Manipulating audio channels with `ffmpeg`

stereo → mono stream

Mix a single stereo stream down to a mono stream. Both channels of the stereo stream will be downmixed into the stream:

```
ffmpeg -i stereo.flac -ac 1 mono.flac
```

Note: Any out of phase stereo will cancel out.

stereo → 2 x mono files

Output each channel in stereo input to individual mono files:

```
ffmpeg -i stereo.wav -map_channel 0.0.0 left.wav -map_channel 0.0.1 right.wav
```

or with the `pan` audio filter:

```
ffmpeg -i stereo.wav -filter_complex "[0:0]pan=1|c0=c0[left];[0:0]pan=1|c0=c1[right]" -map "[left]" left.wav -map "[right]" right.wav
```

stereo → 2 + more streams

Output each channel in stereo input to individual more streams in one output file with the `channelsplit` audio filter:

```
ffmpeg -i in.mp3 -filter_complex channelsplit=channel_layout=stereo out.mka
```

Note: Your player will likely play the first stream by default unless your player allows you to select the desired stream.

mono → stereo

Create a stereo output from one mono input:

```
ffmpeg -i input.mp3 -ac 2 output.mka
```

or with the `amerge` audio filter:

```
ffmpeg -i input.mp3 -filter_complex "[0:a]lib:a]amerge=inputs=2[aout]" -map "[aout]" output.mka
```

Note: These examples will not magically create a "true" stereo output from the mono input, but simply place the same audio into both the left and right channels of the output (both channels will be identical).

2 x mono → stereo

Create a stereo output from two mono inputs with the `amerge` audio filter:

```
ffmpeg -i left.mp3 -i right.mp3 -filter_complex "[0:a][1:a]amerge=inputs=2[aout]" -map "[aout]" output.mka
```

6 x mono → 5.1

Combine 6 mono inputs into one 5.1 (6 channel) output with the `amerge` audio filter:

```
ffmpeg -i front_left.wav -i front_right.wav -i front_center.wav -i lfe.wav -i back_left.wav -i back_right.wav \
-filter_complex "[0:a][1:a][2:a][3:a][4:a][5:a]amerge=inputs=6[aout]" -map "[aout]" output.wav
```

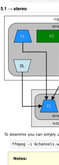
All inputs must have the same sample rate and format. If inputs do not have the same duration the output will stop with the shortest.

5.1 → 6 x mono

Split a 5.1 channel input into individual per-channel files:

```
ffmpeg -i in.wav \
-map "[FL]" front_left.wav \
-map "[FR]" front_right.wav \
-map "[FC]" front_center.wav \
-map "[LFE]" lfe.wav \
-map "[BL]" back_left.wav \
-map "[BR]" back_right.wav
```

5.1 → stereo



To downmix you can simply use `-ac 2`:

```
ffmpeg -i 6channels.wav -ac 2 stereo.wav
```

Notes:

- By default when using `-ac 2` the LFE channel is omitted. See "Digital Audio Compression Standard (Document A/52:2012)", sections 6.1.12 and 7.8 for more downmixing info.
- `ffmpeg` integrates a default down-mix (and up-mix) system that should be preferred (the `-ac` option) over the pan filter unless you have very specific needs.

If you want to map specific channels and drop the rest you can use the `pan` audio filter. This will map the `FL` (Front Left) of the input to the `FL` of the output, and the `FR` (Front Right) of the input to the `FR` of the output:

```
ffmpeg -i 6channels.wav -af "pan=stereo|c0=FL|c1=FR" stereo.wav
```

You can also map specific channels by number. This example will map the first and third channels of the input to the first and second channels of the output.

```
ffmpeg -i 6channels.wav -af "pan=stereo|c0=c0|c1=c2" output.wav
```

If the `|` in a channel specification is replaced by `+`, then the gains for that specification will be renormalized so that the total is 1, thus avoiding clipping noise. See the `pan` audio filter documentation for additional information and examples.

