FFmpeg Bitstream Filters Documentation

Table of Contents

1 Description# TOC

This document describes the bitstream filters provided by the libavcodec library.

A bitstream filter operates on the encoded stream data, and performs bitstream level modifications without performing decoding.

2 Bitstream Filters# TOC

When you configure your FFmpeg build, all the supported bitstream filters are enabled by default. You can list all available ones using the configure option `--list-bsfs`.

You can disable all the bitstream filters using the configure option `--disable-bsfs`, and selectively enable any bitstream filter using the option `--enable-bsf=BSF`, or you can disable a particular bitstream filter using the option `--disable-bsf=BSF`.

The option `-bsfs` of the ff* tools will display the list of all the supported bitstream filters included in your build.

The ff* tools have a -bsf option applied per stream, taking a comma-separated list of filters, whose parameters follow the filter name after a '='.

```
ffmpeg -i INPUT -c:v copy -bsf:v filter1[=opt1=str1:opt2=str2][,filter2] OUTPUT
```

Below is a description of the currently available bitstream filters, with their parameters, if any.

2.1 aac_adtstoasc# TOC

Convert MPEG-2/4 AAC ADTS to an MPEG-4 Audio Specific Configuration bitstream.

This filter creates an MPEG-4 AudioSpecificConfig from an MPEG-2/4 ADTS header and removes the ADTS header.

This filter is required for example when copying an AAC stream from a raw ADTS AAC or an MPEG-TS container to MP4A-LATM, to an FLV file, or to MOV/MP4 files and related formats such as 3GP or M4A. Please note that it is auto-inserted for MP4A-LATM and MOV/MP4 and related formats.

2.2 chomp# TOC

Remove zero padding at the end of a packet.

2.3 dca_core# TOC

Extract the core from a DCA/DTS stream, dropping extensions such as DTS-HD.

2.4 dump_extra# TOC

Add extradata to the beginning of the filtered packets.

The additional argument specifies which packets should be filtered. It accepts the values:

'a'

add extradata to all key packets, but only if *local_header* is set in the `flags2` codec context field

‘`k`’

add extradata to all key packets

‘`e`’

add extradata to all packets

If not specified it is assumed ‘`k`’.

For example the following `ffmpeg` command forces a global header (thus disabling individual packet headers) in the H.264 packets generated by the `libx264` encoder, but corrects them by adding the header stored in extradata to the key packets:

```
ffmpeg -i INPUT -map 0 -flags:v +global_header -c:v libx264 -bsf:v dump_extra out.ts
```

2.5 extract_extradata# TOC

Extract the in-band extradata.

Certain codecs allow the long-term headers (e.g. MPEG-2 sequence headers, or H.264/HEVC (VPS/)SPS/PPS) to be transmitted either "in-band" (i.e. as a part of the bitstream containing the coded frames) or "out of band" (e.g. on the container level). This latter form is called "extradata" in FFmpeg terminology.

This bitstream filter detects the in-band headers and makes them available as extradata.

`remove`

When this option is enabled, the long-term headers are removed from the bitstream after extraction.

2.6 h264_metadata# TOC

Modify metadata embedded in an H.264 stream.

`aud`

Insert or remove AUD NAL units in all access units of the stream.

‘`insert`’
‘`remove`’

`sample_aspect_ratio`

Set the sample aspect ratio of the stream in the VUI parameters.

`video_format`
`video_full_range_flag`

Set the video format in the stream (see H.264 section E.2.1 and table E-2).

`colour_primaries`
`transfer_characteristics`
`matrix_coefficients`

Set the colour description in the stream (see H.264 section E.2.1 and tables E-3, E-4 and E-5).

`chroma_sample_loc_type`

Set the chroma sample location in the stream (see H.264 section E.2.1 and figure E-1).

`tick_rate`

Set the tick rate (num_units_in_tick / time_scale) in the VUI parameters. This is the smallest time unit representable in the stream, and in many cases represents the field rate of the stream (double the frame rate).

`fixed_frame_rate_flag`

Set whether the stream has fixed framerate - typically this indicates that the framerate is exactly half the tick rate, but the exact meaning is dependent on interlacing and the picture structure (see H.264 section E.2.1 and table E-6).

`crop_left`
`crop_right`
`crop_top`
`crop_bottom`

Set the frame cropping offsets in the SPS. These values will replace the current ones if the stream is already cropped.

These fields are set in pixels. Note that some sizes may not be representable if the chroma is subsampled or the stream is interlaced (see H.264 section 7.4.2.1.1).

`sei_user_data`

Insert a string as SEI unregistered user data. The argument must be of the form *UUID+string*, where the UUID is as hex digits possibly separated by hyphens, and the string can be anything.

