed in parentheses, if any.

`frame_size` (`--blocksize`)

Default is 32768.

`compression_level`

Set speed vs. compression tradeoff. Acceptable arguments are listed below:

'0' (`-f`)

Fast mode.

'1'

Normal (default) settings.

'2' (`-h`)

High quality.

'3' (`-hh`)

Very high quality.

'4-8' (`-hh -xEXTRAPROC`)

Same as '3', but with extra processing enabled.

'4' is the same as `-x2` and '8' is the same as `-x6`.

## 8.14 mjpeg# TOC

Motion JPEG encoder.

### 8.14.1 Options# TOC

`huffman`

Set the huffman encoding strategy. Possible values:

`'default'`

Use the default huffman tables. This is the default strategy.

`'optimal'`

Compute and use optimal huffman tables.

## 8.15 wavpack# TOC

WavPack lossless audio encoder.

This is a libavcodec-native WavPack encoder. There is also an encoder based on libwavpack, but there is virtually no reason to use that encoder.

See also libwavpack.

### 8.15.1 Options# TOC

The equivalent options for `wavpack` command line utility are listed in parentheses.

#### 8.15.1.1 Shared options# TOC

The following shared options are effective for this encoder. Only special notes about this particular encoder will be documented here. For the general meaning of the options, see the Codec Options chapter.

`frame_size (--blocksize)`

For this encoder, the range for this option is between 128 and 131072. Default is automatically decided based on sample rate and number of channel.

For the complete formula of calculating default, see `libavcodec/wavpackenc.c`.

`compression_level (-f, -h, -hh, and -x)`

This option's syntax is consistent with libwavpack's.

#### 8.15.1.2 Private options# TOC

`joint_stereo (-j)`

Set whether to enable joint stereo. Valid values are:

`'on (1)'`

Force mid/side audio encoding.

‘off (0)’

Force left/right audio encoding.

‘auto’

Let the encoder decide automatically.

`optimize_mono`

Set whether to enable optimization for mono. This option is only effective for non-mono streams.  
Available values:

‘on’

enabled

‘off’

disabled

## 9 Video Encoders# TOC

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

### 9.1 Hap# TOC

Vidvox Hap video encoder.

#### 9.1.1 Options# TOC

`format integer`

Specifies the Hap format to encode.

hap

hap\_alpha

hap\_q

Default value is hap.

`chunks integer`

Specifies the number of chunks to split frames into, between 1 and 64. This permits multithreaded decoding of large frames, potentially at the cost of data-rate. The encoder may modify this value to divide frames evenly.

Default value is *1*.

`compressor integer`

Specifies the second-stage compressor to use. If set to *none*, chunks will be limited to 1, as chunked uncompressed frames offer no benefit.

*none*  
*snappy*

Default value is *snappy*.

## 9.2 jpeg2000# TOC

The native jpeg 2000 encoder is lossy by default, the `-q:v` option can be used to set the encoding quality. Lossless encoding can be selected with `-pred 1`.

### 9.2.1 Options# TOC

`format`

Can be set to either *j2k* or *jp2* (the default) that makes it possible to store non-rgb *pix\_fmts*.

## 9.3 libkvazaar# TOC

Kvazaar H.265/HEVC encoder.

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

### 9.3.1 Options# TOC

`b`

Set target video bitrate in bit/s and enable rate control.

`kvazaar-params`

Set kvazaar parameters as a list of *name=value* pairs separated by commas (,). See kvazaar documentation for a list of options.

## 9.4 libopenh264# TOC

Cisco libopenh264 H.264/MPEG-4 AVC encoder wrapper.

This encoder requires the presence of the libopenh264 headers and library during configuration. You need to explicitly configure the build with `--enable-libopenh264`. The library is detected using `pkg-config`.



For more information about the library see <http://www.openh264.org>.

### 9.4.1 Options# TOC

The following FFmpeg global options affect the configurations of the libopenh264 encoder.

`b`

Set the bitrate (as a number of bits per second).

`g`

Set the GOP size.

`maxrate`

Set the max bitrate (as a number of bits per second).

`flags +global_header`

Set global header in the bitstream.

`slices`

Set the number of slices, used in parallelized encoding. Default value is 0. This is only used when `slice_mode` is set to 'fixed'.

`slice_mode`

Set slice mode. Can assume one of the following possible values:

'fixed'

a fixed number of slices

'rowmb'

one slice per row of macroblocks

'auto'

automatic number of slices according to number of threads

'dyn'

dynamic slicing

Default value is 'auto'.

