

NAME

ffplay – FFplay media player

SYNOPSIS

ffplay [*options*] [*input_url*]

DESCRIPTION

FFplay is a very simple and portable media player using the FFmpeg libraries and the SDL library. It is mostly used as a testbed for the various FFmpeg APIs.

OPTIONS

All the numerical options, if not specified otherwise, accept a string representing a number as input, which may be followed by one of the SI unit prefixes, for example: 'K', 'M', or 'G'.

If 'i' is appended to the SI unit prefix, the complete prefix will be interpreted as a unit prefix for binary multiples, which are based on powers of 1024 instead of powers of 1000. Appending 'B' to the SI unit prefix multiplies the value by 8. This allows using, for example: 'KB', 'MiB', 'G' and 'B' as number suffixes.

Options which do not take arguments are boolean options, and set the corresponding value to true. They can be set to false by prefixing the option name with "no". For example using "--nofoo" will set the boolean option with name "foo" to false.

Stream specifiers

Some options are applied per-stream, e.g. bitrate or codec. Stream specifiers are used to precisely specify which stream(s) a given option belongs to.

A stream specifier is a string generally appended to the option name and separated from it by a colon. E.g. `-codec:a:1 ac3` contains the `a:1` stream specifier, which matches the second audio stream. Therefore, it would select the ac3 codec for the second audio stream.

A stream specifier can match several streams, so that the option is applied to all of them. E.g. the stream specifier in `-b:a 128k` matches all audio streams.

An empty stream specifier matches all streams. For example, `-codec copy` or `-codec: copy` would copy all the streams without reencoding.

Possible forms of stream specifiers are:

stream_index

Matches the stream with this index. E.g. `-threads:1 4` would set the thread count for the second stream to 4. If *stream_index* is used as an additional stream specifier (see below), then it selects stream number *stream_index* from the matching streams. Stream numbering is based on the order of the streams as detected by libavformat except when a program ID is also specified. In this case it is based on the ordering of the streams in the program.

stream_type[:additional_stream_specifier]

stream_type is one of following: 'v' or 'V' for video, 'a' for audio, 's' for subtitle, 'd' for data, and 't' for attachments. 'v' matches all video streams, 'V' only matches video streams which are not attached pictures, video thumbnails or cover arts. If *additional_stream_specifier* is used, then it matches streams which both have this type and match the *additional_stream_specifier*. Otherwise, it matches all streams of the specified type.

p:program_id[:additional_stream_specifier]

Matches streams which are in the program with the id *program_id*. If *additional_stream_specifier* is used, then it matches streams which both are part of the program and match the *additional_stream_specifier*.

#stream_id or i:stream_id

Match the stream by stream id (e.g. PID in MPEG-TS container).

m:key[:value]

Matches streams with the metadata tag *key* having the specified value. If *value* is not given, matches streams that contain the given tag with any value.

- u** Matches streams with usable configuration, the codec must be defined and the essential information such as video dimension or audio sample rate must be present.

Note that in **ffmpeg**, matching by metadata will only work properly for input files.

Generic options

These options are shared amongst the ff* tools.

-L Show license.

-h, -?, -help, --help [*arg*]

Show help. An optional parameter may be specified to print help about a specific item. If no argument is specified, only basic (non advanced) tool options are shown.

Possible values of *arg* are:

long

Print advanced tool options in addition to the basic tool options.

full Print complete list of options, including shared and private options for encoders, decoders, demuxers, muxers, filters, etc.

decoder=decoder_name

Print detailed information about the decoder named *decoder_name*. Use the **-decoders** option to get a list of all decoders.

encoder=encoder_name

Print detailed information about the encoder named *encoder_name*. Use the **-encoders** option to get a list of all encoders.

demuxer=demuxer_name

Print detailed information about the demuxer named *demuxer_name*. Use the **-formats** option to get a list of all demuxers and muxers.

muxer=muxer_name

Print detailed information about the muxer named *muxer_name*. Use the **-formats** option to get a list of all muxers and demuxers.

filter=filter_name

Print detailed information about the filter named *filter_name*. Use the **-filters** option to get a list of all filters.

bsf=bitstream_filter_name

Print detailed information about the bitstream filter named *bitstream_filter_name*. Use the **-bsfs** option to get a list of all bitstream filters.

protocol=protocol_name

Print detailed information about the protocol named *protocol_name*. Use the **-protocols** option to get a list of all protocols.

-version

Show version.

-buildconf

Show the build configuration, one option per line.

