pulseaudio via Python

Why?

- Thanks to COVID-19, working from home and (too) many online meetings
- Work laptop runs Linux Mint 19.1
- Work laptop dock always reverts audio settings (input/output) back to dock after coming back from sleep
- Wasting time setting up before meetings or, when forgetting, no audio during meeting (embarassing!)

Solution?

- pulseaudio (pa): standard for audio on Linux
- Wrap pa’s command-line tool?
  - pacmd
- But wait! There is a Python library!
  - pulsectl

pulsectl

- What can we do?
  - listen for events
  - list sources and sinks (aka inputs/outputs)
  - change default source/sink
  - get/set volumes
- Let’s try some examples next...

pulsectl examples

- Connect

```
from pulsectl import Pulse
with Pulse('nzpug') as pulse:
    pass
```

your application name

pulsectl examples

- Output default source/sink

```
from pulsectl import Pulse

with Pulse('nzpug') as pulse:

    print("source", pulse.server_info().default_source_name)

    print("sink", pulse.server_info().default_sink_name)
```

pulsectl examples

- Output all source descriptions

```
from pulsectl import Pulse
with Pulse('nzpug') as pulse:
    for i in range(len(pulse.source_list())):
        s = pulse.source_list()[i]
        print(i, s.description)
```

pulsectl examples

- Output details of default source

```
from pulsectl import Pulse
with Pulse('nzpug') as pulse:
    def_source = pulse.server_info().default_source_name
    print("source", def_source)
    for s in pulse.source_list():
        if s.name == def_source:
            print("- desc:", s.description)
            if s.port_active is not None:
                print("- port desc:", s.port_active.description)
            print("- volume: ", s.volume.value_flat, s.volume.values)
```

pulsectl examples

- Setting 4th source as default source at 50% vol.

```
from pulsectl import Pulse
with Pulse('nzpug') as pulse:
    source = pulse.source_list()[4]
    pulse.default_set(source)
    source.volume.value_flat = 0.5
    pulse.volume_set(source, source.volume)
```

pulsectl examples

- Listen to events

```
from pulsectl import Pulse
with Pulse('nzpug') as pulse:
  def print_events(ev):
    print('Pulse event:', ev)

  pulse.event_mask_set('all')
  pulse.event_callback_set(print_events)
  pulse.event_listen()
```

How does that help me?

- We can change default I/O and volumes
- In other words: we all we need for profiles!
  - e.g., meeting/headphones, work/jam/stereo
- python-pulseaudio-profiles (ppp) does exactly that!
- Install via:

```
pip install python-pulseaudio-profiles
```

ppp

- Command-line tools
  - `ppp-info` – output information about system
  - `ppp-create` – create a named profile
  - `ppp-apply` – apply (= restore) a named profile
  - `ppp-rm` – delete a named profile
- Profiles get stored in:

```
$HOME/.config/python-pulseaudio-profiles
```

That’s nice, but...

- Command-line tool is still a cumbersome to use (tend to forget names of profiles, ...)
- Maybe something that’s quicker to access?
- Sure: python-pulseaudio-profiles-trayicon (pppt)

  (great naming, I know, didn’t have time/headspace)

pppt

- GTK3 Tray-icon application
- Uses pycairo and PyGObject under the hood
- Ties into ppp
- Tray-icon menu can:
  - create a profile
  - apply a profile
  - delete a profile
- Install via:

```
pip install python-pulseaudio-profiles-trayicon
```

Questions?

Thanks for listening!