NAME
flac — Free Lossless Audio Codec

SYNOPSIS
flac [*OPTIONS*] [*infile.wav* | *infile.rf64* | *infile.aiff* | *infile.raw* | *infile.flac* | *infile.oga* | *infile.ogg* | −] ...

flac [−d | −−decode | −t | −−test | −a | −−analyze] [*OPTIONS*] [*infile.flac* | *infile.oga* | *infile.ogg* | −] ...

DESCRIPTION
flac is a command-line tool for encoding, decoding, testing and analyzing FLAC streams.

OPTIONS
A summary of options is included below. For a complete description, see the HTML documentation.

General Options

-v, **−−version**
> Show the flac version number

-h, **−−help**
> Show basic usage and a list of all options

-H, **−−explain**
> Show detailed explanation of usage and all options

-d, **−−decode**
> Decode (the default behavior is to encode)

-t, **−−test**
> Test a flac encoded file (same as −d except no decoded file is written)

-a, **−−analyze**
> Analyze a FLAC encoded file (same as −d except an analysis file is written)

-c, **−−stdout**
> Write output to stdout

-s, **−−silent**
> Silent mode (do not write runtime encode/decode statistics to stderr)

−−totally-silent
> Do not print anything of any kind, including warnings or errors. The exit code will be the only way to determine successful completion.

−−no-utf8-convert
> Do not convert tags from local charset to UTF-8. This is useful for scripts, and setting tags in situations where the locale is wrong. This option must appear before any tag options!

-w, **−−warnings-as-errors**
> Treat all warnings as errors (which cause flac to terminate with a non-zero exit code).

-f, **−−force**
> Force overwriting of output files. By default, flac warns that the output file already exists and continues to the next file.

-o *filename*, **−−output-name**=*filename*
> Force the output file name (usually flac just changes the extension). May only be used when encoding a single file. May not be used in conjunction with −−output-prefix.

−−output-prefix=*string*
> Prefix each output file name with the given string. This can be useful for encoding or decoding files to a different directory. Make sure if your string is a path name that it ends with a trailing '/' (slash).

−−delete-input-file

      Automatically delete the input file after a successful encode or decode. If there was an error (including a verify error) the input file is left intact.

−−preserve-modtime

      Output files have their timestamps/permissions set to match those of their inputs (this is default). Use −−no-preserve-modtime to make output files have the current time and default permissions.

−−keep-foreign-metadata

      If encoding, save WAVE, RF64, or AIFF non-audio chunks in FLAC metadata. If decoding, restore any saved non-audio chunks from FLAC metadata when writing the decoded file. Foreign metadata cannot be transcoded, e.g. WAVE chunks saved in a FLAC file cannot be restored when decoding to AIFF. Input and output must be regular files (not stdin or stdout).

−−skip={*#|mm:ss.ss*}

      Skip over the first number of samples of the input. This works for both encoding and decoding, but not testing. The alternative form mm:ss.ss can be used to specify minutes, seconds, and fractions of a second.

−−until={*#|[+|-]mm:ss.ss*}

      Stop at the given sample number for each input file. This works for both encoding and decoding, but not testing. The given sample number is not included in the decoded output. The alternative form mm:ss.ss can be used to specify minutes, seconds, and fractions of a second. If a '+' (plus) sign is at the beginning, the −−until point is relative to the −−skip point. If a '−' (minus) sign is at the beginning, the −−until point is relative to end of the audio.

−−ogg

      When encoding, generate Ogg FLAC output instead of native FLAC. Ogg FLAC streams are FLAC streams wrapped in an Ogg transport layer. The resulting file should have an '.oga' extension and will still be decodable by flac.

      When decoding, force the input to be treated as Ogg FLAC. This is useful when piping input from stdin or when the filename does not end in '.oga' or '.ogg'.

−−serial-number=#

      When used with −−ogg, specifies the serial number to use for the first Ogg FLAC stream, which is then incremented for each additional stream. When encoding and no serial number is given, flac uses a random number for the first stream, then increments it for each additional stream. When decoding and no number is given, flac uses the serial number of the first page.