-formats

Show available formats (including devices).

-demuxers

Show available demuxers.

-muxers

Show available muxers.

-devices

Show available devices.

-codecs

Show all codecs known to libavcodec.

Note that the term 'codec' is used throughout this documentation as a shortcut for what is more correctly called a media bitstream format.

-decoders

Show available decoders.

-encoders

Show all available encoders.

-bsfs

Show available bitstream filters.

-protocols

Show available protocols.

-filters

Show available libavfilter filters.

-pix_fmts

Show available pixel formats.

-sample_fmts

Show available sample formats.

-layouts

Show channel names and standard channel layouts.

-colors

Show recognized color names.

-sources *device*[,*opt1=val1*][,*opt2=val2*]...

Show autodetected sources of the input device. Some devices may provide system-dependent source names that cannot be autodetected. The returned list cannot be assumed to be always complete.

```
ffmpeg -sources pulse,server=192.168.0.4
```

-sinks *device*[,*opt1=val1*][,*opt2=val2*]...

Show autodetected sinks of the output device. Some devices may provide system-dependent sink names that cannot be autodetected. The returned list cannot be assumed to be always complete.

```
ffmpeg -sinks pulse,server=192.168.0.4
```

-loglevel [*flags+*]*loglevel* | **-v** [*flags+*]*loglevel*

Set logging level and flags used by the library.

The optional *flags* prefix can consist of the following values:

repeat

Indicates that repeated log output should not be compressed to the first line and the "Last message repeated n times" line will be omitted.

level

Indicates that log output should add a [*level*] prefix to each message line. This can be used as an alternative to log coloring, e.g. when dumping the log to file.

Flags can also be used alone by adding a '+'/'-' prefix to set/reset a single flag without affecting other *flags* or changing *loglevel*. When setting both *flags* and *loglevel*, a '+' separator is expected between the last *flags* value and before *loglevel*.

loglevel is a string or a number containing one of the following values:

quiet, -8

Show nothing at all; be silent.

panic, 0

Only show fatal errors which could lead the process to crash, such as an assertion failure. This is not currently used for anything.

fatal, 8

Only show fatal errors. These are errors after which the process absolutely cannot continue.

error, 16

Show all errors, including ones which can be recovered from.

warning, 24

Show all warnings and errors. Any message related to possibly incorrect or unexpected events will be shown.

info, 32

Show informative messages during processing. This is in addition to warnings and errors. This is the default value.

verbose, 40

Same as `info`, except more verbose.

debug, 48

Show everything, including debugging information.

trace, 56

For example to enable repeated log output, add the `level` prefix, and set *loglevel* to `verbose`:

```
ffmpeg -loglevel repeat+level+verbose -i input output
```

Another example that enables repeated log output without affecting current state of `level` prefix flag or *loglevel*:

```
ffmpeg [...] -loglevel +repeat
```

By default the program logs to `stderr`. If coloring is supported by the terminal, colors are used to mark errors and warnings. Log coloring can be disabled setting the environment variable `AV_LOG_FORCE_NOCOLOR`, or can be forced setting the environment variable `AV_LOG_FORCE_COLOR`.

-report

Dump full command line and log output to a file named *program-YYYYMMDD-HHMMSS.log* in the current directory. This file can be useful for bug reports. It also implies `-loglevel debug`.

Setting the environment variable `FFREPORT` to any value has the same effect. If the value is a `':'`-separated key=value sequence, these options will affect the report; option values must be escaped if they contain special characters or the options delimiter `':'` (see the “Quoting and escaping” section in the `ffmpeg-utils` manual).

The following options are recognized:

file set the file name to use for the report; `%p` is expanded to the name of the program, `%t` is expanded to a timestamp, `%%` is expanded to a plain `%`

level

set the log verbosity level using a numerical value (see `-loglevel`).

For example, to output a report to a file named *ffreport.log* using a log level of 32 (alias for log level `info`):

```
FFREPORT=file=ffreport.log:level=32 ffmpeg -i input output
```

Errors in parsing the environment variable are not fatal, and will not appear in the report.

-hide_banner

Suppress printing banner.

All FFmpeg tools will normally show a copyright notice, build options and library versions. This option can be used to suppress printing this information.