2 x stereo → stereo

Combine two stereo inputs into one stereo output with the `amerge` and `pan` audio filters:

```
ffmpeg -i input1.wav -i input2.wav -filter_complex "[0:a][1:a]amerge=inputs=2,pan=stereo|c0<c0+c2|c1<c1+c3[aout]" -map "[aout]" output.mp3
```

Or use `-ac 2` instead of the pan audio filter:

```
ffmpeg -i input1.wav -i input2.wav -filter_complex "[0:a][1:a]amerge=inputs=2[aout]" -map "[aout]" -ac 2 output.mp3
```

Note: The output produced with the pan audio filter may not be identical to the output produced with `-ac 2`, so you'll have to listen to your outputs or view audio statistics to determine which output suits you.

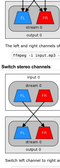
A similar situation as above, but instead use the left and right channels from the first input to make the left channel out the output, and use the left and right channels of the second input to make the right channel of the output.

Just change the channel specifications in the pan filter:

```
ffmpeg -i input1.wav -i input2.wav -filter_complex "[0:a][1:a]amerge=inputs=2,pan=stereo|c0<c0+c1|c1<c2+c3[aout]" -map "[aout]" output.mp3
```

The pan audio filter has to be used in this situation instead of `-ac 2` unlike the previous example.

Mix both stereo channels to stereo



The left and right channels of the output will each contain both the left and right channels of the input:

```
ffmpeg -i input.mp3 -af "pan=stereo|c0<c0+c1|c1<c0+c1" output.ogg
```

Switch stereo channels

Switch left channel to right and right channel to left:

```
ffmpeg -i stereo.ogg -map_channel 0.0.1 -map_channel 0.0.0 output.wav
```

or with the `pan` audio filter:

```
ffmpeg -i stereo.ogg -af pan=stereo|c0=c1|c1=c0 output.wav
```

Virtual Binaural Acoustics

`ffmpeg` can produce virtual binaural acoustics files using sofalizer filter, most known channel layouts are supported for input, output is always stereo.

```
ffmpeg -i input.wav -af sofalizer=/path/to/sofa/file output.flac
```

SOFA files can be found on http://sofacoustics.org/data/database/ari/

Make a channel

This example will mute the first channel (the left channel) but keep the second channel as is:

```
ffmpeg -i stereo.wav -map_channel -1 -map_channel 0.0.1 output.wav
```

