

NAME

opusenc – encode audio into the Opus format

SYNOPSIS

```
opusenc [ -h ] [ -V ] [ --help-picture ] [ --quiet ] [ --bitrate kbit/sec ] [ --vbr ] [ --cvbr ] [ --hard-cbr ] [ --comp complexity ] [ --framesize 2.5, 5, 10, 20, 40, 60 ] [ --expect-loss pct ] [ --downmix-mono ] [ --downmix-stereo ] [ --max-delay ms ] [ --title 'track title' ] [ --artist author ] [ --album 'album title' ] [ --genre genre ] [ --date YYYY-MM-DD ] [ --comment tag=value ] [ --picture filename|specification ] [ --padding n ] [ --discard-comments ] [ --discard-pictures ] [ --raw ] [ --raw-bits bits/sample ] [ --raw-rate Hz ] [ --raw-chan N ] [ --raw-endianness flag ] [ --ignorelength ] [ --serial serial number ] [ --save-range file ] [ --set-ctl-int ctl=value ] input.wav output.opus
```

DESCRIPTION

opusenc reads audio data in Wave, AIFF, FLAC, Ogg/FLAC, or raw PCM format and encodes it into an Ogg Opus stream. If the input file is "-" audio data is read from stdin. Likewise, if the output file is "-" the Ogg Opus stream is written to stdout.

Unless quieted **opusenc** displays fancy statistics about the encoding progress.

OPTIONS**General options**

- h, --help
Show command help
- V, --version
Show the version number
- help-picture
Show help on attaching album art
- quiet
Enable quiet mode. No messages are displayed.

Encoding options

- bitrate *N.nnn*
Set target bitrate in kbit/sec (6-256 per channel)

In VBR mode this specifies the average rate for a large and diverse collection of audio. In CVBR and Hard-CBR mode it specifies the specific output bitrate.

Default for >=44.1kHz input is 64kbps per mono stream, 96kbps per coupled pair.
- vbr
Use variable bitrate encoding (default) In VBR mode the bitrate may go up and down freely depending on the content to achieve more consistent quality.
- cvbr
Use constrained variable bitrate encoding.

Outputs to a specific bitrate. This mode is analogous to CBR in AAC/MP3 encoders and managed mode in Vorbis coders. This delivers less consistent quality than VBR mode but consistent bitrate.
- hard-cbr
Use hard constant bitrate encoding.

With hard-cbr every frame will be exactly the same size, similar to how speech codecs work. This delivers lower overall quality but is useful where bitrate changes might leak data in encrypted channels or on synchronous transports.

- comp N
Set encoding computational complexity (0-10, default: 10). Zero gives the fastest encodes but lower quality, while 10 gives the highest quality but slower encoding.
- framesize N
Set maximum frame size in milliseconds (2.5, 5, 10, 20, 40, 60, default: 20)
Smaller framesizes achieve lower latency but less quality at a given bitrate.
Sizes greater than 20ms are only interesting at fairly low bitrates.
- expect-loss N
Set expected packet loss in percent (default: 0)
- downmix-mono
Downmix to mono
- downmix-stereo
Downmix to stereo (if >2 channels input)
- max-delay N
Set maximum container delay in milliseconds (0-1000, default: 1000)

Metadata options

- title title
Set the track title comment field to *title*
- artist artist
Set the artist comment field to *artist*. This may be used multiple times to list contributing artists individually. Note that some playback software does not display multiple artists gracefully.
- album album
Set the album or collection title field to *album*
- date YYYY-MM-DD
Set the date comment field to *YYYY-MM-DD*. This may be shortened to *YYYY-MM* or *YYYY*.
- genre genre
Set the genre comment field to *genre*. This option may be specified multiple times to tag a track with multiple overlapping genres.
- comment tag=value
Add an extra comment. This may be used multiple times. The argument should be in the form "tag=value". See the vorbis-comment specification for well known tag names: <http://www.xiph.org/vorbis/doc/v-comment.html>
- picture filename|specification
Attach album art for the track.