-cpuflags flags (*global*)

Allows setting and clearing cpu flags. This option is intended for testing. Do not use it unless you know what you're doing.

```
ffmpeg -cpuflags -sse+mmx ...
ffmpeg -cpuflags mmx ...
ffmpeg -cpuflags 0 ...
```

Possible flags for this option are:

x86

```
mmx
mmxext
sse
sse2
sse2slow
sse3
sse3slow
ssse3
atom
sse4.1
sse4.2
avx
avx2
xop
fma3
fma4
3dnow
3dnowext
bmi1
bmi2
cmov
```

ARM

```
armv5te
armv6
armv6t2
vfp
vfpv3
neon
setend
```

AArch64

```
armv8
vfp
neon
```

PowerPC

```
altivec
```

Specific Processors**pentium2****pentium3****pentium4****k6****k62****athlon****athlonxp****k8****-max_alloc** *bytes*

Set the maximum size limit for allocating a block on the heap by ffmpeg's family of malloc functions. Exercise **extreme caution** when using this option. Don't use if you do not understand the full consequence of doing so. Default is INT_MAX.

AVOptions

These options are provided directly by the libavformat, libavdevice and libavcodec libraries. To see the list of available AVOptions, use the **-help** option. They are separated into two categories:

generic

These options can be set for any container, codec or device. Generic options are listed under AVFormatContext options for containers/devices and under AVCodecContext options for codecs.

private

These options are specific to the given container, device or codec. Private options are listed under their corresponding containers/devices/codecs.

For example to write an ID3v2.3 header instead of a default ID3v2.4 to an MP3 file, use the **id3v2_version** private option of the MP3 muxer:

```
ffmpeg -i input.flac -id3v2_version 3 out.mp3
```

All codec AVOptions are per-stream, and thus a stream specifier should be attached to them:

```
ffmpeg -i multichannel.mxf -map 0:v:0 -map 0:a:0 -map 0:a:0 -c:a:0 ac3 -b
```

In the above example, a multichannel audio stream is mapped twice for output. The first instance is encoded with codec ac3 and bitrate 640k. The second instance is downmixed to 2 channels and encoded with codec aac. A bitrate of 128k is specified for it using absolute index of the output stream.

Note: the **-nooption** syntax cannot be used for boolean AVOptions, use **-option 0/-option 1**.

Note: the old undocumented way of specifying per-stream AVOptions by prepending v/a/s to the options name is now obsolete and will be removed soon.

Main options**-x** *width*

Force displayed width.

-y *height*

Force displayed height.

-s *size*

Set frame size (WxH or abbreviation), needed for videos which do not contain a header with the frame size like raw YUV. This option has been deprecated in favor of private options, try **-video_size**.

-fs Start in fullscreen mode.**-an**

Disable audio.

-vn

Disable video.

- sn**
Disable subtitles.
- ss *pos***
Seek to *pos*. Note that in most formats it is not possible to seek exactly, so **ffplay** will seek to the nearest seek point to *pos*.

pos must be a time duration specification, see **the Time duration section in the ffmpeg-utils (1) manual**.
- t *duration***
Play *duration* seconds of audio/video.

duration must be a time duration specification, see **the Time duration section in the ffmpeg-utils (1) manual**.
- bytes**
Seek by bytes.
- seek_interval**
Set custom interval, in seconds, for seeking using left/right keys. Default is 10 seconds.
- nodisp**
Disable graphical display.
- noborder**
Borderless window.
- alwaysontop**
Window always on top. Available on: X11 with SDL >= 2.0.5, Windows SDL >= 2.0.6.
- volume**
Set the startup volume. 0 means silence, 100 means no volume reduction or amplification. Negative values are treated as 0, values above 100 are treated as 100.
- f *fmt***
Force format.
- window_title *title***
Set window title (default is the input filename).
- left *title***
Set the x position for the left of the window (default is a centered window).
- top *title***
Set the y position for the top of the window (default is a centered window).
- loop *number***
Loops movie playback <number> times. 0 means forever.
- showmode *mode***
Set the show mode to use. Available values for *mode* are:
 - 0, video**
show video
 - 1, waves**
show audio waves
 - 2, rdft**
show audio frequency band using RDFT ((Inverse) Real Discrete Fourier Transform)
 Default value is “video”, if video is not present or cannot be played “rdft” is automatically selected.
 You can interactively cycle through the available show modes by pressing the key **w**.

-vf *filtergraph*

Create the filtergraph specified by *filtergraph* and use it to filter the video stream.

filtergraph is a description of the filtergraph to apply to the stream, and must have a single video input and a single video output. In the filtergraph, the input is associated to the label *in*, and the output to the label *out*. See the ffmpeg-filters manual for more information about the filtergraph syntax.

