

Basic Sound in Max - don't lose!

Electronic Music II

Spring 2014

1. DAC~ : The DAC~ object ([dac~] herein) controls the output of sound from Max/MSP.

- a. To enable sound output, the [dac~] must be turned on. To accomplish this, two objects are required.
- b. Create a blank object.



c. Type "dac~" into the object. Press enter.



d. Notice that two inlets have been created on the object box.

e. Three asides at this point:

- i. The process of object creation outlined above is almost always paraphrased, to "create a (*object name*) object," or even "create a (*object name*)."

*ii. Inlets always appear on the top side of an object box. Outlets always appear on the bottom.

iii. The two inlets of [dac~] appear by default, and correspond to a stereo output from the computer. For this demonstration we are only concerned with stereo output.

f. The second object we need is a [toggle] object. The process outlined above will also serve to do this; alternatively one can use the Object Explorer.



g. Now, position the mouse over the outlet of the [toggle]. A red circle will appear around the outlet when the mouse is properly positioned.



h. Click the mouse, and drag to the left inlet of [dac~]. When a similar red circle appears around this inlet, release the mouse button.



i. This will create a patch cord connecting these two points.

*j. The practical impact of this is that now, by switching the [toggle] on or off, we can turn on and off sound output from Max.

2. [sfplay~]: this object plays back sound files from the hard drive.

a. Create an [sfplay~] object.

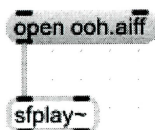


b. Notice that it has both multiple inlets and multiple outlets. These will be explained in the following items.

** c.* The left inlet allows us to control which file is open, and to start and stop playback. Message boxes are used to accomplish this. These can be created either using the Explorer, or the blank-object method laid out above.

d. Once a blank message box has been created, type into it the word “open” followed by the particular file to open. For this demonstration, the filename is “ooh.aiff.”

e. Connect the outlet of this message box to the left inlet of [sfplay~].



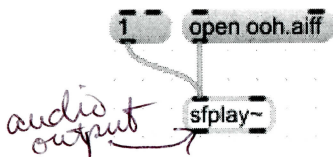
f. Troubleshooting tip: if the filename provided with the “open” message is incorrect or otherwise doesn't exist, [sfplay~] will obviously be unable to open it. Any file currently opened will remain open. When such a filename mismatch occurs, the Max Window (accessible through the Sidebar or through command-M) will display this message:

Object	Message
sfplay~	can't find file dang.aiff

If you are experiencing playback errors, check if this is occurring. *⇒ search folder for file*

g. The second message box we will create will contain only the number 1. Connect the outlet of this message box to the left inlet of [sfplay~].

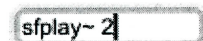
h. At this point, the patch should resemble this:



3. Connecting to [dac~]

a. The audio output of [sfplay~] is mono by default. This means that, since “ooh.aiff” is a stereo file, we will only hear the left channel.

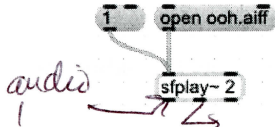
b. The number of output channels from [sfplay~] can be established with a creation argument. Create an [sfplay~] object, but before pressing enter add the number 2, as shown:



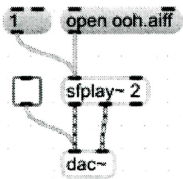
c. Then, after pressing enter an [sfplay~] object will be made, but with three outlets.

sfplay~ 2

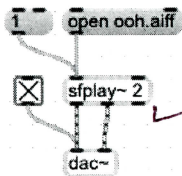
- d. Now, the **two** left outlets are the left and right channels, respectively, of any loaded audio file.
- e. Since we are in fact using a stereo file for this demonstration, I can simply add the number 2 to the pre-existing [sfplay~] shown above.



- f. Now, create patch cords connecting the left outlet of [sfplay~] to the left inlet of [dac~], and right to right, as shown below:



- g. To hear the output of [sfplay~], first click the toggle on.



Use gate not to ~~create~~ receive info from certain sound files

the bang => signal to show us when the file has ended.

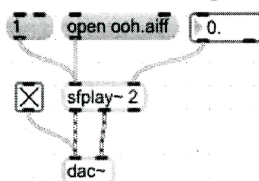
If wanting sounds to accelerate, might be commanded to play a different sound before it's ended.

- h. Second, click on the "open" message box. Then, click on the "1" message box. You will hear the file "ooh.aiff" played back over the speakers (in this particular demonstration).

- i. An aside: Before a file can be played back by [sfplay~] it must be opened. This means that, before clicking the "1" message box will cause the playback of the desired file, the "open" message has to be clicked. Once the "open" message has been clicked, it will not be necessary to click it again until a new file is desired.