Analysis Options

−−residual-text

      Includes the residual signal in the analysis file. This will make the file very big, much larger than even the decoded file.

−−residual-gnuplot

      Generates a gnuplot file for every subframe; each file will contain the residual distribution of the subframe. This will create a lot of files.

Decoding Options

−−cue=[#.#][−[#.#]]

      Set the beginning and ending cuepoints to decode. The optional first #.# is the track and index point at which decoding will start; the default is the beginning of the stream. The optional second #.# is the track and index point at which decoding will end; the default is the end of the stream. If the cuepoint does not exist, the closest one before it (for the start point) or after it (for the end point) will be used. If those don't exist, the start of the stream (for the start point) or end of the stream (for the end point) will be used. The cuepoints are merely translated into sample numbers then used as −−skip and −−until. A CD track can always be cued by, for example, −−cue=9.1-10.1 for track 9, even if the CD has no 10th track.

-F, **−−decode-through-errors**

By default flac stops decoding with an error and removes the partially decoded file if it encounters a bitstream error. With −F, errors are still printed but flac will continue decoding to completion. Note that errors may cause the decoded audio to be missing some samples or have silent sections.

−−apply-replaygain-which-is-not-lossless[=<specification>]

Applies ReplayGain values while decoding.

WARNING: THIS IS NOT LOSSLESS. DECODED AUDIO WILL NOT BE IDENTICAL TO THE ORIGINAL WITH THIS OPTION.

The equals sign and <specification> is optional. If omitted, the default is 0aLn1.

The <specification> is a shorthand notation for describing how to apply ReplayGain. All components are optional but order is important. ’[]’ means ’optional’. ’|’ means ’or’. ’{}’ means required. The format is:

[<preamp>][a|t][l|L][n{0|1|2|3}]

preamp   A floating point number in dB. This is added to the existing gain value.

a|t   Specify ’a’ to use the album gain, or ’t’ to use the track gain. If tags for the preferred kind (album/track) do not exist but tags for the other (track/album) do, those will be used instead.

l|L   Specify ’l’ to peak-limit the output, so that the ReplayGain peak value is full-scale. Specify ’L’ to use a 6dB hard limiter that kicks in when the signal approaches full-scale.

n{0|1|2|3}   Specify the amount of noise shaping. ReplayGain synthesis happens in floating point; the result is dithered before converting back to integer. This quantization adds noise. Noise shaping tries to move the noise where you won’t hear it as much. 0 means no noise shaping, 1 means ’low’, 2 means ’medium’, 3 means ’high’.

For example, the default of 0aLn1 means 0dB preamp, use album gain, 6dB hard limit, low noise shaping.

−−apply-replaygain-which-is-not-lossless=3 means 3dB preamp, use album gain, no limiting, no noise shaping.

flac uses the ReplayGain tags for the calculation. If a stream does not have the required tags or they can’t be parsed, decoding will continue with a warning, and no ReplayGain is applied to that stream.

Encoding Options

-V, **−−verify**

Verify a correct encoding by decoding the output in parallel and comparing to the original

−−lax   Allow encoder to generate non-Subset files. The resulting FLAC file may not be streamable or might have trouble being played in all players (especially hardware devices), so you should only use this option in combination with custom encoding options meant for archival.

−−replay-gain

Calculate ReplayGain values and store them as FLAC tags, similar to vorbisgain. Title gains/peaks will be computed for each input file, and an album gain/peak will be computed for all files. All input files must have the same resolution, sample rate, and number of channels. Only mono and stereo files are allowed, and the sample rate must be one of 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, or 48 kHz. Also note that this option may leave a few extra bytes in a PADDING block as the exact size of the tags is not known until all files are processed. Note that this option cannot be used when encoding to standard output (stdout).

−−cuesheet=*filename*

        Import the given cuesheet file and store it in a CUESHEET metadata block. This option may only be used when encoding a single file. A seekpoint will be added for each index point in the cuesheet to the SEEKTABLE unless −−no-cued-seekpoints is specified.

