The Linux Sound Playing HOWTO

Table of Contents

Table of Contents

The Linux Sound Playing HOWTO

Yoo C. Chung, `wacko@laplace.snu.ac.kr`

v1.6, 11 August 1998

This document lists applications for Linux that play various sound formats.

1.Introduction

This is the Sound Playing HOWTO. It lists the many sound formats and the applications that can be used to play them. It also lists some hacks and advice on using these applications. There are also some other interesting applications related to sound not directly related to playback. However, this document does *not* describe how one can setup a Linux system for sound support. Refer to the Linux Sound HOWTO by Jeff Tranter for instructions on setting up a Linux system for sound support and the supported sound hardware.

This deals with normal user sound applications. That is, it is only concerned about what the average user needs to know on the application side of sound, not exotic stuff like speech synthesis, or hardware stuff which is dealt in the Sound HOWTO.

1.1 Copyright of this document

1.2 Copyright of the listed applications

If there is no mention of any copyright, then the application is under the GNU General Public License.

1.3 Where to get this document

The most recent official version of this document can be obtained from the Linux Documentation Project. The most recent unofficial version of this document can be obtained from http://laplace.snu.ac.kr/~wacko/howto/.

A Korean version of this document (very outdated) is available at http://laplace.snu.ac.kr/~wacko/howto/Sound−Playing−HOWTO.ks.

A Japanese version of this document is available at http://jf.gee.kyoto−u.ac.jp/JF/JF−ftp/euc/Sound−Playing−HOWTO.euc.

1.4 Feedback

I am not omniscient, and I don't use all the applications in here (a few I can't even try), so there are bound to be mistakes. Also, programs usually continuously evolve, so documentation tends to get out of date. Therefore, if you find anything wrong, please send me any corrections. Suggestions or additions to this document are welcome, too.

1.5 Acknowledgments

All the authors of the applications in this HOWTO. Also, Hannu Savolainen for the great sound driver and Linus Torvalds for the great underlying OS.

I'd also like to thank Raymond Nijssen (`raymond@es.ele.tue.nl`), Jeroen Rutten (`jeroen@es.ele.tue.nl`), Antonio Perez (`aperez@arrakis.es`), Ian Jackson (`ijackson@gnu.org`), and Peter Amstutz (`amstpi@freenet.tlh.fl.us`) for their information and help.

2.Playing Various Sound Formats

There are many kinds of sound formats (WAV, MIDI, MPEG etc.). Below, we list the various formats and the applications that can be used to play them.

2.1 MIDI

MIDI stands for Musical Instrument Device Interface. MIDI files usually have the extension `.mid`. They contain sequencing information, that is, information on when to play what instrument in what way, etc. Depending on your hardware (and maybe the software you use to play them), the sound might be awesome, or it might be downright crappy.

adagio

This package includes `mp` (a command−line MIDI file player) and `xmp` (an XView based MIDI file player, not to be confused with the module player also called `xmp`). You will need the SlingShot extensions to use `xmp`. It also contains other programs for playing Adagio scores.

If you have a GUS, `mp` can also play MOD files (see section Modules for more information on modules).

One little annoying bug (as of version 0.5 on some hardware) is that the sound breaks at the end. Namely, instead of ending the sound the way the MIDI file specifies, it ends by playing the note right before the last one in a long interval. It hasn't stopped me from using `mp`, but it might prevent someone from using it for `real' work. It also starts up relatively slowly.

The package does not mention any copyright (at least none that I can find), so I assume it can be freely redistributed and modified. (By a strict interpretation of copyright law, nothing gives one the right to do these things, but I somehow doubt that this was the intention of the author.)

It is a port of the CMU MIDI Toolkit to Linux (though there was enough added to make this questionable) by Greg Lee (`lee@uhunix.uhcc.hawaii.edu`).

It can be obtained from ftp://tsx−11.mit.edu/pub/linux/packages/sound/adagio05.tar.gz. The binaries included here are in a.out format (linked with ancient libraries), and the `xmp` binary segfaults in a X11R6 environment (XFree86 3.1.1, libc 4.7.2). The `mp` binary works fine in an `a.out` environment.

