

## MuSE

# Kissed by the Muse

Now that broadband has come of age, live audio streaming from your own machine has become a genuine possibility thanks to adequate bandwidth and tools such as Icecast. The question remains as to what to feed to your Icecast Server. MuSE provides a convenient streaming source. Quickly turning your computer into the latest DJ, able to mix the audio channels to your desire.

BY CHRISTIAN PERLE

The name *MuSE* stands for *Multiple Streaming Engine*, a general tool for mixing up to six audio channels that can be either local MP3 and OGG Vorbis files or streams from other Shoutcast or Icecast Servers [2]. You can additionally mix in your soundcard's microphone input signal.

*MuSE* can either output the results locally via your soundcard, and/or send them to an Icecast Server as a stream. The server can then broadcast the stream to any attached clients.

## Do-It-Yourself

As *MuSE* provides processor specific optimization, it makes sense to compile

### OUT OF THE BOX

There are thousands of tools and utilities for Linux. "Out of the box" takes a pick of the bunch and each month suggests a little program, which we feel is either absolutely indispensable or unduly ignored.

the program from the source code. Of course, to do so you will need the C/C++ compiler *gcc*, the *gtk* library and *header files*, and the *lame* MP3 encoder. If you have installed the OGG Vorbis headers and libraries, these will also be used.

The *MuSE* source archive is available on the subscription CD or from <http://muse.dyne.org/>. A current version of *lame* (3.93.1) is available from <http://lame.sourceforge.net/>.

Follow these steps first to compile and install *lame*:

```
tar xzf lame-3.93.1.tar.gz
cd lame-3.93.1
./configure && make
su (enter root password)
make install ; strip /usr/
local/bin/lame
exit
```

The second step involves compiling *MuSE*:

```
tar xzf MuSE-0.8.tar.gz
cd MuSE-0.8
./configure --with-x && make
su (enter root password)
make install ; strip /usr/
local/bin/muse
exit
```

## No Antenna

Even if you do not use an Icecast server, *MuSE* is still worth having, as you can do a trial run before you go on air. To do so, enter *muse* & in an *xterm* or *konsole* session to launch the program.

Figure 1 shows the GUI interface for *MuSE* with four channels present, in this case.

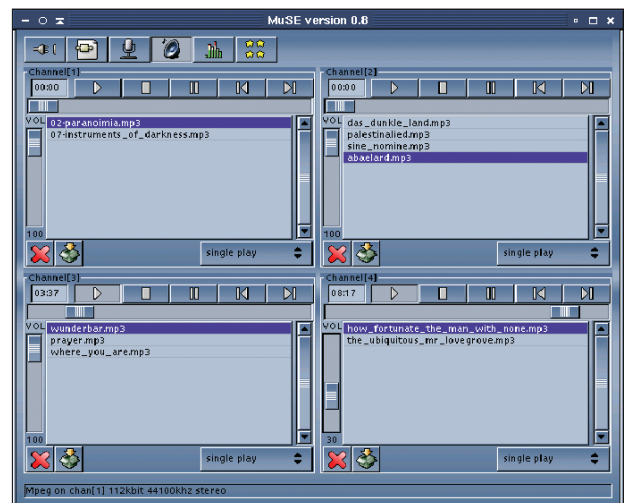


Figure 1: Four Channel MuSE

The top panel of the *MuSE* window provides six buttons with the following functions (from left to right) *Let's stream* (connect to the streaming server), *Add channel*, *Line In* (use a microphone), *Speaker* (output via local soundcard), *Vumeters*, and *Hall of Fame* (list of programmers).

*MuSE* also provides a character based console provided you specified the *--with-rubik* option in the *configure* and set the *-g ncurses* flag when launching the application. Figure 2 shows you what the *ncursesv* version looks like.

In the GUI variant each channel has its own set of controls, such as volume, play, stop and pause keys, a song position indicator and a playlist. Channels can be controlled independently. In our

### THE AUTHOR

Christian Perle currently works as a developer at *secunet Security Networks AG*.

Christian discovered Linux in 1996, after playing around with the *Sinclair ZX 81*, *Atari ST* and finally *IBM PC*. When not hacking Linux stuff he can often be found playing guitar and "Magic: The Gathering".



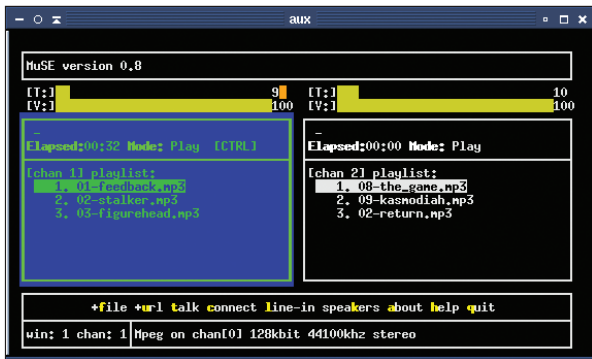


Figure 2: MuSE in character based mode

example, only channels 3 and 4 are actually playing, and the volume has been reduced to 30 per cent on channel 4.

