**Introducing `sfrecord~`**

**Electronic Music II**

**Spring 2013**

A. `sfrecord~` is an object used to record audio files to the hard disk.

    1. Create a blank object and type `sfrecord~`. This will create the object, with the default single channel (mono recording).

    2. To create a stereo `sfrecord~`, set the number of channels (2) in the object:



    3. To use `sfrecord~`, we must send it the following, in order:

        a. An open message, with the filename to be created

        b. A message "samptype int24", to set the output to 24 bits

        c. A trigger to record. This will be a message containing just the number 1.

B. For demonstration, we will be using a special feature of `sfrecord~`'s open messages.

    1. Create a message box containing only the word "open". Connect the outlet to *sfrecord~*'s left inlet.
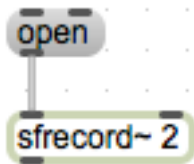


    2. Clicking this message will open a file dialog, where we can choose a location and file name for *sfrecord~*'s output.
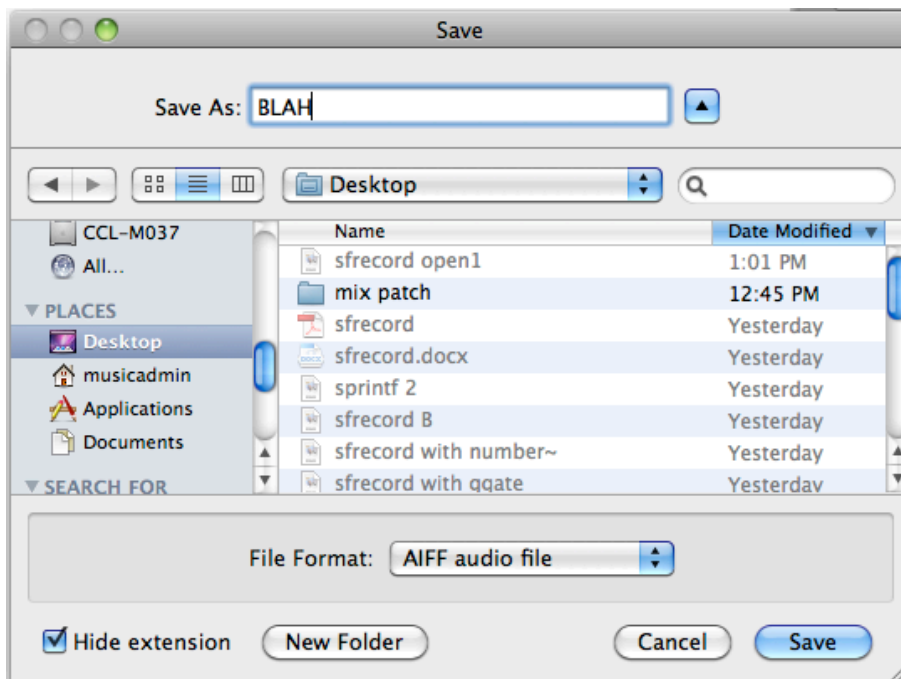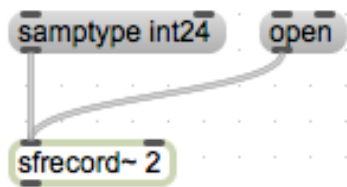


C. The message setting `sfrecord~`'s bit depth will be sent to `sfrecord~` following the open message.

    1. Create a message box containing the following: **samptype int24**. Shown below:

2. Connect the outlet of the message to the left inlet of `sfrecord~`.

```
samptype int24     open


sfrecord~ 2
```

D. The final piece of information `sfrecord~` needs is a 1 to begin recording, or a 0 to stop recording.

1. Create two message boxes, one containing the number 1 and the other the number 0. Attach their outlets to `sfrecord~`'s left inlet:

```
1   0    samptype int24    open


        sfrecord~ 2
```
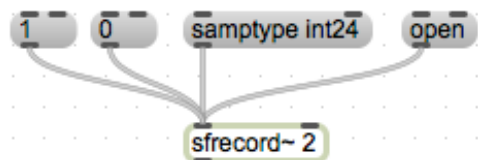
2. This is a functional basic setup.

3. One of the first improvements we can make is to ensure that we send information in the correct order. This is accomplished by using a `trigger` object:

```
t b b b


1   0    samptype int24    open


        sfrecord~ 2
```
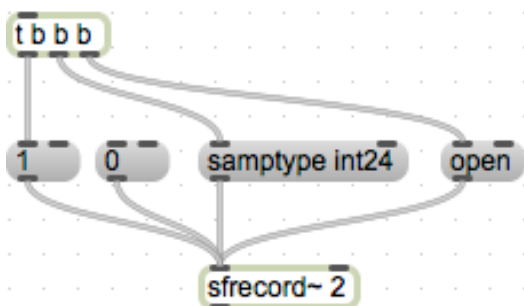
4. Now, every time we click the bang button at the top left, an open message will be sent, followed by our bit depth setting, followed by the trigger to start recording.

5. This setup still requires us to manually stop recording by clicking the 0 message. We can include this function in our trigger setup with a `ggate`:

```
○

t b b


t b b b


1   0    samptype int24    open


        sfrecord~ 2
```
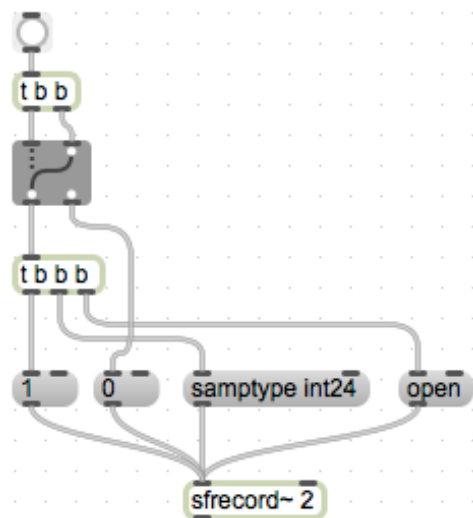
6.  Now, each time we click the bang button, we will either initiate the sequence of events to start a new recording, or stop the current recording.  We cannot send a start trigger without first stopping the recording.

E.  Closing remarks

1.  One convenient troubleshooting object for `sfrecord~` is `number~`.  This serves as a timer, giving us a readout of how many milliseconds `sfrecord~` has been recording.  Attach the outlet of `sfrecord~` to its left inlet.
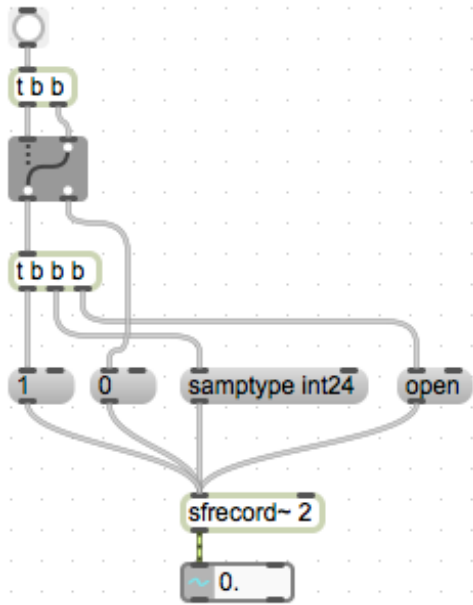


2.  Another convenient reminder-object is an integer box.  Attach the outlets of our 1 and 0 message boxes to its inlet.  This will reflect the current state of the switch.