4. [sfplay~] part 2

- a. The right outlet of [sfplay~] outputs a bang when the file is finished playing back.
- b. The right inlet of [sfplay~] is used to provide a speed ratio for playing back the opened audio file. This functions exactly like a varispeed pitch shift operation; sending a value of 0.5 will result in playback twice as long and lowered an octave in pitch, a value of 2 will result in playback half as long and raised an octave.
- c. The simplest way to control the playback speed is by creating a float box, and attaching the left outlet of the float box to the right inlet of [sfplay~], as shown below:



Click open file, then 1.

So, ~~you can~~ only open one file in this procedure?
That's how it seems

5. Controlling volume

a. First, some notes about digital audio.

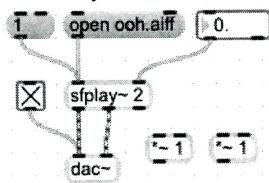
i. Each sample (recalling sample rate theory) is expressed within the computer as a value between -1 and 1. These values are translated into electrical voltage, and sent to the speaker, where they correspond to rarefaction and compression accomplished through the speaker's membrane.

ii. To manipulate volume within the computer (i.e., within Max), therefore, we can multiply the values of the digital audio by some factor.

iii. Multiplying every sample of a file by 0.5 results in the output being half as loud; multiplying by 2 results in it being twice as loud.

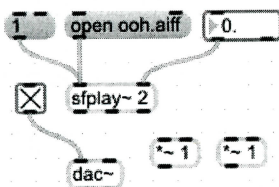
b. To accomplish this operation within Max (that is, without modifying the source audio file), the [*~] object is required. This object performs a multiplication operation on each sample of audio that passes through it.

c. Create two of these objects, and provide them with creation arguments of 1. (This is of course optional, and only mentioned so that once they are connected there will be audible output).



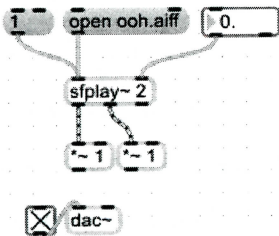
Right inlet can ~~also~~ modify sound register.

d. Now, delete the patch cords connecting [sfplay~] to [dac~].

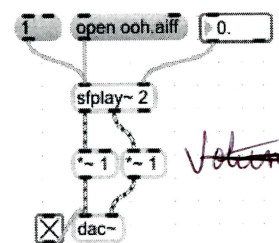


Is there concern of too many chords going to one inlet?

e. Connect the left outlet of [sfplay~] to the left inlet of one of the [*~] objects, and the right outlet to the other.



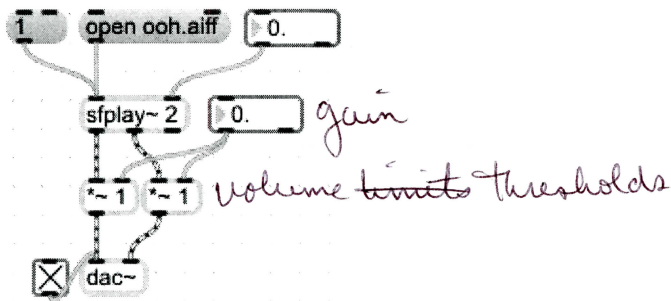
f. Now, connect the outlet of the left [*~] to the left inlet of [dac~], and right to right.



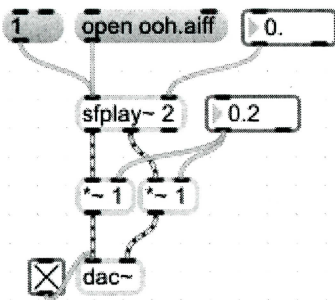
pitch

~~Volume gain~~ *Volume thresholds*

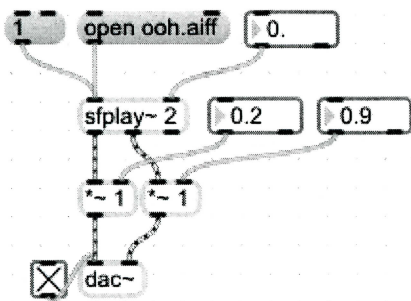
- g. For this demonstration we will control the [*~] objects using float boxes.
- h. First, create one float box. Connect the left outlet of each [*~], as shown.



- i. The value that is set in the float box will be sent to each [*~] concurrently, changing the volume of each channel by the same amount. So, in the following patch, each channel will be heard at 20% of their original volume.



- * j. To change the volume of the two channels independently, simply create another float box. Attach the left outlet of one float box to the right inlet of one [*~]. In the patch shown below, the left channel will playback at 20% original volume, while the right channel will playback at 90%.



Max patch must be in the same folder as the files.