Right clicking will display the dropdown menu for a playlist, allowing you to add new tracks to the list, or remove existing tracks. This allows you to use *MuSE* as a simple digital mixing desk with multiple playlists and cross-fading.

## Stream and Let('s) Stream

It is common to use an external Icecast server for streaming across the Internet, ensuring that the server has sufficient bandwidth to serve any attached clients. The *Let's stream* button allows you to configure access to the server; Figure 3 shows the dialog box to match.

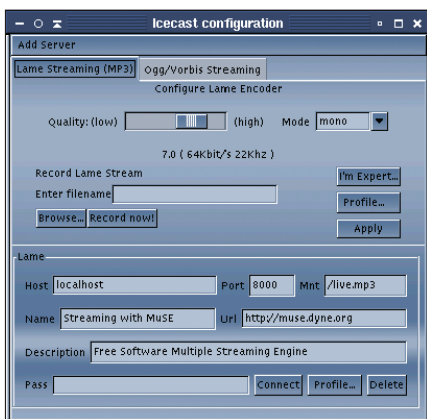


Figure 3: Configuring the server

As is usually the case, the command line provides quicker access to *MuSE*'s server preferences than the GUI dialog box. The following syntax tells *MuSE* to connect to the server at 1.2.4.8 and use **port 8000**. The MP3 stream format with a bitrate of 56 Kbit/s and a 22.05 kHz sample rate are specified. *MuSE* uses the

password 'letmein' to access the server:

```
muse -e mp3 -b 56 -r 22050 -s 1.2.4.8:8000 -p letmein &
```

After establishing the connection to the server, you can attend to the channels and playlists and finally launch the streaming output on the server.

Of course the streaming server can also be run on the server that provides the streaming source. This is why we added *icecast* to the subscription CD, to complete your broadcasting equipment.

Follow these steps to install *icecast*:

```
tar xzf icecast-1.3.12.tar.gz
cd icecast-1.3.12
./configure && make
su (enter root password)
make install
strip /usr/local/icecast/bin/icecast
ln -s /usr/local/icecast/bin/icecast /usr/local/bin/icecast
exit
```

The following command launches the local Icecast server:

```
icecast -P 8282
```

You can then change the server settings for *MuSE* to 127.0.0.1 and port 8282. Online radio listeners can now attach to your machine and, provided you have sufficient bandwidth, enjoy your selection of music. The Icecast server provides a simple console that displays status information. The console also provides a *shutdown* command that allows you to down the *icecast* server. It makes sense to use a firewall to prevent users on the Internet from connecting to your server – this will avoid copyright infringements if you happen to “broadcast” commercial songs.

## Just Checking

You can easily use a local MP3 player, such as *Xmms*, to check your streaming output. To ensure that the player has exclusive access to your soundcard, you will need to set the *-o* flag when launching *MuSE* to prevent the streaming tool from



Figure 4: XMMS listening in

accessing the soundcard.

Now select *Play location* in the main *Xmms* menu, or press [Ctrl+I]. Type “http://127.0.0.1:8282” in the text box. After a short delay while the stream is buffered, your reception test should be up and running (Figure 4).

All that remains for us to say is: Get radioactive!

## INFO

[1] Muse - Linux music editor, <http://muse.seh.de/>

[2] Icecast - Open free streaming audio, <http://www.icecast.org/>

[3] Ogg Vorbis - Audio encoding, <http://www.vorbis.com/>

## GLOSSARY

**MP3:** Abbreviation for MPEG 1 Audio Layer 3. A technique that allows CD quality audio data to be compressed to about 1/10th of their original size without audible loss of quality.

**OGG Vorbis:** The patent free alternative to MP3 [3].

**Icecast:** A system for transmitting live audio streams to one or more receivers. An MP3 or OGG signal is used as the source.

**Compiler:** When the source code of a program is “translated” (compiled), the compiler creates a binary format program that is executable by a machine from legible source code files.

**Header files:** The header or include files contain lists of the functions and parameters stored in a library. The C(++) compiler needs this information to translate the program.

Major distributions typically add the *dev* or *devel* suffix to the name of a header package for a library.

**Port:** This is where network connections “dock”. Ports are assigned numbers, many of which specify a service. Programs that bind to ports can provide services via that port, such as file transfer or remote login.