You will need a bit of hackery to compile it. Actually, it's not much of a hackery. All you have to do is to include the `-lfl` switch at the end of `SHROBJ` and `XMPOBJ` in the Makefile. This is to link in the `flex` library, which is not linked in by default. Then follow the installation instructions. And don't forget to have XView and the SlingShot extensions installed if you want to compile `xmp`.

TiMidity

Some people recommend this *experimental* program because of good sound quality (which is very true, it's much better than `mp` on a Sound Blaster 16, though it probably won't be much different on soundcards with wavetable synthesis like the GUS). However, it suffers from high CPU loads. It plays MIDI by first converting MIDI to WAV and then plays the WAV (you can also convert a MIDI file to a WAV file without playing if you want). This is the reason for its CPU intensive nature.

It also has an optional ncurses, SLang, Tcl/Tk or Motif interface.

You need Gravis Ultrasound patch files to use this. Look into the FAQ included with TiMidity for more information.

The author is Tuukka Toivonen (`tt@cgs.fi`).

The latest version of TiMidity can be found at the TiMidity home page. This page also contains a link to a small library of GUS patches.

playmidi

This is a MIDI player that plays to FM, GUS, and external MIDI. It is supposed to have a faster startup time compared to other MIDI players. It is also able to play Creative Music Files, Microsoft RIFF files, and large MIDI archives from games such as Ultima 7.

It has an X interface and a SVGA interface. It also has an option for real time playback with tracking all the notes on each channel and the current playback clock (included automatically with `xplaymidi` and `splaymidi`).

You should do something like

```
$ splaymidi foo.mid; stty sane
```

if you are going to use the SVGA interface, since it doesn't reset the terminal tty mode properly. The SVGA interface may be removed in the near future.

It was written by Nathan Laredo (`laredo@gnu.org` or `laredo@ix.netcom.com`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/playmidi−2.3.tar.gz.

2.2 Modules

Modules (in computer music) are digital music files, made up of a set of samples and sequencing information, telling the player when to play which sample (instrument) on which track at what pitch, optionally performing an effect, like vibrato for example.

An advantage it has over MIDI is that it can include almost any kind of sound (including human voices). Another is that it sounds just about the same on any platform, because the samples are in the module. A disadvantage it has is that it has a much larger file size compared to MIDI. Another one is that it has no real standard format (the only `real' one is the ProTracker, which many modules aren't quite compatible with). It originated on the Amiga.

The most common format has the extension `.mod`. There are many other extensions depending on what format they are in.

tracker

This very portable program (it has been ported to many platforms) plays Soundtracker and Protracker music modules. It uses 16 bit stereo output, and I consider the quality to be very good. If you need a simple way to reduce CPU load use the `−mono` option.

This is a giftware program (quoting the author). It is by Marc Espie (`Marc.Espie@ens.fr`).

A version of this with the Makefile already tweaked for Linux can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/tracker−4.3−linux.tar.gz.

gmod

This is a music module player for the Gravis Ultrasound card. 4/6/8 channel MOD, 8 channel 669, MultiTracker (MTM), UltraTracker (ULT), FastTracker (XM), and ScreamTracker III (S3M) are the supported formats.

It requires a version 3.0 or later sound driver. And a GUS, of course. You may need to modify the kernel to make volume control work the way you want.

This has an X interface. It uses the QT toolkit (needs version 0.99 or greater). Check the QT toolkit homepage for information on QT.

This can be freely distributed. It was originally written by Hannu Savolainen, and now maintained by Andrew J. Robinson (`robinson@cnj.digex.net`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/gmod−3.1.tar.gz.

MikMod

This portable module player plays XM, ULT, STM, S3M, MTM, MOD and UNI formats. (The UNI format is an internal format used by MikMod.) It has support for zipped module files. It uses 16 bit stereo for the sound output. Use the `−m` option (for mono output) if you need a simple way to lower the CPU load.

The Unix version can either use ncurses or Tcl/Tk for its interface. It can also be used as a library, not just an independent program.

It was originally written by Jean−Paul Mikkers (`mikmak@via.nl`). It is now maintained by Jake Stine (`dracoirs@epix.net`). This is shareware that has to be registered if you want to use it commercially. You also need permission to redistribute it commercially (non−commercial redistribution does not need such permission).

