Configuring the audio output (especially on Windows)

Here is some tips to help you to configure the audio input/output on Windows. Some of these procedures are also valid for other systems.

How to choose the audio host api on Windows

Choosing the good audio API on Windows can turn out to be a real headache.

This document presents a script that will inspect your system and tell you if:

- Pyo can run in duplex mode. That means both audio input and output instead of output only.
- Pyo is able to connect to the different host APIs that are usually available on Windows.

In the hope that this will help you having a good experience with pyo under Windows!

https://github.com/belangeo/pyo/tree/master/scripts/win_audio_drivers_inspector.py

```
"""
Windows audio host inspector.

This script will check if pyo can run in duplex mode (both audio input and output)
and will test every host API to help the user in making his audio device choice.

"""
import sys, time
from pyo import *

if sys.platform == "win32":
    host_apis = ["mme", "directsound", "asio", "wasapi", "wdm-ks"]
else:
    print("This program must be used on a windows system! Ciao!")
    exit()


print("* Checking for any available audio input... *")

input_names, input_indexes  = pa_get_input_devices()

print("* Checking audio output hosts... *")

s = Server(duplex=0)
s.verbosity = 0

host_results = []
for host in host_apis:
    print("* Testing %s... *" % host)
    try:
        s.reinit(buffersize=1024, duplex=0, winhost=host)
        s.boot().start()
        a = Sine(freq=440, mul=0.2).out()
        time.sleep(2)
        s.stop()
        s.shutdown()
        host_results.append(True)
    except:
        host_results.append(False)

print("\nResults")
print("-------\n")

if len(input_names):
    print("Duplex mode OK!")
    print("Initialize the Server with duplex=1 as argument.\n")
else:
    print("No input available. Duplex mode should be turned off.")
    print("Initialize the Server with duplex=0 as argument.\n")

for i, host in enumerate(host_apis):
    if host_results[i]:
        print("Host: %s  ==>  OK!" % host)
    else:
        print("Host: %s  ==>  Failed..." % host)

print("Initialize the Server with the desired host given to winhost argument.")

print("\nFinished!")
```

Tunning the Windows WASAPI driver

The Windows Audio Session API (WASAPI) is Microsoft's most modern method for talking with audio devices. It is available in Windows since Vista. Pyo's default host is the WASAPI. If the script above tells you:

> Host: wasapi ==> Failed...

there is some things you can do to make it work. Open the **Sound** window by double-clicking on the volume icon and choosing *Playback Devices*. Here, select your device and click on the *Properties* button. In the *advanced* tab, make sure that the sampling rate is the same that the one used by pyo (pyo defaults to 44100 Hz). You can check the exclusive mode box if you want, this will bypass the system mixer, default settings, and typically any effects provided by the audio driver.

Perform the same in the *recording* tab if you want to run pyo in duplex mode. If you got the message:

> No input available. Duplex mode should be turned off.

you'll have to make sure first that there is an available input device in that tab.

If you use a cheap soundcard (typically, any built in soundcard is not very good!), you may have to increase the buffer size of the pyo's Server in order to avoid glitches in the audio streams.

Server initialization examples

```
# sampling rate = 44100 Hz, buffer size = 256, channels = 2, full duplex, host = WASAPI
s = Server()

# sampling rate = 48000 Hz, buffer size = 1024, channels = 2, full duplex, host = WASAPI
```

```
s = Server(sr=48000, buffersize=1024)

# sampling rate = 48000 Hz, buffer size = 512, channels = 2, output only, host = ASIO
s = Server(sr=48000, buffersize=512, duplex=0, winhost="asio")

# sampling rate = 96000 Hz, buffer size = 128, channels = 1, full duplex, host = ASIO
s = Server(sr=96000, nchnls=1, buffersize=128, duplex=1, winhost="asio")
```

Choosing a specific device

A single host API can target more than one available devices.

There is some useful functions that can help you in the choice of the audio device:

- **pa_list_host_apis()**: Prints the list of audio host APIs.
- **pa_list_devices()**: Prints the list of audio devices. The first column if the index of the device.
- **pa_get_default_input()**: Returns the index of the default input device.
- **pa_get_default_output()**: Returns the index of the default output device.
- **pa_get_default_devices_from_host(host)**: Returns the default input and output devices for a given audio host.

Run this code to see the current state of your audio setup:

```
from pyo import *

print("Audio host APIS:")
pa_list_host_apis()
pa_list_devices()
print("Default input device: %i" % pa_get_default_input())
print("Default output device: %i" % pa_get_default_output())
```

If the default device for the desired host is not the one you want, you can tell the Server which device you want to use with the *setInputDevice(x)* and *setOutputDevice(x)* methods. These methods take the index of the desired device and must be called before booting the Server. Ex:

```
from pyo import *

s = Server(duplex=0)
s.setOutputDevice(0)
s.boot()
```