For example, '`086f3693-b7b3-4f2c-9653-21492feee5b8+hello`' will insert the string "hello" associated with the given UUID.

2.7 h264_mp4toannexb# TOC

Convert an H.264 bitstream from length prefixed mode to start code prefixed mode (as defined in the Annex B of the ITU-T H.264 specification).

This is required by some streaming formats, typically the MPEG-2 transport stream format (muxer `mpegts`).

For example to remux an MP4 file containing an H.264 stream to mpegts format with `ffmpeg`, you can use the command:

```
ffmpeg -i INPUT.mp4 -codec copy -bsf:v h264_mp4toannexb OUTPUT.ts
```

Please note that this filter is auto-inserted for MPEG-TS (muxer `mpegts`) and raw H.264 (muxer `h264`) output formats.

2.8 h264_redundant_pps# TOC

This applies a specific fixup to some Blu-ray streams which contain redundant PPSs modifying irrelevant parameters of the stream which confuse other transformations which require correct extradata.

A new single global PPS is created, and all of the redundant PPSs within the stream are removed.

2.9 hevc_metadata# TOC

Modify metadata embedded in an HEVC stream.

`aud`

Insert or remove AUD NAL units in all access units of the stream.

`'insert'`
`'remove'`

`sample_aspect_ratio`

Set the sample aspect ratio in the stream in the VUI parameters.

`video_format`
`video_full_range_flag`

Set the video format in the stream (see H.265 section E.3.1 and table E.2).

`colour_primaries`
`transfer_characteristics`
`matrix_coefficients`

Set the colour description in the stream (see H.265 section E.3.1 and tables E.3, E.4 and E.5).

`chroma_sample_loc_type`

Set the chroma sample location in the stream (see H.265 section E.3.1 and figure E.1).

`tick_rate`

Set the tick rate in the VPS and VUI parameters (num_units_in_tick / time_scale). Combined with `num_ticks_poc_diff_one`, this can set a constant framerate in the stream. Note that it is likely to be overridden by container parameters when the stream is in a container.

`num_ticks_poc_diff_one`

Set poc_proportional_to_timing_flag in VPS and VUI and use this value to set num_ticks_poc_diff_one_minus1 (see H.265 sections 7.4.3.1 and E.3.1). Ignored if `tick_rate` is not also set.

`crop_left`
`crop_right`
`crop_top`
`crop_bottom`

Set the conformance window cropping offsets in the SPS. These values will replace the current ones if the stream is already cropped.

These fields are set in pixels. Note that some sizes may not be representable if the chroma is subsampled (H.265 section 7.4.3.2.1).

2.10 hevc_mp4toannexb# TOC

Convert an HEVC/H.265 bitstream from length prefixed mode to start code prefixed mode (as defined in the Annex B of the ITU-T H.265 specification).

This is required by some streaming formats, typically the MPEG-2 transport stream format (muxer `mpegts`).

For example to remux an MP4 file containing an HEVC stream to mpegts format with `ffmpeg`, you can use the command:

```
ffmpeg -i INPUT.mp4 -codec copy -bsf:v hevc_mp4toannexb OUTPUT.ts
```

Please note that this filter is auto-inserted for MPEG-TS (muxer `mpegts`) and raw HEVC/H.265 (muxer `h265` or `hevc`) output formats.

2.11 imxdump# TOC

Modifies the bitstream to fit in MOV and to be usable by the Final Cut Pro decoder. This filter only applies to the mpeg2video codec, and is likely not needed for Final Cut Pro 7 and newer with the appropriate `-tag:v`.

For example, to remux 30 MB/sec NTSC IMX to MOV:

```
ffmpeg -i input.mxf -c copy -bsf:v imxdump -tag:v mx3n output.mov
```

2.12 mjpeg2jpeg# TOC

Convert MJPEG/AVI1 packets to full JPEG/JFIF packets.

MJPEG is a video codec wherein each video frame is essentially a JPEG image. The individual frames can be extracted without loss, e.g. by

```
ffmpeg -i ../some_mjpeg.avi -c:v copy frames_%d.jpg
```

Unfortunately, these chunks are incomplete JPEG images, because they lack the DHT segment required for decoding. Quoting from http://www.digitalpreservation.gov/formats/fdd/fdd000063.shtml:

Avery Lee, writing in the rec.video.desktop newsgroup in 2001, commented that "MJPEG, or at least the MJPEG in AVIs having the MJPG fourcc, is restricted JPEG with a fixed – and *omitted* – Huffman table. The JPEG must be YCbCr colorspace, it must be 4:2:2, and it must use basic Huffman encoding, not arithmetic or progressive. . . . You can indeed extract the MJPEG frames and decode them with a regular JPEG decoder, but you have to prepend the DHT segment to them, or else the decoder won’t have any idea how to decompress the data. The exact table necessary is given in the OpenDML spec."