This can be found at the MikMod home page.

xmp

This is a module player (not to be confused with Adagio's `xmp`) which can play MOD, S3M, MTM, PTM, PTR, STM, 669, and XM modules (other formats are also supported, but still experimental or incomplete). If you have soundcards with wavetable synthesis (GUS or SoundBlaster 32AWE), then you can use this feature of the soundcard to lower the load on the CPU. It also supports compressed modules.

An X frontend to `xmp` is also available.

This was written by Claudio Matsuoka (`claudio@pos.inf.ufpr.br`) and Hipolito Carraro Jr.

This can found at the xmp home page.

s3mod

This plays 4/6/8 track MOD modules and Scream Tracker 3 modules. It uses 8 bit mono output with a sampling rate of 22000 Hz by default. You can use the option `−s` to enable stereo, `−b` to enable 16 bit output, and `−f` to set the sampling frequency. However, the sound output is worse than tracker (some noise), so I recommend using `tracker` instead of `s3mod` for playing ordinary MOD files (unless you have an underpowered machine). It has a much smaller CPU load compared to tracker.

It is copyrighted by Daniel Marks and David Jeske (`jeske@uiuc.edu`), but you can do anything you want with it (except that you can't claim you wrote it).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/s3mod−v1.09.tar.gz.

mod

This *beta* program plays MODs (15/31−instrument, up to 32 voices), MTMs, ULTs and S3Ms on the Gravis Ultrasound card. It can also use packed modules if you have `gzip`, `lharc`, `unzip`, and `unarj` installed. It cannot play Powerpacked modules or modules packed with some Amiga composers ("PACK" signature).

This requires at least version 3.0 of the sound driver. It *won't* work with the 2.90−2 or earlier version of the sound driver. The text interface requires ncurses. There is also an X interface included, which uses Tcl/Tk.

It was written by Mikael Nordqvist (`mech@df.lth.se` or `d91mn@efd.lth.se`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/mod−v0.81.tgz.

nspmod

This is an *alpha* module player which can play MTM, S3M, and MOD modules. It is intended to be a module player for soundcards without a DSP (not to be confused with what Creative Labs calls a DSP). It has a CPU load somewhat similar compared to `tracker`.

It has a feature which lets modules loop if they want to. The number of loops can be limited by the `−l` option. It uses only 8 bit sound output (as of version 0.1).

This was written by Toru Egashira (`toru@jms.jeton.or.jp`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/nspmod−0.1.tar.gz.

yampmod

This *alpha* program was designed to play 4−channel modules using the minimum of CPU resources. It was *not* designed to produce high quality sound. So the only sound output it produces is 22 kHz mono output. Also, the output isn't as clean as it should be, reflecting its alpha status.

It was written by David Groves (`djg@djghome.demon.co.uk`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/yampmod−0.1.tar.gz.

2.3 MPEG audio streams

MPEG is a standard specifying the coding of video and the associated audio for digital storage. MPEG is usually associated with video, but the audio part of the standard can be used separately. The audio part of the MPEG standard defines three layers, layer I, II, and III. Players that can decode higher layers can also decode lower layers (e.g. layer III players can play layer II files). Layer I MPEG audio files usually have the extension `.mpg` (so if there is a file with this extension that can't be played by a MPEG video player, it's probably an audio stream), layer II usually have the extension `.mp2`, and layer III usually have the extension `.mp3`. The audio compression is pretty good. A two megabyte layer II MPEG audio file will probably take up 25 megabytes for a raw PCM sample file with the same quality.

mpg123

This *beta* program is an efficient MPEG audio stream player, which has support for layers I, II, and III. It is based on code from many sources. It is able to play in real time streams that are read by HTTP (i.e. one can play an MPEG audio stream directly over the World Wide Web).

The main author is Michael Hipp (`Michael.Hipp@student.uni−tuebingen.de`). It may be used and distributed in unmodified form freely for non−commercial purposes. Inclusion in a collection of free software (such as CD−ROM images of FTP servers) is explicitly allowed.

