

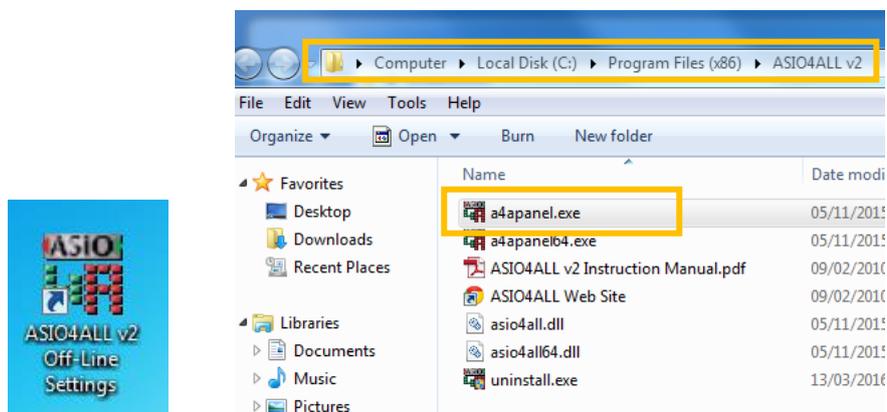
How to use ASIO in Psychopy experiments

ASIO is a driver for playing sounds with consistent time latency of 1ms (instead of 10-30ms in windows built-in driver). It is important when there's a need to synchronize sound stimuli with behavioral responses or with EEG. In order to use it with PsychoPy, there's a need to use the Psy toolbox for python, designed to play sounds.

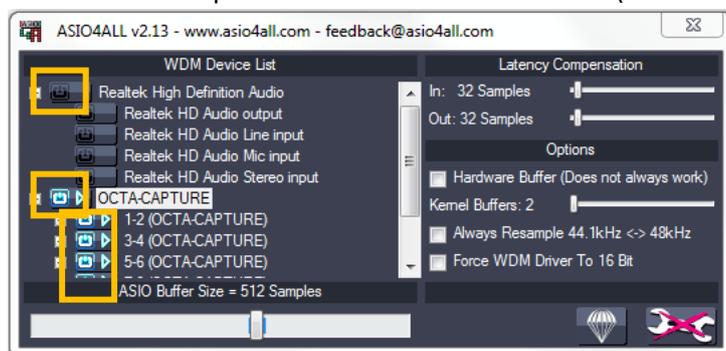
In order for ASIO to be used as the default driver for playing sounds in pyo, the settings in three places need to be handled:

I) **ASIO4All**

1. In case ASIO4ALL is not installed on the computer go to: <http://www.asio4all.com/>
2. Open ASIO4All driver through the shortcut or through 'C:\Program Files (x86)\ASIO4ALL v2'



3. Select all Octa-capture devices and cancel the rest (like Realtek for example).



II) **Octa-Capture Sound Card**

It is advisable to repeat the following procedure before each run of experiments session, to make sure that the settings haven't been changed since the last run.

