Audio Setup

Set of functions to inspect the system’s audio configuration.

Note: These functions are available only if pyo is built with portaudio support.

pa_get_version

pa_get_version()

Returns the version number, as an integer, of the current portaudio installation.

```
>>> v = pa_get_version()
>>> print(v)
1899
```

pa_get_version_text

pa_get_version_text()

Returns the textual description of the current portaudio installation.

```
>>> desc = pa_get_version_text()
>>> print(desc)
PortAudio V19-devel (built Oct 8 2012 16:25:16)
```

pa_count_host_apis

pa_count_host_apis()

Returns the number of host apis found by Portaudio.

```
>>> c = pa_count_host_apis()
>>> print(c)
1
```

pa_list_host_apis

pa_list_host_apis()

Prints a list of all host apis found by Portaudio.

```
>>> pa_list_host_apis()
index: 0, id: 5, name: Core Audio, num devices: 6, default in: 0, default out: 2
```

pa_get_default_host_api

pa_get_default_host_api()

Returns the index number of Portaudio’s default host api.

```
>>> h = pa_get_default_host_api()
>>> print(h)
0
```

pa_get_default_devices_from_host

pa_get_default_devices_from_host(*host*) [source]

Returns the default input and output devices for a given audio host.

This function can greatly help finding the device indexes (especially on Windows) to give to the server in order to use to desired audio host.

Args: host: string

Name of the desired audio host. Possible hosts are:

- For Windows: mme, directsound, asio, wasapi or wdm-ks.
- For linux: alsa, oss, pulse or jack.
- For MacOS: core audio, jack or soundflower.

Return: (default_input_device, default_output_device)

pa_count_devices

pa_count_devices()

Returns the number of devices found by Portaudio.

```
>>> c = pa_count_devices()
>>> print(c)
6
```

pa_list_devices

pa_list_devices()

Prints a list of all devices found by Portaudio.

```
>>> pa_list_devices()
AUDIO devices:
0: IN, name: Built-in Microphone, host api index: 0, default sr: 44100 Hz, latency: 0.0010
```

```
1: IN, name: Built-in Input, host api index: 0, default sr: 44100 Hz, latency: 0.001088 s
2: OUT, name: Built-in Output, host api index: 0, default sr: 44100 Hz, latency: 0.001088
3: IN, name: UA-4FX, host api index: 0, default sr: 44100 Hz, latency: 0.010000 s
3: OUT, name: UA-4FX, host api index: 0, default sr: 44100 Hz, latency: 0.003061 s
4: IN, name: Soundflower (2ch), host api index: 0, default sr: 44100 Hz, latency: 0.010000
4: OUT, name: Soundflower (2ch), host api index: 0, default sr: 44100 Hz, latency: 0.00000
5: IN, name: Soundflower (16ch), host api index: 0, default sr: 44100 Hz, latency: 0.01000
5: OUT, name: Soundflower (16ch), host api index: 0, default sr: 44100 Hz, latency: 0.0000
```

pa_get_devices_infos

pa_get_devices_infos()

Returns informations about all devices found by Portaudio.

This function returns two dictionaries, one containing a dictionary for each input device and one containing a dictionary for each output device. Keys of outer dictionaries are the device index as returned by Portaudio. Keys of inner dictionaries are: ‘name’, ‘host api index’, ‘default sr’ and ‘latency’.

```
>>> inputs, outputs = pa_get_devices_infos()
>>> print('- Inputs:')
>>> for index in sorted(inputs.keys()):
...     print('  Device index:', index)
...     for key in ['name', 'host api index', 'default sr', 'latency']:
...         print('    %s:' % key, inputs[index][key])
>>> print('- Outputs:')
>>> for index in sorted(outputs.keys()):
...     print('  Device index:', index)
...     for key in ['name', 'host api index', 'default sr', 'latency']:
...         print('    %s:' % key, outputs[index][key])
```

pa_get_input_devices

pa_get_input_devices()

Returns input devices (device names, device indexes) found by Portaudio.

device names is a list of strings and *device indexes* is a list of the actual Portaudio index of each device.

```
>>> ins = pa_get_input_devices()
>>> print(ins)
(['Built-in Microphone', 'Built-in Input', 'UA-4FX', 'Soundflower (2ch)', 'Soundflower (16
```

pa_get_output_devices

pa_get_output_devices()

Returns output devices (device names, device indexes) found by Portaudio.

device names is a list of strings and *device indexes* is a list of the actual Portaudio index of each device.

```
>>> outs = pa_get_output_devices()
>>> print(outs)
(['Built-in Output', 'UA-4FX', 'Soundflower (2ch)', 'Soundflower (16ch)'], [2, 3, 4, 5])
```

pa_get_default_input

pa_get_default_input()

Returns the index number of Portaudio’s default input device.

```
>>> names, indexes = pa_get_input_devices()
>>> name = names[indexes.index(pa_get_default_input())]
>>> print(name)
'Built-in Microphone'
```

pa_get_default_output

pa_get_default_output()

Returns the index number of Portaudio’s default output device.

```
>>> names, indexes = pa_get_output_devices()
>>> name = names[indexes.index(pa_get_default_output())]
>>> print(name)
'UA-4FX'
```

pa_get_input_max_channels

pa_get_input_max_channels(*x*)

Retrieve the maximum number of input channels for the specified device.

Args: x: int

Device index as listed by Portaudio (see pa_get_input_devices).

```
>>> device = 'HDA Intel PCH: STAC92xx Analog (hw:0,0)'
>>> dev_list, dev_index =  pa_get_output_devices()
>>> dev = dev_index[dev_list.index(device)]
>>> print('Device index:', dev)
>>> maxouts = pa_get_output_max_channels(dev)
>>> maxins = pa_get_input_max_channels(dev)
>>> print('Max outputs', maxouts)
>>> print('Max inputs:', maxins)
>>> if maxouts >= 2 and maxins >= 2:
...     nchnls = 2
```

```
>>> else:
...     nchnls = 1
```

pa_get_output_max_channels

pa_get_output_max_channels(*x*)

Retrieve the maximum number of output channels for the specified device.

Args: x: int

Device index as listed by Portaudio (see pa_get_output_devices).

```
>>> device = 'HDA Intel PCH: STAC92xx Analog (hw:0,0)'
>>> dev_list, dev_index =  pa_get_output_devices()
>>> dev = dev_index[dev_list.index(device)]
>>> print('Device index:', dev)
>>> maxouts = pa_get_output_max_channels(dev)
>>> maxins = pa_get_input_max_channels(dev)
>>> print('Max outputs:', maxouts)
>>> print('Max inputs:', maxins)
>>> if maxouts >= 2 and maxins >= 2:
...     nchnls = 2
>>> else:
...     nchnls = 1
```