

[BLOG OF STEFAAN LIPPENS](#) [TAGS](#) [ARCHIVE](#) [ABOUT](#)

Audio format conversion cheat sheet (aka how to)

BY STEFAAN LIPPENS ON 2009/12/02

TAGGED: [AUDIO](#), [OGG VORBIS](#), [FEMPEG](#), [MP3](#), [MPLAYER](#), [SOX](#),
[MUSIC](#)

In my day job, I regularly have to convert/transcode/re-encode audio data from one format to another. Because I typically have to do this in batch jobs, I'm mostly dealing with command line tools (on Linux) like [Lame](#), [SoX](#) (Sound eXchange), [MPlayer](#) and [FFmpeg](#). Having a cheat sheet of how to invoke them with the desired options has proven to be very useful, so here is mine. Note that I only cover the operations I mostly need, like format conversion, sample rate conversion, conversion to mono and trimming/cropping. If you need more/other functionality, look in the man pages or ask your favorite search engine.

Update: also see a [follow up blog post](#) about an execution time comparison between SoX, FFmpeg and MPlayer.

Audio manipulation with SoX

[SoX](#) (Sound eXchange) calls itself "the Swiss Army knife of sound processing programs" and offers, apart from standard audio format and sample rate conversion, a basic set of effects (e.g. pitch shifting, reverb, low pass filtering, flanger, etc). It's available for Linux (search for 'sox' in your package manager), Mac OS X and Windows.

```
# Minimal conversion example
sox input.mp3 output.wav

# Convert to mono (two possibilities: by specifying output format
# or with the 'channels' effect.
sox input.mp3 -c 1 output.wav
sox input.mp3 output.wav channels 1

# Change sample rate (again two possibilities)
sox input.mp3 -r 8000 output.wav
sox input.mp3 output.wav rate 8000
# Newer versions of SoX also support
sox input.mp3 output.wav rate 8k

# Trim a fragment of 30 seconds at an offset of 60 seconds
# with the 'trim' effect
sox input.mp3 output.wav trim 60 30

# All together now (trimmed fragment in mono, 22.05 Hz sample rate)
sox input.mp3 output.wav trim 60 30 channels 1 rate 22050
```

One issue with SoX is that default installs typically do not support writing MP3 files because of the patent and licensing issues with MP3. Reading MP3 files worked for me (Ubuntu 8.04 and higher) after installing the "libsox-fmt-all" package. If you're up to it, you can recompile SoX with MP3 encoding support, but there are other options if you really want MP3 encoding (see below).

Decode to WAV (from wide variety of formats) with MPlayer

[MPlayer](#) is a media player that supports a wide range of multimedia formats. It is typically used for playing video with a GUI, but can also be used (in batch mode without a GUI) to convert the audio to WAV format. MPlayer is available for Linux (package "mplayer"), Windows and Mac OS X.

The invocation bit more complex than with the other decoders shown here. For clarity, the command is spread out over several lines here (do not forget to remove the backslashes when you want it on one line):

```
# Decode the audio channel to PCM (WAV) and ignore the video channels
mplayer \
    -ao pcm:fast:waveheader:file=output.wav \
    -vo null -vc null \
    input.mp3

# Use additional audio filters (-af) to resample to 22050 Hz
# and mix down to mono.
mplayer \
    -ao pcm:fast:waveheader:file=output.wav \
    -af resample=22050,pan=1:0.5:0.5 \
    -vo null -vc null \
    input.mp3

# By default, one expects 16 bits per sample. On some setups however,
# MPlayer uses 32 bits per sample by default.
# To avoid this, set the format explicitly with:
#   -format s16le

# Pick the 30 seconds fragment at an offset of 1 minute:
mplayer \
    -ao pcm:fast:waveheader:file=output.wav \
    -vo null -vc null \
    -ss 60 -endpos 30 \
    input.mp3
```

Note: on some platforms I had to add the option `-format s16le` to make sure MPlayer encoded 16 bit PCM samples instead of 24 bit or even 32 bit, which can cause problems with some audio players/tools.

Transcode with FFmpeg (from and to a wide variety of formats)

FFmpeg is another powerful open source tool for multimedia handling like conversion/transcoding. Installing is easy with a sufficient recent Linux distribution, install the "ffmpeg" package (note: on Ubuntu 9.10 aka Karmic Koala, I also had to install "libavcodec-unstripped-52", to make MP3 encoding possible, your mileage may vary). Getting it working on Windows apparently requires you to compile it yourself (or trusting a website that provides binaries). For Mac OS X, I installed the "ffmpeg" package through MacPorts, and there is also one for Fink.

