

# HOW TO USE TESTJAVASOUND

Version 0.3

Jacques Ferber  
[ferber@lirmm.fr](mailto:ferber@lirmm.fr)  
[www.lirmm.fr/~ferber](http://www.lirmm.fr/~ferber)  
February 2009

## Presentation

Java Sound Midi is both simple and tricky to use. Simple because the concepts are simple, and tricky because everything is based on Interfaces, and the only **MidiDevices** you can get from Java Sound is the Sequencer and the Synthesizer. But when you want to build another device, you are confronted to emptiness.

Happily, folks at Tritonus ([www.tritonius.org](http://www.tritonius.org)) tried to ease that thing. But it is still difficult to get into it, and there are no examples at [jsresources.org](http://jsresources.org) that really explain how to use devices apart from the sequencer and the synth.

That's why you will find here a very, very simple kit in order for newbies in Java Sound Midi (which I am actually) to get into it and understand it.

The module **TestJavaSound** contains two important classes: **TestJavaSound** and **MidiConnector**.

The first one contains some code to test your configuration. To launch it, type

```
java -cp testjawasound.jar midi.TestJavaSound
```

and this will show you all the devices that are known and registered on your computer: the midi in ports and midi out ports, plus the sequencer and synthesizer of Sun. For instance, on my computer (Mac OsX) here is what I got:

```
0 IN IAC Bus 1 <MIn:0>, Apple, Version 2.08 64-bit Experimental, Midi IN port:0
1 IN XStation <MIn:1>, Novation DMS, Version 2.08 64-bit Experimental, Midi IN port:1
2 OUT IAC Bus 1 <MOut:0>, Apple, Version 2.08 64-bit Experimental, Midi OUT port:0
3 OUT XStation <MOut:1>, Novation DMS, Version 2.08 64-bit Experimental, Midi OUT port:1
4 IN OUT Real Time Sequencer, Sun Microsystems, Version 1.0, Software sequencer
5 OUT Java Sound Synthesizer, Sun Microsystems, Version 1.0, Software wavetable
synthesizer and receiver
Default seq :com.sun.media.sound.RealTimeSequencer@d9dc39
Default synth :com.sun.media.sound.MixerSynth@93dcd
Max transmitters du seq :-1
Transmitters du seq :[com.sun.media.sound.RealTimeSequencer$SequencerTransmitter@dd20f6]
Viewing the transmitter of the seq :com.sun.media.sound.RealTimeSequencer
$SequencerTransmitter@9efb05
Max receivers du synth :-1
```

```
Viewing the receiver of the synth :com.sun.media.sound.MixerSynth$SynthReceiver@8814e9
Receivers du synth :[com.sun.media.sound.MixerSynth$SynthReceiver@8814e9]
Default SoundBank: com.sun.media.sound.HeadspaceSoundbank@1503a3
Instruments.. 189
```

If the program fails (has java error) it means that there are problems with your midi configuration on your machines. This happens sometimes on Linux based systems, where the midi configuration has not been setup properly. Check with your Linux distribution.

The MidiConnector program is launched directly from the jar. Simply type:

```
java -jar testjvasound.jar
```

This will launch a Java frame window which allows for connecting a midi in port to a midi out port, and play some tune to check if it works.. First, connect an input port to an output port. If you do not have any in ports in your system (for instance if you do not have an audio/midi interface) clicking on the 'connect' button will only connect the output ports.

Note that a Transposer module (which for the moment does only recopy its input (data sent to its receivers) to its output (i.e. its transmitters).

When you click the 'play' button, a little music is send to the transposer device, and if a synthesizer is connected, you should hear a sweet melody (thanks to Benjamin Bonnet).

TMidiDevice.java is part of Tritonus and DumpReceiver comes from jsresources.org.

All files are distributed with a GPL license.

## **Forthcoming features (or features that should be added)**

1. The transposer should transpose 😊
2. Add a 'piano' like device which could be used to enter notes with the mouse.
3. Add a 'panic' button to stop all notes
4. Add a midi event filter to show only specific events.
5. Add the sequencer and a midi player..