The latest version can be obtained from the mpg123 homepage.

maplay 1.2

This MPEG audio stream player only has support for layer I and layer II streams, and lacks support for layer III streams. It supports 16 bit sound cards on Linux.

It is pretty CPU intensive, taking up to about 55% CPU time on a 60MHz Pentium. The output is intolerable on a 66MHz 486 because the CPU just can't catch up with the sound. If this happens to you, try playing only one side of the audio stream (with the `−l` or `−r` option), instead of the default stereo.

A slight change in one of the files may be necessary in order to compile it. Namely, you may need to add the following line to the beginning of the file `configuration.sh`.

```
#! /bin/sh
```

The author is Tobias Bading (`bading@cs.tu-berlin.de`). `maplay` 1.2 can be obtained from ftp://ftp.cs.tu−berlin.de/pub/misc/maplay1.2/maplay1_2.tar.

maplay 1.3b

This is an unofficial modification (i.e. not by the original author) of `maplay` 1.2, so that it can run with a much lower load on the CPU. It accomplishes this mainly by making u−law output actually work on other platforms besides the SPARC. Note that it uses u−law output by default, so the sound quality is lower.

The modifications were made by Orlando Andico (`orly@gibson.eee.upd.edu.ph`).

This can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/maplay−1.3b−Linux.tar.gz.

maplay3

This is another derivative of `maplay` 1.2. It adds support for MPEG Layer 3 audio streams. Currently it seems to have some bugs in its playback (you may hear some screeching noises). You may have to twiddle with the options to solve this.

The modifications were made by Timo Jantunen (`timo.jantunen@hut.fi` or `jeti@cc.hut.fi`). It says that it can be used freely, but making money off of it is not allowed. However, I'm not entirely sure about the validity of this copyright, since the original `maplay` is under the GNU General Public License, which does not allow derivative works to have a different copyright.

This can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/maplay3.tar.gz.

splay

This *beta* player is another derivative of `maplay` 1.2 (actually, it is a derivative of `maplay` 1.2+, which is a MS Windows only derivative of `maplay` 1.2). It adds support for MPEG Layer 3 audio streams. It is also able to play WAV files. It can also play audio streams received over an HTTP connection.

Another feature of `splay` is that it can be used as a library (under the LGPL), so that it can be used in other programs. It also tries to improve performance by using threading (you need `pthread` to use this feature) and a little inline assembly.

`splay` uses a command line interface and an optional X interface (which uses QT).

If after compiling it doesn't work (e.g. it segmentation faults), try compiling it again without threading.

This is by Jung Woo−jae (`jwj95@eve.kaist.ac.kr`).

It can be obtained from splay's home page.

Sajber Jukebox

This program is a MPEG audio player with a graphical user interface. It is based on `splay`, so it includes support for MPEG audio layers up to III. It is also able to play MPEG audio streams in real time with the stream being fed by HTTP. It is also easy to configure.

It uses the QT toolkit (at least version 1.2 is required). It also uses the LinuxThreads library (the included binary only works with version 0.5).

The author is Joel Lindholm (`wizball@kewl.campus.luth.se`).

The latest version can be obtained from ftp://kewl.campus.luth.se/pub/jukebox.

amp

This *beta* MPEG audio player only has support for MPEG Layer 3 audio streams. It is able to play directly to the soundcard, and it can output to raw PCM or WAV files. This also gives quite a load on the CPU (about 60% on a 133MHz Pentium).

This was written by Tomislav Uzelac (`tuzelac@rasip.fer.hr`). It can be freely used and distributed, as long as it is not sold commercially without permission (including it in CD−ROMs that contain free software is explicitly permitted, though).

It can be obtained from ftp://ftp.rasip.fer.hr/pub/mpeg/amp−0.7.3.tgz.

XAudio

This *alpha* library was written to be a fast implementation of an MPEG audio decoding library to be used by various GUI front−ends. It supports MPEG audio layers I, II, and III. It is capable of random access to bitstreams. A command−line interface is included. A Motif (Lesstif) front−end is also included in the Linux version.

This is by Gilles Boccon−Gibod, Alain Jobart and others. The front−ends to the libary can be freely downloaded. The library itself must be licensed to be used (a source and binary license is available).