FFmpeg is typically used for video, but audio transcoding works too and is pretty simple:

```
# Minimal example: transcode from MP3 to WMA
ffmpeg -i input.mp3 output.wma

# You can get the list of supported formats with:
ffmpeg -formats

# Convert WAV to MP3, mix down to mono (use 1 audio channel),
# set bit rate to 64 kbps and sample rate to 22050 Hz
ffmpeg -i input.wav -ac 1 -ab 64000 -ar 22050 output.mp3
# Note: you can also use '-ab 64k', but I'm not sure how well this
# is supported in different version of FFmpeg

# Picking the 30 seconds fragment at an offset of 1 minute:
# In seconds
ffmpeg -i input.mp3 -ss 60 -t 30 output.wav
# In HH:MM:SS format
ffmpeg -i input.mp3 -ss 0:01:00 -t 0:00:30 output.wav
```

Encode as MP3 or re-encode an MP3 file to a different bit rate with Lame

Lame is a well known open source MP3 encoder. Installing on Linux should be easy: just look for the "lame" package. For Mac OS X, you can use the "lame" package of MacPorts or Fink. For Windows you have to compile it yourself, or trust some websites that provide binaries.

You can use it for example to encode from WAV format to MP3 or to re-encode an MP3 to a different bit rate. Some examples:

```
# Minimal example of converting a wave file to MP3
lame input.wav output.mp3

# Re-encode existing MP3 to 64 kbps MP3
lame -b 64 original.mp3 new.mp3

# More interesting options
# -m m: save as mono
# -m s: save as stereo
# -m j: save as joint stereo (exploits inter-channel correlation
#       more than regular stereo)
# -q 2: quality tweaking: the lower the value, the better the
#       quality, but the slower the algorithm. Default is 5.

# By default, lame uses constant bit rate (CBR) encoding.
# You can also use average bit rate (ABR) encoding,
# e.g. for an average bit rate of 123 kbps:
lame --abr 123 input.wav output.mp3
# or variable (VBR) encoding, e.g. between 32 kbps and 192 kbps:
lame -v -b 32 -B 192 input.wav output.mp3
```

Encode in Ogg Vorbis format

With the "oggenc" tool you can encode audio in WAV format (or raw or AIFF) to Ogg Vorbis format. On Ubuntu I had to install the "vorbis-tools" package to get "oggenc".

```
# Minimal example
oggenc audio.wav -o audio.ogg
# Setting the bit rate, downmix to mono and set the sample rate:
oggenc -b 32 --downmix --resample 22050 input.wav -o output.ogg
```

Getting information about audio files

To get basic information about an audio file (like the number of channels, sample rate, duration, etc), there is the 'soxi' tool, which is part of the sox package:

```
soxi file.mp3
```

which returns something like:

```
Input File      : 'file.mp3'
Channels       : 2
Sample Rate    : 44100
Precision      : 16-bit
Duration       : 00:03:55.35 = 10378847 samples = 17651.1 CDDA sectors
File Size      : 1.88M
Bit Rate       : 64.0k
Sample Encoding: MPEG audio (layer I, II or III)
```

You can easily specify multiple files too.

When soxi is not available (e.g. it isn't on Ubuntu 8.04) or when soxi does not recognize the file format, there are some alternatives based on FFmpeg and MPlayer.

With FFmpeg, just don't specify an output file, for example:

```
ffmpeg -i file.mp3
```

which returns something like:

```
... [version information] ...
Input #0, mp3, from file.mp3:
  Duration: 00:03:55.2, start: 0.000000, bitrate: 63 kb/s
  Stream #0.0: Audio: mp2, 44100 Hz, stereo, 64 kb/s
Must supply at least one output file
```

You can supply several files, but you need to put the flag -i in front of each one.

With MPlayer, it's a bit more involved:

```
mpplayer -vo null -ao null -frames 0 -identify file.mp3
```

which returns something like:

```
... [version information] ...
Playing file.mp3.
ID_AUDIO_ID=0
Audio file file format detected.
ID_FILENAME=file.mp3
ID_DEMUXER=audio
ID_AUDIO_FORMAT=80
ID_AUDIO_BITRATE=64000
ID_AUDIO_RATE=44100
ID_AUDIO_NCH=0
ID_LENGTH=235.00
=====
Forced audio codec: mad
Opening audio decoder: [libmad] libmad mpeg audio decoder
AUDIO: 44100 Hz, 2 ch, s16le, 64.0 kbit/4.54% (ratio: 8000->176400)
ID_AUDIO_BITRATE=64000
ID_AUDIO_RATE=44100
ID_AUDIO_NCH=2
Selected audio codec: [mad] afm: libmad (libMAD MPEG layer 1-2-3)
=====
A0: [null] 44100Hz 2ch s16le (2 bytes per sample)
ID_AUDIO_CODEC=mad
Video: no video
Starting playback...

Exiting... (End of file)
```

Further reading

- [SoX \(Sound eXchange\)](#)
- [MPlayer](#)
- [FFmpeg](#)
- [Lame](#)
- Interesting how-to on ffmpeg: <http://howto-pages.org/ffmpeg/>