This bitstream filter patches the header of frames extracted from an MJPEG stream (carrying the AVI1 header ID and lacking a DHT segment) to produce fully qualified JPEG images.

```
ffmpeg -i mjpeg-movie.avi -c:v copy -bsf:v mjpeg2jpeg frame_%d.jpg
exiftran -i -9 frame*.jpg
ffmpeg -i frame_%d.jpg -c:v copy rotated.avi
```

2.13 mjpegadump# TOC

Add an MJPEG A header to the bitstream, to enable decoding by Quicktime.

2.14 mov2textsub# TOC

Extract a representable text file from MOV subtitles, stripping the metadata header from each subtitle packet.

See also the text2movsub filter.

2.15 mp3decomp# TOC

Decompress non-standard compressed MP3 audio headers.

2.16 mpeg2_metadata# TOC

Modify metadata embedded in an MPEG-2 stream.

`display_aspect_ratio`

Set the display aspect ratio in the stream.

The following fixed values are supported:

```
4/3
16/9
221/100
```

Any other value will result in square pixels being signalled instead (see H.262 section 6.3.3 and table 6-3).

`frame_rate`

Set the frame rate in the stream. This is constructed from a table of known values combined with a small multiplier and divisor - if the supplied value is not exactly representable, the nearest representable value will be used instead (see H.262 section 6.3.3 and table 6-4).

`video_format`

Set the video format in the stream (see H.262 section 6.3.6 and table 6-6).

`colour_primaries`
`transfer_characteristics`
`matrix_coefficients`

Set the colour description in the stream (see H.262 section 6.3.6 and tables 6-7, 6-8 and 6-9).

2.17 mpeg4_unpack_bframes# TOC

Unpack DivX-style packed B-frames.

DivX-style packed B-frames are not valid MPEG-4 and were only a workaround for the broken Video for Windows subsystem. They use more space, can cause minor AV sync issues, require more CPU power to decode (unless the player has some decoded picture queue to compensate the 2,0,2,0 frame per packet style) and cause trouble if copied into a standard container like mp4 or mpeg-ps/ts, because MPEG-4 decoders may not be able to decode them, since they are not valid MPEG-4.

For example to fix an AVI file containing an MPEG-4 stream with DivX-style packed B-frames using `ffmpeg`, you can use the command:

```
ffmpeg -i INPUT.avi -codec copy -bsf:v mpeg4_unpack_bframes OUTPUT.avi
```

2.18 noise# TOC

Damages the contents of packets or simply drops them without damaging the container. Can be used for fuzzing or testing error resilience/concealment.

Parameters:

`amount`

> A numeral string, whose value is related to how often output bytes will be modified. Therefore, values below or equal to 0 are forbidden, and the lower the more frequent bytes will be modified, with 1 meaning every byte is modified.

`dropamount`

> A numeral string, whose value is related to how often packets will be dropped. Therefore, values below or equal to 0 are forbidden, and the lower the more frequent packets will be dropped, with 1 meaning every packet is dropped.

The following example applies the modification to every byte but does not drop any packets.

```
ffmpeg -i INPUT -c copy -bsf noise[=1] output.mkv
```

2.19 null# TOC

This bitstream filter passes the packets through unchanged.

2.20 remove_extra# TOC

Remove extradata from packets.

It accepts the following parameter:

`freq`

> Set which frame types to remove extradata from.
>
> ‘`k`’
>
> > Remove extradata from non-keyframes only.
>
> ‘`keyframe`’

Remove extradata from keyframes only.

‘e, all’

Remove extradata from all frames.

2.21 text2movsub# TOC

Convert text subtitles to MOV subtitles (as used by the mov_text codec) with metadata headers.

See also the mov2textsub filter.

2.22 trace_headers# TOC

Log trace output containing all syntax elements in the coded stream headers (everything above the level of individual coded blocks). This can be useful for debugging low-level stream issues.

Supports H.264, H.265 and MPEG-2.

2.23 vp9_superframe# TOC

Merge VP9 invisible (alt-ref) frames back into VP9 superframes. This fixes merging of split/segmented VP9 streams where the alt-ref frame was split from its visible counterpart.

2.24 vp9_superframe_split# TOC

Split VP9 superframes into single frames.

2.25 vp9_raw_reorder# TOC

Given a VP9 stream with correct timestamps but possibly out of order, insert additional show-existing-frame packets to correct the ordering.

3 See Also# TOC

ffmpeg, ffplay, ffprobe, ffserver, libavcodec

4 Authors# TOC

The FFmpeg developers.

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

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

This document was generated using *makeinfo*.