EsounD

The Enlightened Sound Daemon

Eric 'Ricdude Mitchell

EsounD: The Enlightened Sound Daemon
by Eric 'Ricdude Mitchell

Table of Contents

Chapter 1. Introduction

So you’ve got your (insert sound playing program here) cruising at full tilt, and you want to check out this cool Monty Python quote, but you don’t want to kill the original sound? This is just the fix, with a little tweaking, of course. =P

The Enlightened Sound Daemon can mix several audio streams into one sound device. It will mix in pre-loaded samples, too. Want to play a frightening sound whenever the user presses the "go" button? No problem, just cache it, and play it back by sample id number.

Caveats

This is hot off the presses software, get it while it’s still hot. It’s still got a ways to go yet. Be prepared for a bumpy ride. Any and all reasonable patches and improvements accepted. I expect the communication protocol to change at least once more before it hits 1.0.

Requirements

The Enlightened Sound Daemon requires the following:

Linux, version 2.??? or higher. I run 2.0.34-RH5.1, but I don’t think the sound API has significantly changed recently. If you find it to be incompatible with different kernel/OSS revs, let me know.

And of course, a sound device.

Currently, this only runs under Linux. Everything except recording seems to work ok on SGI/Irix soon. Reports indicate it works ok on Solaris. Ports to other platforms are requested. Basic drivers for other platforms are in the distribution, but may need work from what’s in the driver_platform.c files.

Chapter 2. Downloading, Compiling, and Installing

Downloading

The Enlightened Sound Daemon is available for download from

- Enlightenment.org public ftp site[1]
- Winblowz.com Enlightenment Mirror Site[2]
- The EsounD Web Site[3]

Compiling

1. Extract the contents of the archive.

   ```
   > tar xvfz esound-0.2.4.tar.gz
   ```

2. To compile the package, change to the newly created esound-0.2.4 directory, and type:

   ```
   > configure ; make all
   ```

3. A sample test script is provided, follow the directions given by the test script:

   ```
   > ./test-script
   ```

The sound daemon, and the command line toolset should build as is. All executables should run fine from the source directory.

Installing

Installing the EsounD package is accomplished by typing:

1.

   ```
   > make isntall
   ```

This will copy all relevant files to useful/accessible places.

Notes

1. ftp://ftp.enlightenment.org/pub/ricware/esound-0.2.4.tar.gz
2. http://winblowz.com/~e/files/TAR/esound-0.2.4.tar.gz
3. http://www.netcom.com/~ericmit/esound_0_2_4.tgz

Chapter 3. Running EsounD

esd

The Enlightened Sound Daemon

esd `[options]`

option	purpose
-nobeeps	inhibits the default startup tone sequence
-d DEVICE	use audio device DEVICE (esd -h for values)
-b	run server in 8 bit sound mode
-r RATE	run server at sample rate of RATE
-as SECS	free audio device after SECS of inactivity
-vt	enable trace diagnostic info (debug builds only)
-vc	enable comms diagnostic info (debug builds only)
-vm	enable mixer diagnostic info (debug builds only)
-port port	accept connections on given port (default=5001)

In addition, "kill -HUP esd-pid resets ownership of the daemon. The file, ~/.esd-auth, is used for authentication, and is created if needed.

esdctl

controls certain aspects of the sound daemon.

esdctl `[options]`

commands	purpose
lock	foreign clients may not use the server
unlock	foreign clients may use the server
standby, off	suspend sound output for other programs
resume, on	resume sound output
cache SAMPLE	cache sample from file SAMPLE in the server
getid NAME	retrieve a sample id from its name
free NAME	uncache a sample in the server by NAME
play NAME	play a cached sample once
loop NAME	play a cached sample in a loop
stop NAME	stop a looping sample at end of sample
serverinfo	get server info from server
allinfo	get player and sample info from server

commands	purpose
panstream ID LEFT RIGHT	set left and right volume levels for a stream
pansample ID LEFT RIGHT	set default left and right volume levels for a sample

lock - only the "owner" of the daemon can play sounds. the first key (obtained from ~/.esd-auth) submitted to the daemon identifies the owner.

unlock - anyone can play sounds through the daemon.

standby - frees the audio device for use by other programs. all sounds are ignored until the daemon is resumed.

resume - brings daemon out of standby mode

NOTE: more than one option may be listed on the command line

esdcat, esdmon, esdrec

esdcat plays a raw audio stream through the daemon. esdmon outputs the mixed stream from the daemon. esdrec outputs from the sound device's current input.