`loopfilter`

Enable loop filter, if set to 1 (automatically enabled). To disable set a value of 0.

`profile`

Set profile restrictions. If set to the value of 'main' enable CABAC (set the `SEncParamExt.iEntropyCodingModeFlag` flag to 1).

`max_nal_size`

Set maximum NAL size in bytes.

`allow_skip_frames`

Allow skipping frames to hit the target bitrate if set to 1.

## 9.5 libtheora# TOC

libtheora Theora encoder wrapper.

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

For more information about the libtheora project see <http://www.theora.org/>.

### 9.5.1 Options# TOC

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 in bit/s for CBR (Constant Bit Rate) mode. In case VBR (Variable Bit Rate) mode is enabled this option is ignored.

`flags`

Used to enable constant quality mode (VBR) encoding through the `qscale` flag, and to enable the `pass1` and `pass2` modes.

`g`

Set the GOP size.

`global_quality`

Set the global quality as an integer in lambda units.

Only relevant when VBR mode is enabled with `flags +qscale`. The value is converted to QP units by dividing it by `FF_QP2LAMBDA`, clipped in the [0 - 10] range, and then multiplied by 6.3 to get a value in the native libtheora range [0-63]. A higher value corresponds to a higher quality.

q

Enable VBR mode when set to a non-negative value, and set constant quality value as a double floating point value in QP units.

The value is clipped in the [0-10] range, and then multiplied by 6.3 to get a value in the native libtheora range [0-63].

This option is valid only using the `ffmpeg` command-line tool. For library interface users, use `global_quality`.

## 9.5.2 Examples# TOC

- Set maximum constant quality (VBR) encoding with `ffmpeg`:

```
ffmpeg -i INPUT -codec:v libtheora -q:v 10 OUTPUT.ogg
```

- Use `ffmpeg` to convert a CBR 1000 kbps Theora video stream:

```
ffmpeg -i INPUT -codec:v libtheora -b:v 1000k OUTPUT.ogg
```

## 9.6 libvpx# TOC

VP8/VP9 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`.

### 9.6.1 Options# TOC

The following options are supported by the libvpx wrapper. The `vpxenc`-equivalent options or values are listed in parentheses for easy migration.

To reduce the duplication of documentation, only the private options and some others requiring special attention are documented here. For the documentation of the undocumented generic options, see the Codec Options chapter.

To get more documentation of the libvpx options, invoke the command `ffmpeg -h encoder=libvpx`, `ffmpeg -h encoder=libvpx-vp9` or `vpxenc --help`. Further information is available in the libvpx API documentation.

`b (target-bitrate)`

Set bitrate in bits/s. Note that FFmpeg's `b` option is expressed in bits/s, while `vpxenc`'s `target-bitrate` is in kilobits/s.

`g (kf-max-dist)`

`keyint_min (kf-min-dist)`

`qmin (min-q)`

`qmax (max-q)`

`bufsize (buf-sz, buf-optimal-sz)`

Set ratecontrol buffer size (in bits). Note `vpxenc`'s options are specified in milliseconds, the `libvpx` wrapper converts this value as follows: `buf-sz = bufsize * 1000 / bitrate`,  
`buf-optimal-sz = bufsize * 1000 / bitrate * 5 / 6`.

`rc_init_occupancy (buf-initial-sz)`

Set number of bits which should be loaded into the `rc` buffer before decoding starts. Note `vpxenc`'s option is specified in milliseconds, the `libvpx` wrapper converts this value as follows:  
`rc_init_occupancy * 1000 / bitrate`.

`undershoot-pct`

Set datarate undershoot (min) percentage of the target bitrate.

`overshoot-pct`

Set datarate overshoot (max) percentage of the target bitrate.

`skip_threshold (drop-frame)`

`qcomp (bias-pct)`

`maxrate (maxsection-pct)`

Set GOP max bitrate in bits/s. Note `vpxenc`'s option is specified as a percentage of the target bitrate, the `libvpx` wrapper converts this value as follows: `(maxrate * 100 / bitrate)`.

`minrate (minsection-pct)`

Set GOP min bitrate in bits/s. Note `vpxenc`'s option is specified as a percentage of the target bitrate, the `libvpx` wrapper converts this value as follows: `(minrate * 100 / bitrate)`.