−−picture={*FILENAME*|*SPECIFICATION*}

        Import a picture and store it in a PICTURE metadata block. More than one −−picture command can be specified. Either a filename for the picture file or a more complete specification form can be used. The SPECIFICATION is a string whose parts are separated by | (pipe) characters. Some parts may be left empty to invoke default values. FILENAME is just shorthand for "||||FILENAME". The format of SPECIFICATION is

        [TYPE]|[MIME-TYPE]|[DESCRIPTION]|[WIDTHxHEIGHTxDEPTH[/COLORS]]|FILE

        TYPE is optional; it is a number from one of:

        0: Other

        1: 32x32 pixels 'file icon' (PNG only)

        2: Other file icon

        3: Cover (front)

        4: Cover (back)

        5: Leaflet page

        6: Media (e.g. label side of CD)

        7: Lead artist/lead performer/soloist

        8: Artist/performer

        9: Conductor

        10: Band/Orchestra

        11: Composer

        12: Lyricist/text writer

        13: Recording Location

        14: During recording

        15: During performance

        16: Movie/video screen capture

        17: A bright coloured fish

        18: Illustration

        19: Band/artist logotype

        20: Publisher/Studio logotype

        The default is 3 (front cover). There may only be one picture each of type 1 and 2 in a file.

        MIME-TYPE is optional; if left blank, it will be detected from the file. For best compatibility with players, use pictures with MIME type image/jpeg or image/png. The MIME type can also be −−> to mean that FILE is actually a URL to an image, though this use is discouraged.

        DESCRIPTION is optional; the default is an empty string.

        The next part specifies the resolution and color information. If the MIME-TYPE is image/jpeg, image/png, or image/gif, you can usually leave this empty and they can be detected from the file. Otherwise, you must specify the width in pixels, height in pixels, and color depth in bits-per-pixel. If the image has indexed colors you should also specify the number of colors used. When manually specified, it is not checked against the file for accuracy.

FILE is the path to the picture file to be imported, or the URL if MIME type is −−>

For example, "|image/jpeg|||../cover.jpg" will embed the JPEG file at ../cover.jpg, defaulting to type 3 (front cover) and an empty description. The resolution and color info will be retrieved from the file itself.

The specification "4|−−>|CD|320x300x24/173|http://blah.blah/backcover.tiff" will embed the given URL, with type 4 (back cover), description "CD", and a manually specified resolution of 320x300, 24 bits-per-pixel, and 173 colors. The file at the URL will not be fetched; the URL itself is stored in the PICTURE metadata block.

−−sector-align
> Align encoding of multiple CD format files on sector boundaries. See the HTML documentation for more information. This option is DEPRECATED and may not exist in future versions of flac.

−−ignore-chunk-sizes
> When encoding to flac, ignore the file size headers in WAV and AIFF files to attempt to work around problems with over-sized or malformed files.
>
> WAV and AIFF files both have an unsigned 32 bit numbers in the file header which specifes the length of audio data. Since this number is unsigned 32 bits, that limits the size of a valid file to being just over 4 Gigabytes. Files larger than this are mal-formed, but should be read correctly using this option.

-S {#|X|#x|#s}, **−−seekpoint**={#|X|#x|#s}
> Include a point or points in a SEEKTABLE. Using #, a seek point at that sample number is added. Using X, a placeholder point is added at the end of a the table. Using #x, # evenly spaced seek points will be added, the first being at sample 0. Using #s, a seekpoint will be added every # seconds (# does not have to be a whole number; it can be, for example, 9.5, meaning a seekpoint every 9.5 seconds). You may use many −S options; the resulting SEEK-TABLE will be the unique-ified union of all such values. With no −S options, flac defaults to '−S 10s'. Use −−no-seektable for no SEEKTABLE. Note: '−S #x' and '−S #s' will not work if the encoder can't determine the input size before starting. Note: if you use '−S #' and # is >= samples in the input, there will be either no seek point entered (if the input size is determinable before encoding starts) or a placeholder point (if input size is not determinable).

