

Linux audio recording guide (PulseAudio or PipeWire)

This article explains how to record an audio stream on a Linux system running either PulseAudio or PipeWire. (The commands below are all PulseAudio commands, but they should work on a PipeWire system thanks to its PulseAudio compatibility.)

You can record both input streams (microphones) and output streams (whatever you are hearing in your headphones). However, each stream goes to a separate file. If you want to record a conversation, record both streams and mix them afterwards in an audio editor/DAW.

First, find out the PulseAudio name of the stream you want to record.

If you want to record a microphone, run

```
pactl --format=json list sources | jq '.[[] | select(.monitor_source == "") | {name: .name, desc: .description}]'
```

Here is an example output:

```
{
  "name": "alsa_input.pci-0000_06_00.6.analog-stereo",
  "desc": "Family 17h/19h HD Audio Controller Analog Stereo"
}
{
  "name": "alsa_input.usb-Audient_EV04-00.analog-surround-40",
  "desc": "EV04 Analog Surround 4.0"
}
```

The `"name"` component is what you need; the description is just there to help you figure out which stream is the right one.

If you want to record an output (e.g. a person you're talking to), similarly run

```
pactl --format=json list sources | jq '.[[] | select(.monitor_source != "") | {name: .name, desc: .description}]'
```

Once you know the name of the stream you want to record, run

```
parecord --channels=1 --file-format=flac --device $STREAM_NAME filename.flac
```

When you are done recording, terminate the process by pressing `Ctrl-C`.

`--channels=1` forces your recording to mono, which is usually what you want.

To see the list of supported file formats, run

```
parecord --list-file-formats
```

If you try to record a generic output, such as `alsa_output.pci-0000_06_00.6.analog-stereo.monitor`, you might also capture some unwanted output from other applications or notification sounds. To make sure only a specific application is recorded, here's what you can do:

1. Create a fresh sink that will send its input to headphones, but to which no application is connected by default.
2. Direct the sound from the specific applications you want to record into this new sink.
3. Record the output of the new sink.

Find out the name of your output device by running

```
pactl --format=json list sinks | jq '[] | {name: .name, desc: .description}'
```

In my case, it says

```
{
  "name": "alsa_output.pci-0000_06_00.6.analog-stereo",
  "desc": "Family 17h/19h HD Audio Controller Analog Stereo"
}
{
  "name": "alsa_output.usb-Audient_EV04-00.analog-surround-40",
  "desc": "EVO4 Analog Surround 4.0"
}
```

Let's say my headphones are connected to EVO 4. Then I would run:

```
OUTPUT_NAME=alsa_output.usb-Audient_EV04-00.analog-surround-40
pactl load-module module-combine-sink sink_name=recording
sink_properties=device.description=Recording slaves=$OUTPUT_NAME
parecord --channels=1 --file-format=flac --device recording.monitor filename.flac
```

Now, redirect the sound to the Recording sink:

1. Run the `pavucontrol` command (a graphical window will appear) and go to the “Playback” tab.
2. Start the application you’d like to record.
3. The application should appear in `pavucontrol`. If it doesn’t, make sure the application produces some sound. Unfortunately, until the application tries to play something, PulseAudio/PipeWire cannot “see” it.
4. Choose the `Recording` sink for the application as shown on the screenshot:

[pavucontrol.png](#)

A screenshot of a pavucontrol window

Revision #1

Created 2024-04-08 21:00:46 UTC by willi

Updated 2024-11-02 15:34:58 UTC by willi