Guide to the cmusic Sources: Victor Lazzarini

cmusic – by F. Richard Moore

These are the sources and makefiles for a build of the classic CARL **cmusic** and related programs. The build has been tested on OS X and Linux, and although a number of warnings are issued, it compiles and links successfully (see 'Known Issues' below).

Some Important Notes

1. The source tree is based on the Carl-0.02 Linux distribution, but it has been pruned somewhat to remove code that is either outdated, irrelevant or won't build for some reason.

2. The build locations have been modified so that the required libraries and binaries are created locally under `./lib` and `./bin`, respectively.

3. The original m4-based makefile framework has been removed and substituted for a simpler fixed set of makefiles.

Known Issues

1. On Linux, *tosf* is not working when taking *floatsams* from a pipe.

2. On Linux, '*typein*' does not build because of '`struct sgttyb`' not being defined anywhere.

Building cmusic

These sources include some code written in **FORTRAN**. This has been adapted to be built and linked using **gfortran**. The compiler and libraries need to be present. For this, you will need to install the gfortran compiler, you can find at http://gcc.gnu.org/wiki/GFortran.

Also, added to the original *cmusic* package is a small utility called '**todac**' that is based on portaudio. For this to be built successfully, you will need to install the portaudio v.19 library (please follow the instructions in http://www.portaudio.com).

With these installed, at the toplevel directory, just type:

```
$ make
```

To install binaries and headers at /usr/local

```
$ sudo make install
```

To install elsewhere, just edit the 'PREFIX' location in the toplevel makefile.

Running cmusic

The cmusic command produces 32-bit float samples. When these are written to the console, they will appear as ASCII (text) characters. However, the most usual way of running involves redirecting the output to a file. In that case, you will get a binary stream of 32-bit floats (using the system endianess). To create a 'raw' soundfile you can use the following command, where *toot.sc* is your cmusic score (or choose one from the *examples* or *scoresWorking* directories):

```
$ cmusic toot.sc > toot.raw
```

To create a self-describing soundfile, you should pipe its output to '**tosf**' as in this example:

```
$ cmusic toot.sc | tosf -if -of -R44100 -c2 toot.irc
```

This creates a soundfile in the ~/soundfiles directory (in the top-level user directory for example: /Users/db/soundfiles). If this directory does not already exist, you must create the ~/soundfiles directory first!

To play a cmusic score directly to a soundcard, you can use the supplied utility '**todac**':

```
$ cmusic toot.sc | todac -d2 -r44100 -b1024 -c2
```

where the options are:

`-d` *N* use device number *N* (default: 0)
`-r` *R* use sampling rate *R* (default: 44100)
`-c` *N* use *N* channels of audio output (default: 1)
`-b` *N* set buffer size to *N* frames *(*default: 4096)

`-h` print a usage statement

Note that on some systems, certain devices are input-only. In this case the program will fail to open these for output and so you will need to choose another device number.