-P #, **−−padding=#**
> Tell the encoder to write a PADDING metadata block of the given length (in bytes) after the STREAMINFO block. This is useful if you plan to tag the file later with an APPLICATION block; instead of having to rewrite the entire file later just to insert your block, you can write directly over the PADDING block. Note that the total length of the PADDING block will be 4 bytes longer than the length given because of the 4 metadata block header bytes. You can force no PADDING block at all to be written with −−no-padding. The encoder writes a PADDING block of 8192 bytes by default (or 65536 bytes if the input audio stream is more that 20 minutes long).

-T *FIELD=VALUE*, **−−tag=***FIELD=VALUE*
> Add a FLAC tag. The comment must adhere to the Vorbis comment spec; i.e. the FIELD must contain only legal characters, terminated by an 'equals' sign. Make sure to quote the comment if necessary. This option may appear more than once to add several comments. NOTE: all tags will be added to all encoded files.

−−tag-from-file=*FIELD=FILENAME*
> Like −−tag, except FILENAME is a file whose contents will be read verbatim to set the tag value. The contents will be converted to UTF-8 from the local charset. This can be used to store a cuesheet in a tag (e.g. −−tag-from-file="CUESHEET=image.cue"). Do not try to store binary data in tag fields! Use APPLICATION blocks for that.

-b #, **−−blocksize=#**

> Specify the block size in samples. Subset streams must use one of 192, 576, 1152, 2304, 4608, 256, 512, 1024, 2048, 4096 (and 8192 or 16384 if the sample rate is >48kHz).

-m, **−−mid-side**

> Try mid-side coding for each frame (stereo input only)

-M, **−−adaptive-mid-side**

> Adaptive mid-side coding for all frames (stereo input only)

-0..**-8**, **−−compression-level-0**..**−−compression-level-8**

> Fastest compression..highest compression (default is −5). These are synonyms for other options:

> > **-0**, **−−compression-level-0**
> >
> > > Synonymous with −l 0 −b 1152 −r 3

> > **-1**, **−−compression-level-1**
> >
> > > Synonymous with −l 0 −b 1152 −M −r 3

> > **-2**, **−−compression-level-2**
> >
> > > Synonymous with −l 0 −b 1152 −m −r 3

> > **-3**, **−−compression-level-3**
> >
> > > Synonymous with −l 6 −b 4096 −r 4

> > **-4**, **−−compression-level-4**
> >
> > > Synonymous with −l 8 −b 4096 −M −r 4

> > **-5**, **−−compression-level-5**
> >
> > > Synonymous with −l 8 −b 4096 −m −r 5

> > **-6**, **−−compression-level-6**
> >
> > > Synonymous with −l 8 −b 4096 −m −r 6 −A tukey(0.5) −A partial_tukey(2)

> > **-7**, **−−compression-level-7**
> >
> > > Synonymous with −l 8 −b 4096 −m −e −r 6 −A tukey(0.5) −A partial_tukey(2)

> > **-8**, **−−compression-level-8**
> >
> > > Synonymous with −l 12 −b 4096 −m −e −r 6 −A tukey(0.5) −A partial_tukey(2) −A punchout_tukey(3)

−−fast   Fastest compression. Currently synonymous with −0.

−−best   Highest compression. Currently synonymous with −8.

-e, **−−exhaustive-model-search**

> Do exhaustive model search (expensive!)

-A *function*, **−−apodization=***function*

> Window audio data with given the apodization function. The functions are: bartlett, bartlett_hann, blackman, blackman_harris_4term_92db, connes, flattop, gauss(STDDEV), hamming, hann, kaiser_bessel, nuttall, rectangle, triangle, tukey(P), partial_tukey(n[/ov[/P]]), punchout_tukey(n[/ov[/P]]), welch.

> For gauss(STDDEV), STDDEV is the standard deviation (0<STDDEV<=0.5).

> For tukey(P), P specifies the fraction of the window that is tapered (0<=P<=1; P=0 corresponds to "rectangle" and P=1 corresponds to "hann").