4. Turn ON Octa-Capture external sound card through the desktop shortcut:



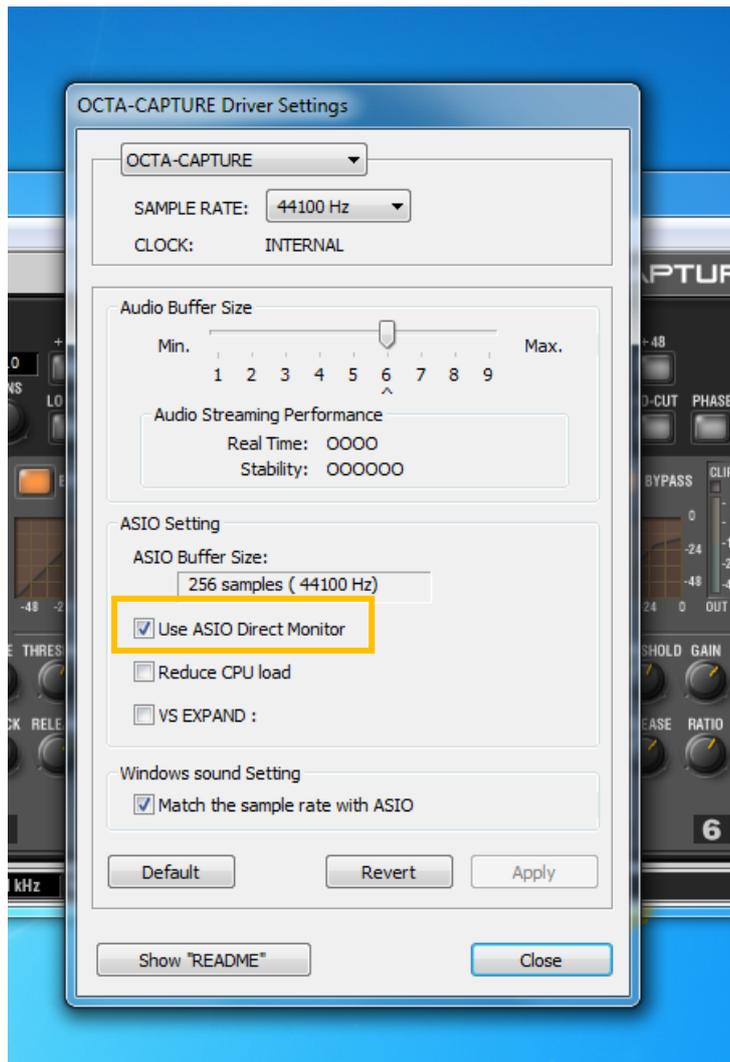
5. Go to Device → Initialize → ALL



6. Go to Device → open the patch Bay → make sure output 1-2 is "waveout 1-2"



7. Go to Driver → Driver settings → make sure that the "Use ASIO direct Monitor" is selected.



III) Your Psychopy/Python code

8. Now you can use ASIO using Pyo and Psychopy. You can write it in a code or embed the code through the builder.

- a. From the Builder: (variables that are specific are marked with yellow)

Begin Experiment

```
from pyo import *
#print pyo.pa_get_output_devices()
s=Server(duplex=0,bufferize = 256, nchnls = 8)
s.setInOutDevice(28)
```

Begin Routine

```
s=s.boot()
Speech_sound=SfPlayer(Speech_stim, speed=1, loop=False)
```

```
SquareWave = SfPlayer('singleSquareWave.wav', speed=1, loop=False) #  
square wave for the EEG synchronization. In the ASIO library
```

```
SoundMixer=Mixer(outs=8)  
SoundMixer.addInput(0,Speech_sound)  
SoundMixer.addInput(1,SquareWave)
```

```
# put speech in BOTH ears, and square wave in chan 3  
SoundMixer.setAmp(0,0,1)#(input channel, output Channel , volume)  
SoundMixer.setAmp(0,1,1)  
SoundMixer.setAmp(1,2,0.01)  
SoundMixerStatus = 0
```

Each Frame

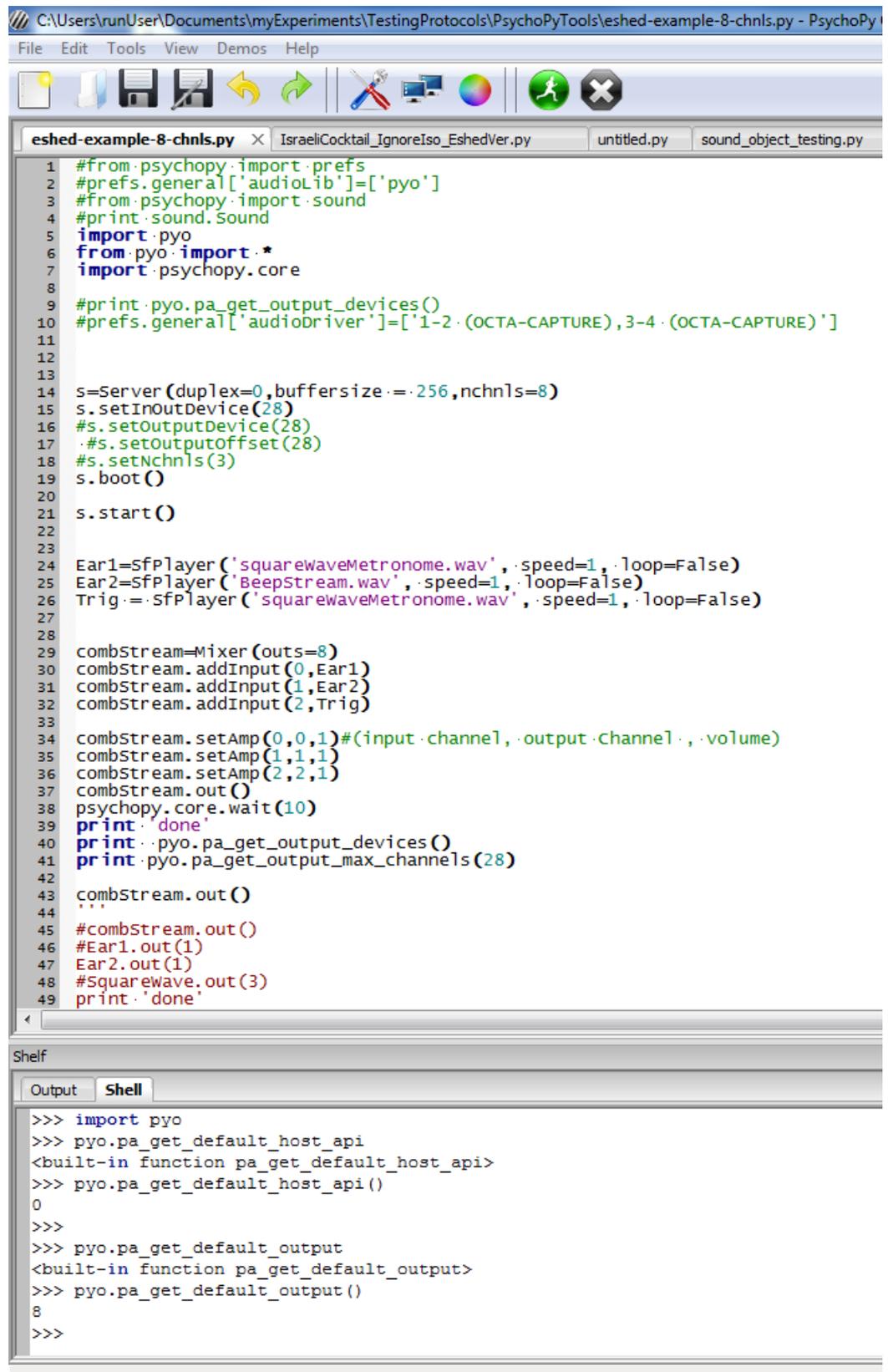
```
SoundMixer.out() # command to play sound
```

```
if t >= 0: #manipulate start time according to your experiment needs  
    if SoundMixerStatus == 0:  
        s.start()  
        #SoundMixerName.tStart = t # underestimates by a little under one  
frame  
        #SoundMixerName.frameNStart = frameN # exact frame index  
        SoundMixerStatus = 1  
        SoundMixer.out() # start the sound (it finishes automatically)  
  
    if t >= (0+ (sound_duration-win.monitorFramePeriod*0.75)): #most of one  
frame period left  
        time.sleep(0.5)  
        SoundMixer.stop() #ensure sound has stopped at end of routine  
        s.stop()  
        time.sleep(0.5)  
        s.shutdown()  
        continueRoutine = False
```