`minrate, maxrate, b end-usage=cbr`

`(minrate == maxrate == bitrate).`

`crf (end-usage=cq, cq-level)`

`tune (tune)`

```
'psnr (psnr)'  
'ssim (ssim)'  
quality, deadline (deadline)  
'best'
```

Use best quality deadline. Poorly named and quite slow, this option should be avoided as it may give worse quality output than good.

```
'good'
```

Use good quality deadline. This is a good trade-off between speed and quality when used with the `cpu-used` option.

```
'realtime'
```

Use realtime quality deadline.

```
speed, cpu-used (cpu-used)
```

Set quality/speed ratio modifier. Higher values speed up the encode at the cost of quality.

```
nr (noise-sensitivity)  
static-thresh
```

Set a change threshold on blocks below which they will be skipped by the encoder.

```
slices (token-parts)
```

Note that FFmpeg's `slices` option gives the total number of partitions, while `vpxenc`'s `token-parts` is given as  $\log_2(\text{partitions})$ .

```
max-intra-rate
```

Set maximum I-frame bitrate as a percentage of the target bitrate. A value of 0 means unlimited.

```
force_key_frames
```

```
VPX_EFLAG_FORCE_KF
```

```
Alternate reference frame related  
auto-alt-ref
```

Enable use of alternate reference frames (2-pass only).

```
arnr-max-frames
```

Set altref noise reduction max frame count.

`arnr-type`

Set altref noise reduction filter type: backward, forward, centered.

`arnr-strength`

Set altref noise reduction filter strength.

`rc-lookahead, lag-in-frames` (*lag-in-frames*)

Set number of frames to look ahead for frametype and ratecontrol.

`error-resilient`

Enable error resiliency features.

`VP9-specific options`

`lossless`

Enable lossless mode.

`tile-columns`

Set number of tile columns to use. Note this is given as `log2(tile_columns)`. For example, 8 tile columns would be requested by setting the `tile-columns` option to 3.

`tile-rows`

Set number of tile rows to use. Note this is given as `log2(tile_rows)`. For example, 4 tile rows would be requested by setting the `tile-rows` option to 2.

`frame-parallel`

Enable frame parallel decodability features.

`aq-mode`

Set adaptive quantization mode (0: off (default), 1: variance 2: complexity, 3: cyclic refresh, 4: equator360).

`colorspace` *color-space*

Set input color space. The VP9 bitstream supports signaling the following colorspaces:

`'rgb'` *sRGB*

`'bt709'` *bt709*

`'unspecified'` *unknown*

`'bt470bg'` *bt601*

```
`smpte170m'  smpte170
`smpte240m'  smpte240
`bt2020_ncl' bt2020
row-mt boolean
```

Enable row based multi-threading.

```
tune-content
```

Set content type: default (0), screen (1), film (2).

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

## 9.7 libwebp# TOC

libwebp WebP Image encoder wrapper

libwebp is Google's official encoder for WebP images. It can encode in either lossy or lossless mode. Lossy images are essentially a wrapper around a VP8 frame. Lossless images are a separate codec developed by Google.

### 9.7.1 Pixel Format# TOC

Currently, libwebp only supports YUV420 for lossy and RGB for lossless due to limitations of the format and libwebp. Alpha is supported for either mode. Because of API limitations, if RGB is passed in when encoding lossy or YUV is passed in for encoding lossless, the pixel format will automatically be converted using functions from libwebp. This is not ideal and is done only for convenience.

### 9.7.2 Options# TOC

```
-lossless boolean
```

Enables/Disables use of lossless mode. Default is 0.

```
-compression_level integer
```

For lossy, this is a quality/speed tradeoff. Higher values give better quality for a given size at the cost of increased encoding time. For lossless, this is a size/speed tradeoff. Higher values give smaller size at the cost of increased encoding time. More specifically, it controls the number of extra algorithms and compression tools used, and varies the combination of these tools. This maps to the *method* option in libwebp. The valid range is 0 to 6. Default is 4.

```
-qscale float
```

For lossy encoding, this controls image quality, 0 to 100. For lossless encoding, this controls the effort and time spent at compressing more. The default value is 75. Note that for usage via libavcodec, this option is called *global\_quality* and must be multiplied by *FF\_QP2LAMBDA*.

`-preset type`

Configuration preset. This does some automatic settings based on the general type of the image.

`none`

Do not use a preset.

`default`

Use the encoder default.