> For partial_tukey(n) and punchout_tukey(n), n apodization functions are added that span different parts of each block. Values of 2 to 6 seem to yield sane results. If necessary, an overlap can be specified, as can be the taper parameter, for example partial_tukey(2/0.2) or partial_tukey(2/0.2/0.5). ov should be smaller than 1 and can be negative.

Please note that P, STDDEV and ov are locale specific, so a comma as decimal separator might be required instead of a dot.

More than one −A option (up to 32) may be used. Any function that is specified erroneously is silently dropped. The encoder chooses suitable defaults in the absence of any −A options; any −A option specified replaces the default(s).

When more than one function is specified, then for every subframe the encoder will try each of them separately and choose the window that results in the smallest compressed subframe. Multiple functions can greatly increase the encoding time.

-l #, −−max-lpc-order=#
Specifies the maximum LPC order. This number must be <= 32. For Subset streams, it must be <=12 if the sample rate is <=48kHz. If 0, the encoder will not attempt generic linear prediction, and use only fixed predictors. Using fixed predictors is faster but usually results in files being 5-10% larger.

-p, −−qlp-coeff-precision-search
Do exhaustive search of LP coefficient quantization (expensive!). Overrides −q; does nothing if using −l 0

-q #, −−qlp-coeff-precision=#
Precision of the quantized linear-predictor coefficients, 0 => let encoder decide (min is 5, default is 0)

-r [#,]#, −−rice-partition-order=[#,]#
Set the [min,]max residual partition order (0..15). min defaults to 0 if unspecified. Default is −r 5.

Format Options

−−endian={*big|little*}
Set the byte order for samples

−−channels=#
Set number of channels.

−−bps=#   Set bits per sample.

−−sample-rate=#
Set sample rate (in Hz).

−−sign={*signed|unsigned*}
Set the sign of samples (the default is signed).

−−input-size=#
Specify the size of the raw input in bytes. If you are encoding raw samples from stdin, you must set this option in order to be able to use −−skip, −−until, −−cuesheet, or other options that need to know the size of the input beforehand. If the size given is greater than what is found in the input stream, the encoder will complain about an unexpected end-of-file. If the size given is less, samples will be truncated.

−−force-raw-format
Force input (when encoding) or output (when decoding) to be treated as raw samples (even if filename ends in **.wav**).

−−force-aiff-format
Force the decoder to output AIFF format. This option is not needed if the output filename (as set by −o) ends with **.aif** or **.aiff**. Also, this option has no effect when encoding since input AIFF is auto-detected.

−−force-rf64-format
Force the decoder to output RF64 format. This option is not needed if the output filename (as set by −o) ends with **.rf64**. Also, this option has no effect when encoding since input RF64 is

auto-detected.

−−force-wave64-format

Force the decoder to output Wave64 format. This option is not needed if the output filename (as set by −o) ends with **.w64**. Also, this option has no effect when encoding since input Wave64 is auto-detected.

Negative Options

−−no-adaptive-mid-side

−−no-cued-seekpoints

−−no-decode-through-errors

−−no-delete-input-file

−−no-preserve-modtime

−−no-keep-foreign-metadata

−−no-exhaustive-model-search

−−no-force

−−no-lax

−−no-mid-side

−−no-ogg

−−no-padding

−−no-qlp-coeff-prec-search

−−no-replay-gain

−−no-residual-gnuplot

−−no-residual-text

−−no-sector-align

−−no-seektable

−−no-silent

−−no-verify

−−no-warnings-as-errors

These flags can be used to invert the sense of the corresponding normal option.

SEE ALSO

metaflac(1)

The programs are documented fully by HTML format documentation, available in **/usr/share/doc/libflac-doc/html** on **Debian GNU/Linux** systems.

AUTHOR

This manual page was initially written by Matt Zimmerman mdz@debian.org for the **Debian GNU/Linux** system (but may be used by others). It has been kept up-to-date by the Xiph.org Foundation.