The front−ends to the library can be obtained from the XAudio home page.

Layer 3 Shareware Encoder/Decoder

This is actually a converter that converts MPEG Layer 3 audio streams to WAV, AIFF, SND, AIFC, or just raw PCM sample files. The Linux version does not directly output the sound to the soundcard. One has to first convert it to some other format.

However, when you try to play a converted file using `sox`, you'll probably just get noise because the word order in the PCM samples is not right (at least on Intel platforms). You need to give `sox` the option `−x` to solve this problem. But there are some players that don't have to be told that the word order is wrong, so you might not have to worry about this.

If you have a really fast computer (probably at least a 100Mhz Pentium), then you can try to play MPEG Layer 3 streams directly without having to first convert the audio file to another format like in the following example (this example assumes that you're using `sox` and playing a 44.1 kHz stereo sample).

```
$ l3dec foo.mp3 −sto | play −t raw −x −u −w −c 2 −r 44100 −
```

The number after `−r` is the sample rate of the audio stream, and the number after `−c` depends on whether it is mono or stereo (or even quad). If this looks too complicated, you can use something like a shell script or an alias.

This is shareware copyrighted by Fraunhofer−IIS. A demo version for Linux on x86 systems can be obtained from ftp://ftp.fhg.de/pub/layer3. The demo version only converts layer III audio streams.

X11Amp

This *beta* software is an MPEG audio stream player with a graphical interface, similar to the one used by the Windows program `winamp`.

There is no copyright mentioned anywhere (I assume that it can be freely used for personal use). It is maintained by Mikael Alm (`psy@x11amp.bz.nu`), Thomas Nilsson (`fatal@x11amp.bz.nu`), and Olle Hallnas (`crocodile@x11amp.bz.nu`).

It can be obtained from X11Amp's homepage. Only binaries for Intel Linux and FreeBSD are made available here.

2.4 WAV

Quote from the `sox` man page:

> These appear to be very similar to IFF files, but not the same. They are the native sound file format of Windows 3.1. Obviously, Windows 3.1 is of such incredible importance to the computer industry that it just had to have its own sound file format.

These usually have the extension `.wav`.

Also see section sox and bplay for other WAV players besides the ones listed here.

wavplay

This program supports playing and recording with the WAV format. It uses locking so that only one sound may be played at a time. Its locking capabilities can also be used separately from its sound playing capabilities.

In addition to a command−line interface, it also has a Motif interface, which can be used with Lesstif.

It was originally written by Andre Fuechsel (`af1@irz.inf.tu-dresden.de`), but was evolved to the point of being completely rewritten by Warren W. Gay (`bx249@freenet.toronto.on.ca` or `wwg@ica.net`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/wavplay−1.0.tar.gz.

2.5 Other stuff

This section lists stuff that play sound formats that don't deserve a separate section (i.e. formats that have only one player available), or players that play more than one format.

sox

This program is actually a converter, that is, it converts one sound format to another. However, some versions of `sox`, when invoked as `play`, plays the sound (the `play` application in the Sound HOWTO probably refers to this). It supports raw (no header) binary and textual data, IRCAM Sound Files, Sound Blaster `.voc`, SPARC `.au` (w/header), Mac HCOM, PC/DOS `.sou`, Sndtool, and Sounder, NeXT `.snd`, Windows 3.1 RIFF/WAV, Turtle Beach `.smp`, CD−R, and Apple/SGI AIFF and 8SVX formats

Since somewhere in the 1.3.6x kernels, you might have to make a small change in one file to make it play the sound directly. Namely, you may have to change line 179 in `sbdsp.c` from

```
        if (abuf_size < 4096 || abuf_size > 65536) {
```

to

```
        if (abuf_size < 1 || abuf_size > 65536) {
```

But then again, you may not have to do this. But doing this won't break anything.

It is written and copyrighted by many people, and can be used for any purpose.

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/convert/Lsox−linux.tar.gz.