`picture`

Digital picture, like portrait, inner shot

`photo`

Outdoor photograph, with natural lighting

`drawing`

Hand or line drawing, with high-contrast details

`icon`

Small-sized colorful images

`text`

Text-like

## 9.8 libx264, libx264rgb# TOC

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

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

libx264 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).

Many libx264 encoder options are mapped to FFmpeg global codec options, while unique encoder options are provided through private options. Additionally the `x264opts` and `x264-params` private options allows one to pass a list of key=value tuples as accepted by the libx264 `x264_param_parse` function.



The x264 project website is at <http://www.videolan.org/developers/x264.html>.

The libx264rgb encoder is the same as libx264, except it accepts packed RGB pixel formats as input instead of YUV.

### 9.8.1 Supported Pixel Formats# TOC

x264 supports 8- to 10-bit color spaces. The exact bit depth is controlled at x264's configure time. FFmpeg only supports one bit depth in one particular build. In other words, it is not possible to build one FFmpeg with multiple versions of x264 with different bit depths.

### 9.8.2 Options# TOC

The following options are supported by the libx264 wrapper. The x264-equivalent options or values are listed in parentheses for easy migration.

To reduce the duplication of documentation, only the private options and some others requiring special attention are documented here. For the documentation of the undocumented generic options, see the Codec Options chapter.

To get a more accurate and extensive documentation of the libx264 options, invoke the command `x264 --fullhelp` or consult the libx264 documentation.

`b` (*bitrate*)

Set bitrate in bits/s. Note that FFmpeg's `b` option is expressed in bits/s, while x264's `bitrate` is in kilobits/s.

`bf` (*bframes*)

`g` (*keyint*)

`qmin` (*qpmin*)

Minimum quantizer scale.

`qmax` (*qpmax*)

Maximum quantizer scale.

`qdiff` (*qpstep*)

Maximum difference between quantizer scales.

`qblur` (*qblur*)

Quantizer curve blur

`qcomp` (*qcomp*)

Quantizer curve compression factor

`refs` (*ref*)

Number of reference frames each P-frame can use. The range is from *0-16*.

`sc_threshold` (*scenecut*)

Sets the threshold for the scene change detection.

`trellis` (*trellis*)

Performs Trellis quantization to increase efficiency. Enabled by default.

`nr` (*nr*)

`me_range` (*merange*)

Maximum range of the motion search in pixels.

`me_method` (*me*)

Set motion estimation method. Possible values in the decreasing order of speed:

`'dia` (*dia*)'

`'epzs` (*dia*)'

Diamond search with radius 1 (fastest). `'epzs'` is an alias for `'dia'`.

`'hex` (*hex*)'

Hexagonal search with radius 2.

`'umh` (*umh*)'

Uneven multi-hexagon search.

`'esa` (*esa*)'

Exhaustive search.

`'tesa` (*tesa*)'

Hadamard exhaustive search (slowest).

`forced-idr`

Normally, when forcing a I-frame type, the encoder can select any type of I-frame. This option forces it to choose an IDR-frame.

`subq (subme)`

Sub-pixel motion estimation method.

`b_strategy (b-adapt)`

Adaptive B-frame placement decision algorithm. Use only on first-pass.

`keyint_min (min-keyint)`

Minimum GOP size.

`coder`

Set entropy encoder. Possible values:

`'ac'`

Enable CABAC.

`'vlc'`

Enable CAVLC and disable CABAC. It generates the same effect as x264's `--no-cabac` option.

`cmp`

Set full pixel motion estimation comparison algorithm. Possible values:

`'chroma'`

Enable chroma in motion estimation.

`'sad'`

Ignore chroma in motion estimation. It generates the same effect as x264's `--no-chroma-me` option.

`threads (threads)`

Number of encoding threads.

`thread_type`

Set multithreading technique. Possible values:

`'slice'`

Slice-based multithreading. It generates the same effect as x264's `--sliced-threads` option.

`'frame'`

Frame-based multithreading.

`flags`

Set encoding flags. It can be used to disable closed GOP and enable open GOP by setting it to `-cgop`. The result is similar to the behavior of x264's `--open-gop` option.

`rc_init_occupancy (vbv-init)`  
`preset (preset)`

Set the encoding preset.

`tune (tune)`

Set tuning of the encoding params.

`profile (profile)`

Set profile restrictions.

`fastfirstpass`

Enable fast settings when encoding first pass, when set to 1. When set to 0, it has the same effect of x264's `--slow-firstpass` option.