Statistics

The `astats` audio filter can display information including length, DC offset, min/max levels, peak/RMS level dB:

```
$ ffmpeg -i input.wav -af astats -f null -
[Parsed_astats_0 @ 0x10dbe260] Channel: 1
[Parsed_astats_0 @ 0x10dbe260] DC offset: -0.001029
[Parsed_astats_0 @ 0x10dbe260] Min level: -0.605072
[Parsed_astats_0 @ 0x10dbe260] Max level: 0.605072
[Parsed_astats_0 @ 0x10dbe260] Peak level dB: -4.326430
[Parsed_astats_0 @ 0x10dbe260] RMS level dB: -20.298984
[Parsed_astats_0 @ 0x10dbe260] RMS peak dB: -12.305851
[Parsed_astats_0 @ 0x10dbe260] Noise floor dB: -144.494934
[Parsed_astats_0 @ 0x10dbe260] Crest factor: 6.281354
[Parsed_astats_0 @ 0x10dbe260] Flat factor: 0.000000
[Parsed_astats_0 @ 0x10dbe260] Peak count: 2
[Parsed_astats_0 @ 0x10dbe260] Channel: 2
[Parsed_astats_0 @ 0x10dbe260] DC offset: -0.001026
[Parsed_astats_0 @ 0x10dbe260] Min level: -0.585998
[Parsed_astats_0 @ 0x10dbe260] Max level: 0.585998
[Parsed_astats_0 @ 0x10dbe260] Peak level dB: -4.648451
[Parsed_astats_0 @ 0x10dbe260] RMS level dB: -20.519669
[Parsed_astats_0 @ 0x10dbe260] RMS peak dB: -12.526635
[Parsed_astats_0 @ 0x10dbe260] RMS trough dB: -36.784661
[Parsed_astats_0 @ 0x10dbe260] Crest factor: 6.406208
[Parsed_astats_0 @ 0x10dbe260] Flat factor: 0.000000
[Parsed_astats_0 @ 0x10dbe260] Peak count: 2
[Parsed_astats_0 @ 0x10dbe260] Overall:
[Parsed_astats_0 @ 0x10dbe260] DC offset: -0.001029
[Parsed_astats_0 @ 0x10dbe260] Min level: -0.605072
[Parsed_astats_0 @ 0x10dbe260] Max level: 0.605072
[Parsed_astats_0 @ 0x10dbe260] Peak level dB: -4.326430
[Parsed_astats_0 @ 0x10dbe260] RMS level dB: -20.408871
[Parsed_astats_0 @ 0x10dbe260] RMS peak dB: -12.416472
[Parsed_astats_0 @ 0x10dbe260] RMS trough dB: -36.784661
[Parsed_astats_0 @ 0x10dbe260] Flat factor: 0.000000
[Parsed_astats_0 @ 0x10dbe260] Peak count: 4
[Parsed_astats_0 @ 0x10dbe260] Number of samples: 1440896
```

Layouts

Output from `ffmpeg -layouts`:

```
  Individual channels:
  NAME           DESCRIPTION
  FL             front left
  FR             front right
  FC             front center
  LFE            low frequency
  BL             back left
  BR             back right
  FLC            front left-of-center
  FRC            front right-of-center
  BC             back center
  SL             side left
  SR             side right
  TC             top center
  TFL            top front left
  TFC            top front center
  TFR            top front right
  TBL            top back left
  TBC            top back center
  TBR            top back right
  DL             downmix left
  DR             downmix right
  WL             wide left
  WR             wide right
  SDL            surround direct left
  SDR            surround direct right
  LFE2           low frequency 2

  Standard channel layouts:
  NAME           DECOMPOSITION
  mono           FC
  stereo         FL+FR
  2.1            FL+FR+LFE
  3.0            FL+FR+FC
  3.0(back)      FL+FR+BC
  4.0            FL+FR+FC+BC
  quad           FL+FR+BL+BR
  quad(side)     FL+FR+SL+SR
  3.1            FL+FR+FC+LFE
  5.0            FL+FR+FC+BL+BR
  5.0(side)      FL+FR+FC+SL+SR
  4.1            FL+FR+FC+LFE+BC
  5.1            FL+FR+FC+LFE+BL+BR
  5.1(side)      FL+FR+FC+LFE+SL+SR
  6.0            FL+FR+FC+BC+SL+SR
  6.0(front)     FL+FR+FLC+FRC+SL+SR
  hexagonal      FL+FR+FC+BL+BR+BC
  6.1            FL+FR+FC+LFE+BC+SL+SR
  6.1(back)      FL+FR+FC+LFE+BL+BR+BC
  6.1(front)     FL+FR+FLC+FRC+LFE+SL+SR
  7.0            FL+FR+FC+BL+BR+SL+SR
  7.0(front)     FL+FR+FC+FLC+FRC+SL+SR
  7.1            FL+FR+FC+LFE+BL+BR+SL+SR
  7.1(side)      FL+FR+FC+LFE+BL+BR+FLC+FRC
  7.1(wide)      FL+FR+FC+LFE+BL+BR+WL+WR
  octagonal      FL+FR+FC+BL+BR+BC+SL+SR
  hexadecagonal  FL+FR+FC+BL+BR+BC+SL+SR+TFL+TFC+TFR+TBL+TBC+TBR+WL+WR
  downmix        DL+DR
```