esdcat `[options] [file=stdin]`

esdmon `[options] [file=stdout]`

esdrec `[options] [file=stdout]`

option	purpose
-s SERVER	communicate with esound running at SERVER
-b	read/write 8 bit sample data
-m	read/write mono data
-r RATE	use a sampling rate of RATE

NOTE: options for esdcat, esdmon, and esdrec default to 16 bit (signed), stereo, 44.1kHz

esdsample, esdloop

esdsample is test scaffolding for sample cache, play back, and free. esdloop is test scaffolding for sample cache, loop, and free. These utilities have no practical purpose beyond testing sample functionality.

esdsample `[options] file`

esdloop `[options] file`

option	purpose
-s SERVER	communicate with esound running at SERVER
-b	read/write 8 bit sample data
-m	read/write mono data
-r RATE	use a sampling rate of RATE

options: -s server = communicate with esound running at server -b = 8 bit unsigned char -m = mono -r rate = sample rate of rate Hz.

options for esdsample and esdloop default to 16 bit signed short, stereo, 44.1kHz

Environment Variables

All client programs (except esdctl) can connect to remote hosts via the ESPEAKER environment variable:

```
bash $ export ESPEAKER=inet.addr.of.host:port

tcsh > setenv ESPEAKER inet.addr.of.host:port
```

The client will connect to EsounD running on the specified host on the specified port.

Chapter 4. Miscellaneous Information

New Features

Version 0.2.4 more bug fixes, mostly works on irix now. incompatible changes to client side API. streams and samples now have names, and you can specify a host in esd_open_sound().

Version 0.2.3 bug fixes

Version 0.2.2 added reference counting to samples, so freeing samples works added first cut at drivers for other platforms. no idea if they work

Version 0.2.1 the configure style installation now works if you don't already have a previous version installed. Evil gnu automake >=P better handling of "dead" time: should use 0% cpu if it's not actually doing anything

Version 0.2 a configure style installation is now available. added network support optional /dev/dsp fallback for stream play and record other forgotten features and bug fixes

Known Problems, Deficiencies, and Possible Workarounds

"unsupported sound format: 33" - This happens on cards that don't support playback of 16bit, stereo, 44.1kHz sound. Try running in 8 bit mode (esd -b), or at a lower playback rate (esd -r 22050), or both (esd -b -r 22050)

"unable to connect to server at port 35091" - This means the client cannot contact the server. Make sure you have SysV IPC enabled in your kernel

Documentation of the API is in sad shape, but for now it's still in development (i.e. subject to change). See the source of the sample command line driver utilities (esd?*) for usage.

Any other pressing issues, problems, or patches may be directed to ericmit@ix.netcom.com. Please mention EsounD and the version number (date for cvs checkouts) in the subject line.

Planned Development and Top Suggestions So Far...

Porting of more sound playing applications. For sound programs that send their output to /dev/dsp, this is usually a trivial process. Anyone willing to lend a hand in this effort will be duly appreciated.

Planned Feature Additions

See the TODO file for unorganized thoughts about where to go from here.

Chapter 5. Enlightened Sound Daemon Aware Audio Programs

eMusic

eMusic[1] is an Enlightenend music player, which is capableof playing MP3, MOD, WAV, AU, and CD files. All graphic elements of the program are fully customizable.

The Extended Module Player

Extended Mudule Player[2] is a MOD tracker, which also has an Xwindows interface.

mpg123

mpg123[3] is a command line program which plays mpeg layer 3 audio files.

Notes

1. http://www.icom.net/~smelecat/emp3/index.htm
2. http://xmp.home.ml.org
3. http://www.netcom.com/~ericmit/mpg123_esd.tgz

Chapter 6. The EsounD API

Need to put something useful in here.

Better yet, extract it from the source files.

Chapter 7. Rough Design Docs

Covered Formats

Digital Waveforms

No MIDI, No MOD

Client/Server Organization

Server does the dirty work for byte swapping, scaling, etc.

The Audio Signal Path

The audio signal travels the following path from source to speaker. Samples and streams are read via players

Authentication Model

Your .esd_auth file

The Player Type

read_player()

Anything Else?

Anything Else?

I. EsounD Command Reference

esd

Name

`esd` — The Enlightened Sound Daemon

Synopsis

```
esd [options]
```

options

-nobeeps

Specifies an array of child widgets. Each child must be of class RectObj or any subclass thereof.

-as [SECS]

Specifies the number of elements in *children*.

usage

unmanaging widgets is the usual method for temporarily making them invisible. they can be re-managed with `xtmanagechildren()`.

Appendix A. Remaining details

Although this booklet is quite complete, here I will mention some details I never got to.

Use of the word dude

Here's an example of how to say *dude*: DUDE.