`crf (crf)`

Set the quality for constant quality mode.

`crf_max (crf-max)`

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

`qp (qp)`

Set constant quantization rate control method parameter.

`aq-mode (aq-mode)`

Set AQ method. Possible values:

`'none (0)'`

Disabled.

`'variance (1)'`

Variance AQ (complexity mask).

`'autovariance (2)'`

Auto-variance AQ (experimental).

`aq-strength (aq-strength)`

Set AQ strength, reduce blocking and blurring in flat and textured areas.

`psy`

Use psychovisual optimizations when set to 1. When set to 0, it has the same effect as x264's `--no-psy` option.

`psy-rd (psy-rd)`

Set strength of psychovisual optimization, in *psy-rd:psy-trellis* format.

`rc-lookahead (rc-lookahead)`

Set number of frames to look ahead for frametype and ratecontrol.

`weightb`

Enable weighted prediction for B-frames when set to 1. When set to 0, it has the same effect as x264's `--no-weightb` option.

`weightp (weightp)`

Set weighted prediction method for P-frames. Possible values:

`'none (0)'`

Disabled

`'simple (1)'`

Enable only weighted refs

`'smart (2)'`

Enable both weighted refs and duplicates

`ssim` (*ssim*)

Enable calculation and printing SSIM stats after the encoding.

`intra-refresh` (*intra-refresh*)

Enable the use of Periodic Intra Refresh instead of IDR frames when set to 1.

`avcintra-class` (*class*)

Configure the encoder to generate AVC-Intra. Valid values are 50,100 and 200

`bluray-compat` (*bluray-compat*)

Configure the encoder to be compatible with the bluray standard. It is a shorthand for setting "bluray-compat=1 force-cfr=1".

`b-bias` (*b-bias*)

Set the influence on how often B-frames are used.

`b-pyramid` (*b-pyramid*)

Set method for keeping of some B-frames as references. Possible values:

`'none'` (*none*)

Disabled.

`'strict'` (*strict*)

Strictly hierarchical pyramid.

`'normal'` (*normal*)

Non-strict (not Blu-ray compatible).

`mixed-refs`

Enable the use of one reference per partition, as opposed to one reference per macroblock when set to 1. When set to 0, it has the same effect as x264's `--no-mixed-refs` option.

`8x8dct`

Enable adaptive spatial transform (high profile 8x8 transform) when set to 1. When set to 0, it has the same effect as x264's `--no-8x8dct` option.

`fast-pskip`

Enable early SKIP detection on P-frames when set to 1. When set to 0, it has the same effect as x264's `--no-fast-pskip` option.

`aud` (*aud*)

Enable use of access unit delimiters when set to 1.

`mbtree`

Enable use macroblock tree ratecontrol when set to 1. When set to 0, it has the same effect as x264's `--no-mbtree` option.

`deblock` (*deblock*)

Set loop filter parameters, in *alpha:beta* form.

`cplxblur` (*cplxblur*)

Set fluctuations reduction in QP (before curve compression).

`partitions` (*partitions*)

Set partitions to consider as a comma-separated list of. Possible values in the list:

`'p8x8'`

8x8 P-frame partition.

`'p4x4'`

4x4 P-frame partition.

`'b8x8'`

4x4 B-frame partition.

`'i8x8'`

8x8 I-frame partition.

`'i4x4'`

4x4 I-frame partition. (Enabling `'p4x4'` requires `'p8x8'` to be enabled. Enabling `'i8x8'` requires adaptive spatial transform (`8x8dct` option) to be enabled.)

`'none` (*none*)'

Do not consider any partitions.

`'all (all)'`

Consider every partition.

`direct-pred (direct)`

Set direct MV prediction mode. Possible values:

`'none (none)'`

Disable MV prediction.

`'spatial (spatial)'`

Enable spatial predicting.

`'temporal (temporal)'`

Enable temporal predicting.

`'auto (auto)'`

Automatically decided.

`slice-max-size (slice-max-size)`

Set the limit of the size of each slice in bytes. If not specified but RTP payload size (ps) is specified, that is used.

`stats (stats)`

Set the file name for multi-pass stats.

`nal-hrd (nal-hrd)`

Set signal HRD information (requires `vbv-bufsize` to be set). Possible values:

`'none (none)'`

Disable HRD information signaling.

`'vbr (vbr)'`

Variable bit rate.

`'cbr (cbr)'`

Constant bit rate (not allowed in MP4 container).