- b. Directly in the code:

Make sure to select the ASIO device in the line `s.SetInOutDevice()`. Here it is

28



```
C:\Users\runUser\Documents\myExperiments\TestingProtocols\PsychoPyTools\eshed-example-8-chnls.py - PsychoPy
File Edit Tools View Demos Help
sheshed-example-8-chnls.py x IsraeliCocktail_IgnoreIso_EshedVer.py untyped.py sound_object_testing.py
1 #from psychopy import prefs
2 #prefs.general['audioLib']=['pyo']
3 #from psychopy import sound
4 #print sound.Sound
5 import pyo
6 from pyo import *
7 import psychopy.core
8
9 #print pyo.pa_get_output_devices()
10 #prefs.general['audioDriver']=['1-2.(OCTA-CAPTURE),3-4.(OCTA-CAPTURE)']
11
12
13
14 s=Server(duplex=0,buffersize=.256,nchnls=8)
15 s.setInOutDevice(28)
16 #s.setOutputDevice(28)
17 #s.setOutputOffset(28)
18 #s.setNchnls(3)
19 s.boot()
20
21 s.start()
22
23
24 Ear1=SfPlayer('squarewaveMetronome.wav',.speed=1,.loop=False)
25 Ear2=SfPlayer('BeepStream.wav',.speed=1,.loop=False)
26 Trig.=SfPlayer('squarewaveMetronome.wav',.speed=1,.loop=False)
27
28
29 combStream=Mixer(outs=8)
30 combStream.addInput(0,Ear1)
31 combStream.addInput(1,Ear2)
32 combStream.addInput(2,Trig)
33
34 combStream.setAmp(0,0,1)#(input.channel,output.channel,.volume)
35 combStream.setAmp(1,1,1)
36 combStream.setAmp(2,2,1)
37 combStream.out()
38 psychopy.core.wait(10)
39 print 'done'
40 print pyo.pa_get_output_devices()
41 print pyo.pa_get_output_max_channels(28)
42
43 combStream.out()
44
45 #combStream.out()
46 #Ear1.out(1)
47 Ear2.out(1)
48 #Squarewave.out(3)
49 print 'done'
```

Shelf

```
Output Shell
>>> import pyo
>>> pyo.pa_get_default_host_api
<built-in function pa_get_default_host_api>
>>> pyo.pa_get_default_host_api()
0
>>>
>>> pyo.pa_get_default_output
<built-in function pa_get_default_output>
>>> pyo.pa_get_default_output()
8
>>>
```

9. If it's a new computer and you're not sure which device is ASIO you can check it by using Psychopy's python environment:
go to cmd and type:
C:\Program Files (x86)\PsychoPy2\python.exe
>>> import pyo
>>> pyo.pa_list_devices()
Now search the list to determine which device is ASIO, and use it in your code in the line setInOutDevice().

By using ASIO in Psychopy you get:

- precise and predictable latency of 2 ms.
- You won't be able to change volume via windows but only by changing your code or change the volume physically in the soundcard using the button.
- Your code won't run on any machine which is not connected to octa-capture and is not configured with ASIO (for testing comment-out the "selectInOutDevicie" line in your code).
- Pay attention that if you are going to run your experiment on another computer you must check first which device is the ASIO device by using the commands above.
- By using pyo and ASIO it is possible to send 8 different outputs, to each channel of the sound card (for triggers, wav, etc.).
- The downside of using ASIO is that you must implement the audio playing in your code instead of using the "drag and drop" Psychopy's GUI (which uses the sound.sound class).