Either a *filename* for the artwork or a more complete *specification* form can be used. The picture is added to a **METADATA_BLOCK_PICTURE** comment field similar to what is used in FLAC. The *specification* is a string whose parts are separated by | (pipe) characters. Some parts may be left empty to invoke default values. Passing a plain filename is just shorthand for the "||||filename" specification.

The format of *specification* is **[type][media-type][description][widthxheightxdepth[/colors]][filename]**

type is an optional number describing the nature of the picture. Defined values are from one of:

- 0: Other
- 1: 32x32 pixel 'file icon' (PNG only)
- 2: Other file icon

- 3: Cover (front)
- 4: Cover (back)
- 5: Leaflet page
- 6: Media (e.g., label side of a 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 colored fish
- 18: Illustration
- 19: Band/artist logotype
- 20: Publisher/studio logotype

The default is 3 (front cover). More than one `--picture` option can be specified to attach multiple pictures. There may only be one picture each of type 1 and 2 in a file.

media-type is optional. If left blank, it will be detected from the file. For best compatibility with players, use pictures with a *media-type* of `image/jpeg` or `image/png`. The *media-type* can also be `-->` to mean that *filename* is actually a URL to an image, though this use is discouraged. The file at the URL will not be fetched. The URL itself is stored in the metadata.

description is optional. The default is an empty string.

The next part specifies the resolution and color information. If the *media-type* is `image/jpeg`, `image/png`, or `image/gif`, this can usually be left empty and the information will be read 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. If possible, these are checked against the file for accuracy.

filename is the path to the picture file to be imported, or the URL if the *media-type* is `-->`.

`--padding n`

Reserve *n* extra bytes for metadata tags. This can make later tag editing more efficient. Defaults to 512.

`--discard-comments`

Don't propagate metadata tags from the input file.

`--discard-pictures`

Don't propagate pictures or art from the input file.

Input options

`--raw` Interpret input as raw PCM data without headers

`--raw-bits N`

Set bits/sample for raw input (default: 16)

`--raw-rate N`

Set sampling rate for raw input (default: 48000)

- `--raw-chan N`
Set number of channels for raw input (default: 2)
- `--raw-endianness [0/1]`
Set the endianness for raw input: 1 for big endian, 0 for little (default: 0)
- `--ignorelength`
Ignore the data length in Wave headers. Opusenc automatically ignores the length when its implausible (very small or very large) but some STDIN usage may still need this option to avoid truncation.

Diagnostic options

- `--serial n`
Force use of a specific stream serial number, rather than one that is randomly generated. This is used to make the encoder deterministic for testing and is not generally recommended.
- `--save-range file`
Save check values for every frame to a file
- `--set-ctl-int x=y`
Pass the encoder control x with value y (advanced). Preface with s: to direct the ctl to multistream s
This may be used multiple times

EXAMPLES

Simplest usage. Take input as input.wav and produce output as output.opus:

```
opusenc input.wav output.opus
```

Produce a very high quality encode with a target rate of 160kbps:

```
opusenc --bitrate 160 input.wav output.opus
```

Record and send a live stream to an Icecast HTTP streaming server using oggfwfwd:

```
arecord -c 2 -r 48000 -twav - | opusenc --bitrate 96 - - | oggfwfwd icecast.somewhere.org 8000 password/stream.opus
```

NOTES

While it is possible to use opusenc for low latency streaming (e.g. with `--max-delay` set to 0 and netcat instead of Icecast) it's not really designed for this, and the Ogg container and TCP transport aren't the best tools for that application. Shell pipelines themselves will often have high buffering. The ability to set framesizes as low as 2.5 ms in opusenc mostly exists to try out the quality of the format with low latency settings, but not really for actual low latency usage.

Interactive usage should use UDP/RTP directly.

AUTHORS

Gregory Maxwell <greg@xiph.org>

SEE ALSO

opusdec(1), **opusinfo(1)**, **oggfwfwd(1)**