`x264opts` (N.A.)

Set any x264 option, see `x264 --fullhelp` for a list.

Argument 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 -c:v libx264 -x264opts keyint=123:min-keyint=20 -an out.mkv
```

`a53cc` *boolean*

Import closed captions (which must be ATSC compatible format) into output. Only the `mpeg2` and `h264` decoders provide these. Default is 1 (on).

`x264-params` (N.A.)

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

This option is functionally the same as the `x264opts`, but is duplicated for compatibility with the Libav fork.

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

```
ffmpeg -i INPUT -c:v libx264 -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 OUTPUT
```

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

## 9.9 libx265# TOC

`x265` H.265/HEVC encoder wrapper.

This encoder requires the presence of the `libx265` headers and library during configuration. You need to explicitly configure the build with `--enable-libx265`.

### 9.9.1 Options# TOC

`preset`

Set the x265 preset.

`tune`

Set the x265 tune parameter.

`forced-idr`

Normally, when forcing a I-frame type, the encoder can select any type of I-frame. This option forces it to choose an IDR-frame.

`x265-params`

Set x265 options using a list of *key=value* couples separated by ":". See `x265 --help` for a list of options.

For example to specify libx265 encoding options with `-x265-params`:

```
ffmpeg -i input -c:v libx265 -x265-params crf=26:psy-rd=1 output.mp4
```

## 9.10 libxvid# TOC

Xvid MPEG-4 Part 2 encoder wrapper.

This encoder requires the presence of the libxvidcore headers and library during configuration. You need to explicitly configure the build with `--enable-libxvid --enable-gpl`.

The native `mpeg4` encoder supports the MPEG-4 Part 2 format, so users can encode to this format without this library.

### 9.10.1 Options# TOC

The following options are supported by the libxvid wrapper. Some of the following options are listed but are not documented, and correspond to shared codec options. See the Codec Options chapter for their documentation. The other shared options which are not listed have no effect for the libxvid encoder.

`b`  
`g`  
`qmin`  
`qmax`  
`mpeg_quant`  
`threads`  
`bf`  
`b_qfactor`  
`b_qoffset`  
`flags`

Set specific encoding flags. Possible values:

`'mv4'`

Use four motion vector by macroblock.

`'aic'`

Enable high quality AC prediction.

`'gray'`

Only encode grayscale.

`'gmc'`

Enable the use of global motion compensation (GMC).

`'qpel'`

Enable quarter-pixel motion compensation.

`'cgop'`

Enable closed GOP.

`'global_header'`

Place global headers in extradata instead of every keyframe.

`trellis`  
`me_method`

Set motion estimation method. Possible values in decreasing order of speed and increasing order of quality:

`'zero'`

Use no motion estimation (default).

`'phods'`

`'x1'`

`'log'`

Enable advanced diamond zonal search for 16x16 blocks and half-pixel refinement for 16x16 blocks. `'x1'` and `'log'` are aliases for `'phods'`.

`'epzs'`

Enable all of the things described above, plus advanced diamond zonal search for 8x8 blocks, half-pixel refinement for 8x8 blocks, and motion estimation on chroma planes.

`'full'`

Enable all of the things described above, plus extended 16x16 and 8x8 blocks search.

`mbd`

Set macroblock decision algorithm. Possible values in the increasing order of quality:

`'simple'`

Use macroblock comparing function algorithm (default).

`'bits'`

Enable rate distortion-based half pixel and quarter pixel refinement for 16x16 blocks.

`'rd'`

Enable all of the things described above, plus rate distortion-based half pixel and quarter pixel refinement for 8x8 blocks, and rate distortion-based search using square pattern.

`lumi_aq`

Enable lumi masking adaptive quantization when set to 1. Default is 0 (disabled).

`variance_aq`

Enable variance adaptive quantization when set to 1. Default is 0 (disabled).

When combined with `lumi_aq`, the resulting quality will not be better than any of the two specified individually. In other words, the resulting quality will be the worse one of the two effects.

`ssim`

Set structural similarity (SSIM) displaying method. Possible values:

`'off'`

Disable displaying of SSIM information.

`'avg'`

Output average SSIM at the end of encoding to stdout. The format of showing the average SSIM is:

Average SSIM: %f

For users who are not familiar with C, %f means a float number, or a decimal (e.g. 0.939232).

`'frame'`

Output both per-frame SSIM data during encoding and average SSIM at the end of encoding to stdout. The format of per-frame information is:

