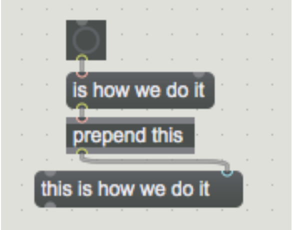
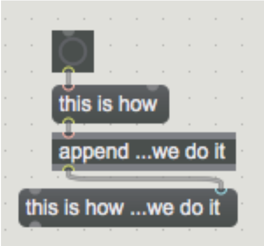
Composition: Electronic Media II
Spring 2015
How to Automatically Write Files Using [Prepend] and [Append]

1. [Prepend] and [Append] are used to add messages to existing inputs.
   a. [Prepend] adds a message to the beginning of an input:

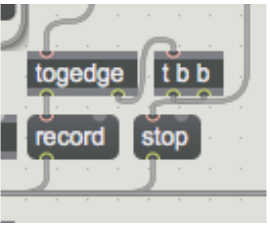


   b. [Append], as you might suspect, adds a message to the end of an input:



2. [Prepend] is used to add the message “write”, which will eventually become a command, to a unique file name. The result of this process is that a command to write (filename).mid is sent to the sequencer object.

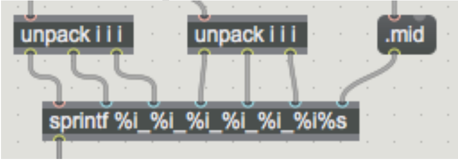
3. The process of writing the file is triggered by a [togedge] object, which routes bangs between two locations in alternating fashion.
   a. When “bang #1” reaches the [togedge] object, it is relayed to the record command.
   b. When “bang #2” reaches the [togedge] object, it is relayed to a trigger that sends two more bangs simultaneously: one tells the recording to stop, and the other starts the file-writing process.



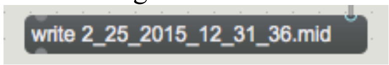
=_

4. There are three components of the file name, and they are triggered by four simultaneous bangs ([t b b b]).
   a. The first component is the date, which is formatted as (month# day# year); in order for this component to work properly down the chain, its data has to be “unpacked” into three discrete integers.
   b. The second component is the time in hour, minute, second. This data must also be unpacked into three discrete integers.
   c. The third and final component is the file extension (“.mid”).

5. All three components are combined and formatted by the [sprintf] object:



6. “Write” is then prepended to the output from [sprintf], and the whole message, shown below, is sent to the sequencer, which recognizes the command to write and creates a file in the same folder that contains the patch.



write 2_25_2015_12_31_36.mid