A more recent version by Chris Bagwell (`cbagwell@sprynet.com`) (based on the latest gamma version of the original `sox`, and includes the above fix) can be obtained from http://home.sprynet.com/sprynet/cbagwell/projects.html. In addition, this version supports MS ADPCM and IMA ADPCM WAV formats.

bplay

This *beta* program plays raw audio, WAV, and VOC files. It's also able to record to these files. It uses a variety of techniques to get the highest speed possible so that it can run acceptably even on slow machines. One of these techniques require that the installed programs be setuid root. The paranoid hoping to use this may want to use the Debian package by Ian Jackson (`ijackson@gnu.org`), which disables the feature that needs the setuid bit.

The author is David Monro (`davidm@gh.cs.usyd.edu.au`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/bplay−0.96.tar.gz.

SIDPLAY

This program emulates the Sound Interface Device chip (MOS 6581, commonly called SID) and the Micro Processor Unit (MOS 6510) of the Commodore 64. Therefore it is able to load and execute C64 machine code programs which produce music or sound. In general these are independent fragments of code and data which have been ripped from games and demonstration programs and have been transferred directly from the

C64.

It uses a command line interface by default. There are also Tk and QT interfaces available separately from the main package.

It is maintained by Michael Schwendt (`sidplay@geocities.com`).

It can be obtained from SIDPLAY's home page.

RealAudio Player

This lets you listen to sound, which is stored in a proprietary format, in real time over the Internet without downloading the whole sound file first. It could be used stand alone, but it is really intended to be used along with a web browser (the explicitly supported ones are Mosaic and Netscape). It cannot be used without X (you wouldn't be able to get it working with Lynx in a text console).

However, there exists a hack which allows one to run the RealAudio player from the text console. It requires the X virtual frame buffer (Xvfb) server to work. This hack can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/traplayer−0.5.tar.gz.

This is by Progressive Networks, Inc. This cannot be redistributed, modified etc. Look at the license for exact details on what you can do. It can be obtained by registering with no cost at the RealAudio home page.

cat

One might think what `cat`, the sometimes overused concatenating utility, has to do with playing sounds. I'll show a use of it through an example.

```
$ cat sample.voc > /dev/dsp
$ cat sample.wav > /dev/dsp
$ cat sample.au > /dev/audio
```

Doing a `cat` of an `.au` file to `/dev/audio` will usually work, and if you're lucky enough that the file has the correct byte order (for your platform) etc., a `cat` of a sound file that uses PCM samples (like `.wav` or `.voc`) to `/dev/dsp` might even sound right.

This isn't a totally useless use of `cat`. It might be useful, for example, if you have a sound file that none of your programs recognize, and you know that it uses PCM samples, then you might be able to get a very approximate idea on how it sounds like this way (if you're lucky).

3.Other useful sound utilities

This section has nothing to do with the actual playing of sound files. Rather, it is a collection of some sound utilities that one might find useful.

3.1 volume

This is a simple command line interface for controlling the volume (what else could it be?). It also has a separate program with a Tcl/Tk interface included in the package for controlling the volume and playing `.au` sound files. A very simple Tcl/Tk CD player is also included.

This is Freeware and it is written by Sam Lantinga (`slouken@cs.ucdavis.edu`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/soundcard/volume−2.1.tar.gz.

3.2 Sound Studio

This is a Tcl/Tk application that supports playback, recording, and editing of digital sound using `sox`. It includes `sox` in the distribution to avoid compatibility problems.

This was written by Paul Sharpe and N. J. Bailey (`N.J.Bailey@leeds.ac.uk`). It may be freely used and redistributed if a postcard is sent.

It can be found at Sound Studio's home page.

3.3 Tickle Music

This *beta* Tcl/Tk program is a music file browser that allows you to play various sound formats as long as an appropriate program to play it is on your system. By default `gmod` is used for playing MOD files and `mp` for playing MIDI files (you can change the source to use other programs).

It is written and copyrighted by Shannon Hendrix (`shendrix@pcs.cnu.edu` or `shendrix@escape.widomaker.com`).

It can be obtained from ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/tmusic−1.0.tar.gz.

4.References

1. The documentation included with the applications in this document.
2. The Linux Sound HOWTO. It can be found at the Linux Documentation Project.
3. Linux MIDI and Sound Applications
4. Programmer's Guide to OSS
5. SoX home page