```
SSIM: avg: %1.3f min: %1.3f max: %1.3f
```

For users who are not familiar with C, %1.3f means a float number rounded to 3 digits after the dot (e.g. 0.932).

`ssim_acc`

Set SSIM accuracy. Valid options are integers within the range of 0-4, while 0 gives the most accurate result and 4 computes the fastest.

## 9.11 mpeg2# TOC

MPEG-2 video encoder.

### 9.11.1 Options# TOC

`seq_disp_ext integer`

Specifies if the encoder should write a `sequence_display_extension` to the output.

`-1`

`auto`

Decide automatically to write it or not (this is the default) by checking if the data to be written is different from the default or unspecified values.

`0`

`never`

Never write it.

`1`

`always`

Always write it.

## 9.12 png# TOC

PNG image encoder.

### 9.12.1 Private options# TOC

`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

## 9.13 ProRes# TOC

Apple ProRes encoder.

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

### 9.13.1 Private Options for prores-ks# TOC

`profile integer`

Select the ProRes profile to encode

`'proxy'`  
`'lt'`  
`'standard'`  
`'hq'`  
`'4444'`  
`'4444xq'`

`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.

### 9.13.2 Speed considerations# TOC

In the default mode of operation the encoder has to honor frame constraints (i.e. not produce 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.

### 9.14 QSV encoders# TOC

The family of Intel QuickSync Video encoders (MPEG-2, H.264 and HEVC)

The ratecontrol method is selected as follows:

- When `global_quality` is specified, a quality-based mode is used. Specifically this means either
  - - *CQP* - constant quantizer scale, when the `qscale` codec flag is also set (the `-qscale ffmpeg` option).
  - - *LA\_ICQ* - intelligent constant quality with lookahead, when the `look_ahead` option is also set.
  - - *ICQ* – intelligent constant quality otherwise.
- Otherwise, a bitrate-based mode is used. For all of those, you should specify at least the desired average bitrate with the `b` option.
  - - *LA* - VBR with lookahead, when the `look_ahead` option is specified.
  - - *VCM* - video conferencing mode, when the `vcm` option is set.
  - - *CBR* - constant bitrate, when `maxrate` is specified and equal to the average bitrate.
  - - *VBR* - variable bitrate, when `maxrate` is specified, but is higher than the average bitrate.
  - - *AVBR* - average VBR mode, when `maxrate` is not specified. This mode is further configured by the `avbr_accuracy` and `avbr_convergence` options.

Note that depending on your system, a different mode than the one you specified may be selected by the encoder. Set the verbosity level to *verbose* or higher to see the actual settings used by the QSV runtime.

Additional libavcodec global options are mapped to MSDK options as follows:

- `g/gop_size` -> `GopPicSize`
- `bf/max_b_frames+1` -> `GopRefDist`
- `rc_init_occupancy/rc_initial_buffer_occupancy` -> `InitialDelayInKB`
- `slices` -> `NumSlice`
- `refs` -> `NumRefFrame`
- `b_strategy/b_frame_strategy` -> `BRefType`
- `cgop/CLOSED_GOP` codec flag -> `GopOptFlag`
- For the *CQP* mode, the `i_qfactor/i_qoffset` and `b_qfactor/b_qoffset` set the difference between *QPP* and *QPI*, and *QPP* and *QPB* respectively.
- Setting the `coder` option to the value *vlc* will make the H.264 encoder use CAVLC instead of CABAC.

## 9.15 snow# TOC

### 9.15.1 Options# TOC

`iterative_dia_size`

dia size for the iterative motion estimation

## 9.16 VAAPI encoders# TOC

Wrappers for hardware encoders accessible via VAAPI.

These encoders only accept input in VAAPI hardware surfaces. If you have input in software frames, use the `hwupload` filter to upload them to the GPU.

The following standard libavcodec options are used:

- `g/gop_size`
- `bf/max_b_frames`
- `profile`
- `level`
- `b/bit_rate`
- `maxrate/rc_max_rate`
- `bufsize/rc_buffer_size`
- `rc_init_occupancy/rc_initial_buffer_occupancy`
- `compression_level`



Speed / quality tradeoff: higher values are faster / worse quality.

- `q / global_quality`

Size / quality tradeoff: higher values are smaller / worse quality.

- `qmin` (only: `qmax` is not supported)
- `i_qfactor / i_quant_factor`
- `i_qoffset / i_quant_offset`
- `b_qfactor / b_quant_factor`
- `b_qoffset / b_quant_offset`

`h264_vaapi`

`profile` sets the value of *profile\_idc* and the *constraint\_set\*\_flags*. `level` sets the value of *level\_idc*.