You can specify this parameter multiple times and cycle through the specified filtergraphs along with the show modes by pressing the key **w**.

-af *filtergraph*

filtergraph is a description of the filtergraph to apply to the input audio. Use the option “-filters” to show all the available filters (including sources and sinks).

-i *input_url*

Read *input_url*.

Advanced options**-pix_fmt** *format*

Set pixel format. This option has been deprecated in favor of private options, try **-pixel_format**.

-stats

Print several playback statistics, in particular show the stream duration, the codec parameters, the current position in the stream and the audio/video synchronisation drift. It is shown by default, unless the log level is lower than **info**. Its display can be forced by manually specifying this option. To disable it, you need to specify **-nostats**.

-fast

Non-spec-compliant optimizations.

-genpts

Generate pts.

-sync *type*

Set the master clock to audio (*type=audio*), video (*type=video*) or external (*type=ext*). Default is audio. The master clock is used to control audio-video synchronization. Most media players use audio as master clock, but in some cases (streaming or high quality broadcast) it is necessary to change that. This option is mainly used for debugging purposes.

-ast *audio_stream_specifier*

Select the desired audio stream using the given stream specifier. The stream specifiers are described in the **Stream specifiers** chapter. If this option is not specified, the “best” audio stream is selected in the program of the already selected video stream.

-vst *video_stream_specifier*

Select the desired video stream using the given stream specifier. The stream specifiers are described in the **Stream specifiers** chapter. If this option is not specified, the “best” video stream is selected.

-sst *subtitle_stream_specifier*

Select the desired subtitle stream using the given stream specifier. The stream specifiers are described in the **Stream specifiers** chapter. If this option is not specified, the “best” subtitle stream is selected in the program of the already selected video or audio stream.

-autoexit

Exit when video is done playing.

-exitonkeydown

Exit if any key is pressed.

-exitonmousedown

Exit if any mouse button is pressed.

-codec:*media_specifier codec_name*

Force a specific decoder implementation for the stream identified by *media_specifier*, which can assume the values *a* (audio), *v* (video), and *s* subtitle.

-acodec *codec_name*

Force a specific audio decoder.

-vcodec *codec_name*

Force a specific video decoder.

-scodec *codec_name*

Force a specific subtitle decoder.

-autorotate

Automatically rotate the video according to file metadata. Enabled by default, use **-noautorotate** to disable it.

-framedrop

Drop video frames if video is out of sync. Enabled by default if the master clock is not set to video. Use this option to enable frame dropping for all master clock sources, use **-noframedrop** to disable it.

-inbuf

Do not limit the input buffer size, read as much data as possible from the input as soon as possible. Enabled by default for realtime streams, where data may be dropped if not read in time. Use this option to enable infinite buffers for all inputs, use **-noinbuf** to disable it.

-filter_threads *nb_threads*

Defines how many threads are used to process a filter pipeline. Each pipeline will produce a thread pool with this many threads available for parallel processing. The default is 0 which means that the thread count will be determined by the number of available CPUs.

While playing**q, ESC**

Quit.

f Toggle full screen.**p, SPC**

Pause.

m Toggle mute.**9, 0** Decrease and increase volume respectively.**/, *** Decrease and increase volume respectively.**a** Cycle audio channel in the current program.**v** Cycle video channel.**t** Cycle subtitle channel in the current program.**c** Cycle program.**w** Cycle video filters or show modes.**s** Step to the next frame.

Pause if the stream is not already paused, step to the next video frame, and pause.

left/right

Seek backward/forward 10 seconds.

down/up

Seek backward/forward 1 minute.

page down/page up

Seek to the previous/next chapter. or if there are no chapters Seek backward/forward 10 minutes.

right mouse click

Seek to percentage in file corresponding to fraction of width.

left mouse double-click

Toggle full screen.

SEE ALSO

ffplay-all(1), **ffmpeg**(1), **ffprobe**(1), **ffmpeg-utils**(1), **ffmpeg-scaler**(1), **ffmpeg-resampler**(1), **ffmpeg-codecs**(1), **ffmpeg-bitstream-filters**(1), **ffmpeg-formats**(1), **ffmpeg-devices**(1), **ffmpeg-protocols**(1), **ffmpeg-filters**(1)

AUTHORS

The FFmpeg developers.

For details about the authorship, see the Git history of the project ([git://source.ffmpeg.org/ffmpeg](https://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>](http://source.ffmpeg.org).

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