

## 5.1 DIGITAL SOUND

### GLOBAL AUDIO SETTINGS IN MSP

In order to manage the global audio settings and choose the audio interface that MSP will communicate with, you will need to open the Audio Status window which is found in the Options Menu (figure 5.1).

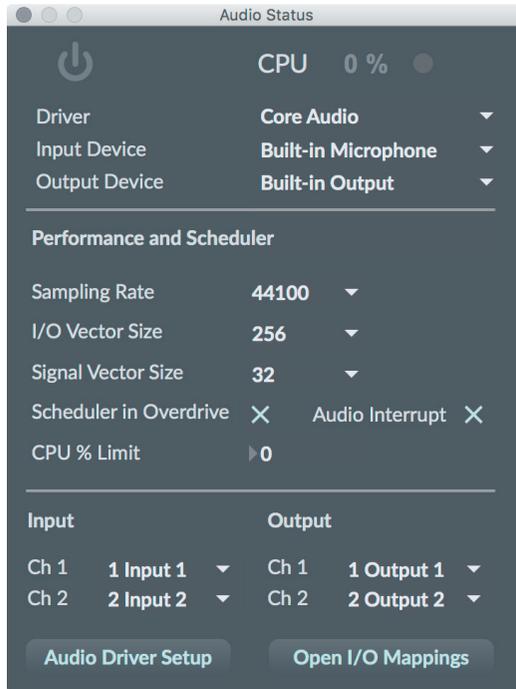


Fig. 5.1 The Audio Status window

We will now take a look at the main features of this important window. It is not necessary to memorize all the information that is described here; you can always refer back to this section each time you need information about the Audio Status window.

The window is divided into three boxed sections, each of which contains a group of related parameters. The first section primarily deals with the audio interface. Let's look at some of its main features:

In the upper part of the window, we can see, on the left, a power button which can be used to start and stop the DSP engine. On the right, we have a value which indicates the percentage of the computer's processor that MSP occupies in order to perform the active patch's audio algorithm. Obviously, the same patch will use different percentages on computers with different processing power. When a patch requires a percentage equal to or above 95% the computer will become difficult to control and will respond extremely slowly to commands; it is therefore a parameter to keep an eye on..

**Driver:** this chooses the audio driver used by MSP. It can be used either to select the driver for an audio interface connected to the computer (in the figure above, the “CoreAudio” driver of a Macintosh computer is selected), or to send the signal to an application compatible with the ReWire protocol<sup>1</sup>, by selecting the “ad\_rewire” menu option. The “NonRealTime” option can also be selected to generate a signal out of real-time, which can be very useful if the computation algorithm is too complex to be able to be executed in real-time by the computer. In this case the signal will naturally be saved to disk so you can listen to it when processing is finished. Finally, by selecting the option “Live,” users of Max For Live<sup>2</sup> can send the audio directly to the program Ableton Live.

**Input Device, Output Device:** these two parameters allow us to specify the input and output devices. Their function depends on the type of driver and audio interface used.

The second boxed section allows us to regulate the relationship between efficiency and latency for the audio processing, as well as to regulate the temporal precision of Max messages.

**Sampling Rate:** here we can specify the sampling rate. The list of sampling rates available will vary depending on the audio interface used.

**I/O Vector Size:** digital audio signals do not pass between MSP and the audio interface one sample at a time, but rather in groups of samples called *vectors*. The size in samples of the input/output vector can be set here. The smaller the vector is, the less latency (i.e., delay) there will be between input and output. On the other hand, processing each vector has a certain computational cost – this means that, when using a very small vector size, more vectors will need to be computed per second than with larger vectors, and this will increase the percentage of CPU needed to compute the signal. What’s more, a vector that is too large can create problems, in so much as MSP may not be able to calculate the signal in the time available, and this could create a click in the audio being output. We recommend using a setting no greater than 256 samples, even though the range of possible values for this setting will depend on the audio interface (some interfaces, for example, could have a minimum I/O latency of 512 samples).

**Signal Vector Size:** this parameter indicates the number of samples that the MSP *patch* itself will process at a time. Also in this case, the larger the vector is, the less the computational cost will be. The difference between this and the former vector setting is that the signal vector size will have no effect on latency, and cannot be set to values larger than that the I/O Vector Size. For certain objects (we will see which ones at the appropriate moment) there can nonetheless be useful minimum values. We recommend using a value between 16 and 128 samples.

---

<sup>1</sup> For more about the ReWire protocol, see <http://www.propellerheads.se>.

<sup>2</sup> We will talk about Max For Live at the end of this Volume.

**Scheduler in Overdrive:** this option can also be set using the Options menu. When Max is in overdrive, it gives priority to timed events (for example, bangs sent by the **metro** object) and MIDI messages that it receives, over other tasks of secondary importance, such as the refreshing the patch's graphics, or responding to mouse or computer keyboard input. This means that Max will be rhythmically more precise, but may run the risk of no longer responding to keyboard or mouse input if there are too many timed events to deal with. Nonetheless, MSP signals always have priority over Max messages.

**Audio Interrupt:** this option is available only when Max is in overdrive mode. When this option is set to *yes*, timed events are always processed immediately before calculating each signal vector. This allows us to more precisely synchronize audio with Max messages<sup>3</sup>. However, when using this option we should select a very small Signal Vector size (less than 64 samples) otherwise the Max messages are likely to be sent at times significantly different from those expected. This happens because Max must "wait" for MSP to compute its signal vector before being able to generate its timed message. For example, if the signal vector were 1024 samples long using a sampling rate of 44100 Hz, the vectors would be calculated roughly every 23 milliseconds. This means that a sequence of timed Max events would be reproduced in successive bursts every 23 milliseconds. With a signal vector of 16 samples, on the other hand, the time interval between two vectors would be less than a half a millisecond, and therefore would only create very small delays in the timing of Max events and thus absolutely imperceptible to the listener.

**CPU % Limit:** this allows you set limits on the percentage of CPU that MSP is able to use (a value equal to zero means "no limit"). This can be useful to keep the program from "taking over" all of the computer's resources.

**Input Channels, Output Channels:** these are the number of input and output channels used by the audio interface.

The four menus provided can be used to set the first two of the audio interface's input and output channels. In order to set other channels you need to click on the lower right-hand button marked "I/O Mappings." The button on the lower left, labeled "Audio Driver Setup" can be used to access the preferences for the audio interface you are using.

---

<sup>3</sup> This option is very useful in patches where the Max messages activate the production of sound in MSP, such as in the patch `IB_04_sequence.maxpat` which was discussed in Interlude B of the first volume: we recommend that you also keep it activated for the patches which will be described in successive sections of this chapter.