`low_power`

Use low-power encoding mode.

`coder`

Set entropy encoder (default is *cabac*). Possible values:

‘ac’  
‘cabac’

Use CABAC.

‘vlc’  
‘cavlc’

Use CAVLC.

`hevc_vaapi`

`profile` and `level` set the values of *general\_profile\_idc* and *general\_level\_idc* respectively.

`mjpeg_vaapi`

Always encodes using the standard quantisation and huffman tables - `global_quality` scales the standard quantisation table (range 1-100).

`mpeg2_vaapi`

`profile` and `level` set the value of *profile\_and\_level\_indication*.

No rate control is supported.

vp8\_vaapi

B-frames are not supported.

`global_quality` sets the *q\_idx* used for non-key frames (range 0-127).

`loop_filter_level`

`loop_filter_sharpness`

Manually set the loop filter parameters.

vp9\_vaapi

`global_quality` sets the *q\_idx* used for P-frames (range 0-255).

`loop_filter_level`

`loop_filter_sharpness`

Manually set the loop filter parameters.

B-frames are supported, but the output stream is always in encode order rather than display order. If B-frames are enabled, it may be necessary to use the `vp9_raw_reorder` bitstream filter to modify the output stream to display frames in the correct order.

Only normal frames are produced - the `vp9_superframe` bitstream filter may be required to produce a stream usable with all decoders.

## 9.17 vc2# TOC

SMPTE VC-2 (previously BBC Dirac Pro). This codec was primarily aimed at professional broadcasting but since it supports yuv420, yuv422 and yuv444 at 8 (limited range or full range), 10 or 12 bits, this makes it suitable for other tasks which require low overhead and low compression (like screen recording).

### 9.17.1 Options# TOC

b

Sets target video bitrate. Usually that's around 1:6 of the uncompressed video bitrate (e.g. for 1920x1080 50fps yuv422p10 that's around 400Mbps). Higher values (close to the uncompressed bitrate) turn on lossless compression mode.

field\_order

Enables field coding when set (e.g. to tt - top field first) for interlaced inputs. Should increase compression with interlaced content as it splits the fields and encodes each separately.

`wavelet_depth`

Sets the total amount of wavelet transforms to apply, between 1 and 5 (default). Lower values reduce compression and quality. Less capable decoders may not be able to handle values of `wavelet_depth` over 3.

`wavelet_type`

Sets the transform type. Currently only `5_3` (LeGall) and `9_7` (Deslauriers-Dubuc) are implemented, with `9_7` being the one with better compression and thus is the default.

`slice_width`

`slice_height`

Sets the slice size for each slice. Larger values result in better compression. For compatibility with other more limited decoders use `slice_width` of 32 and `slice_height` of 8.

`tolerance`

Sets the undershoot tolerance of the rate control system in percent. This is to prevent an expensive search from being run.

`qm`

Sets the quantization matrix preset to use by default or when `wavelet_depth` is set to 5

- - *default* Uses the default quantization matrix from the specifications, extended with values for the fifth level. This provides a good balance between keeping detail and omitting artifacts.
- - *flat* Use a completely zeroed out quantization matrix. This increases PSNR but might reduce perception. Use in bogus benchmarks.
- - *color* Reduces detail but attempts to preserve color at extremely low bitrates.

## 10 Subtitles Encoders# TOC

### 10.1 dvdsub# TOC

This codec encodes the bitmap subtitle format that is used in DVDs. Typically they are stored in VOBSUB file pairs (\*.idx + \*.sub), and they can also be used in Matroska files.

#### 10.1.1 Options# TOC

`even_rows_fix`

When set to 1, enable a work-around that makes the number of pixel rows even in all subtitles. This fixes a problem with some players that cut off the bottom row if the number is odd. The work-around just adds a fully transparent row if needed. The overhead is low, typically one byte per subtitle on average.

By default, this work-around is disabled.

## 11 See Also# TOC

ffmpeg, ffplay, ffprobe, ffserver, libavcodec

## 12 Authors# TOC

The FFmpeg developers.

For details about the authorship, see the Git history of the project ([git://source.ffmpeg.org/ffmpeg](http://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 using *makeinfo*.