Converting to .mp3 format using ffmpeg and the libmp3lame codec

*Note: These notes use **.wav** files as an example, but the source file can be pretty much anything containing an audio stream, including video files.*

The simplest command, using default settings - 128kbits/s constant bit rate (CBR)

ffmpeg -i *source.wav output.mp3*

or explicitly specifying the audio codec

ffmpeg -i *source.wav* **-acodec libmp3lame** *output.mp3*

Specifying a constant bit rate – example using 64kbit/s

ffmpeg -i *source.wav* **-b:a 64k** *output.mp3*

CBR options are: 8k, 16k, 24k, 32k, 40k, 48k, 64k, 80k, 96k, 112k, 128k, 160k, 192k, 224k, 256k, or 320k. Higher numbers give better quality but larger files.

Specifying a variable bit rate (VBR) using one of libmp3lame's quality options

ffmpeg -i *source.wav* **-q:a 4** *output.mp3*

Quality number	*Bitrate range (kbit/sec)*	*Specify as*
0	220-260	**-q:a 0**
1	190-250	**-q:a 1**
2	170-210	**-q:a 2**
3	150-195	**-q:a 3**
4	140-185	**-q:a 4**
5	120-150	**-q:a 5**
6	100-130	**-q:a 6**
7	80-120	**-q:a 7**
8	70-105	**-q:a 8**
9	45-85	**-q:a 9**

VBR encoding produces smaller files than CBR encoding for the same perceived quality. Zero gives best quality with largest file size. Higher numbers trade off audio quality for smaller file size. Using 7 produces a file that sounds much the same as 128kbit/s CBR.

Sample script to convert all .wav files in current folder

```
for file in *.wav
do
ffmpeg -i "$file" -acodec libmp3lame -q:a 4 "${file/%.wav/.mp3}"
done
```

Note: This script uses the substring replacement facility in bash to create the new filename:

${string/%substring/replacement}

If **substring** *matches <u>back</u> <u>end</u> of* **string**, *substitute* **replacement** *for* **substring**.

©Powys Linux User Group 2016 CC Attribution-NonCommercial-ShareAlike 4.0 International