

MUSE

Published : 2011-06-22

License : None

INTRODUCTION

1. MuSE

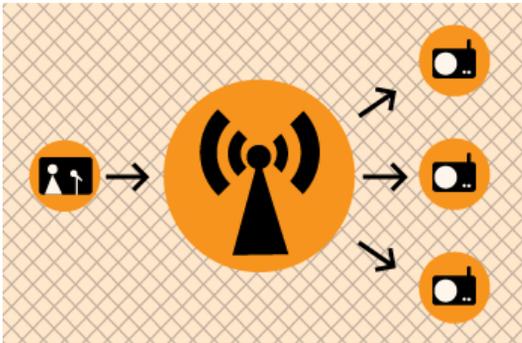
2. What is streaming?

1. MUSE

MUSE is the Multiple Streaming Engine developed by the programmers at Dyne (<http://www.dyne.org>).

Streaming is the term given to sending live audio or video over the internet. MUSE enables you to stream audio.

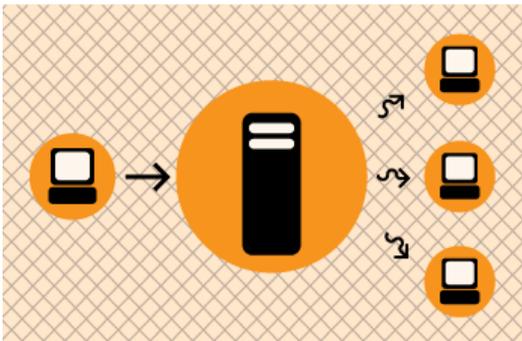
Perhaps a good way to understand what streaming is, is to imagine a radio station. A radio station comprises of three components - a *studio*, a *transmitter*, and the *receivers* that your audience has...



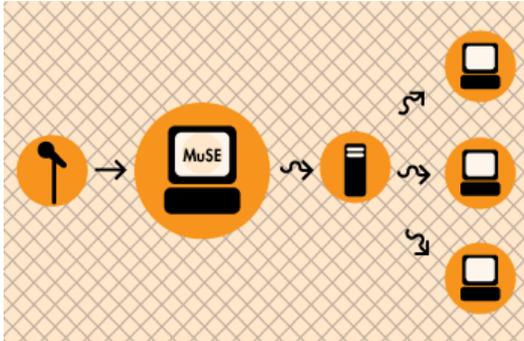
Above is a basic diagram showing how a transmitting radio station works. The radio studio is the source of the audio. In this space there are usually mixing desks, cd-players, minidisc players, turntables etc. Then from the studio an audio signal is sent to the transmitter. This can be sent from the studio to the transmitter by either a cable (sometimes called a "landline") or by a microwave link. Then the transmitter sends the audio via FM so that radio receivers (tuners) can pick it up and play it.

Radio works this way because it is trying to distribute the studio audio to as many people as possible. If you imagine the radio studio without the transmitter then the station would have a fairly reduced audience! Essentially only people that could fit into the studio would be able to listen. So the transmitter works as a distributor, allowing more people to connect via their radio receivers and hence the potential audience is enlarged.

This is a close analogy to why streaming exists and how it works. If you were just playing audio on a computer in your room then the audience isn't going to be so big...so, we utilise streaming to distribute this audio to more people.



The analogy is ...the computer replaces the radio studio, the streaming server replaces the transmitter, and your listeners connect by computers to the server rather than with radio receivers to the broadcast signal. Using this model we can see that MuSE is the software that sends the audio from your computer to the streaming server.



In this way you can use the program for creating a live online radio station or perhaps for event broadcasting, or other cultural or tactical projects. You can use MuSE under **Linux** or **Mac OSX**.

2. WHAT IS STREAMING?

Streaming Media is the term used to describe the *real-time delivery* of audio and video over the internet. Streaming Media allows for *live transmission* of audio or video over the internet, transforming the internet into a broadcasting medium. Content can also be archived in streaming file formats[?], allowing internet users to experience recordings of events after they happen.

VARIETIES OF STREAMING

There are three types of delivery of audio and video data over the internet: Download and Play, Progressive Download, and True Streaming.

Download and play

To experience Download and Play media you must first download the entire media file to your hard disk before you can play it. For this reason Download and Play media cannot be used for live broadcasts, however it is often a good way to deliver high quality media content over any bandwidth. A high quality movie, for example, can be downloaded over any internet connection. However because you cannot play the movie until you have downloaded the entire file, download time becomes a factor. To download a DVD movie (usually about 4.6 GB in size) over a dial-up modem (56kbps) it would take over 8 days! Most people would find this quite tedious!

Download and Play media is *not* streaming media (more on this later) but it is a lot more popular than streaming. File sharing networks are Download and Play mechanisms, and this is an extremely popular activity on the internet. As many countries make high bandwidth connections more accessible there is an increasing popularity of trading Download and Play movies on these networks.

Progressive download

This enables you to experience media as it downloads to your harddrive. Progressive Download is useful but is less efficient than True Streaming and cannot provide some functionality such as multiple bitrate encoding. However this method has an advantage over Download and Play as you do not have to wait for the file to finish downloading before you can start playing it. There are some constraints - for example, if the file does not download as fast as you are playing it, then you will find that playback will stop while you wait for more of the file to download. This can be annoying and it is often the reason why some choose True Streaming as a preferred method of delivery.

Quicktime (Apple's suite of streaming technologies) calls Progressive Download "Fast Start". This highlights one of the difficulties of learning streaming - in the effort to differentiate technologies, each technology provider has made up their own jargon. However throughout this manual I will stick to the more generic terminologies wherever possible.

True streaming

This enables the user to view or listen to the media as it is delivered over the internet in real-time. True Streaming is the only variety of streaming that allows for live broadcasting, and it also supports more advanced functionality than Progressive Download, enabling the user to control the media experience by pausing, skipping forward, or rewinding. Although different terms have been used to describe True

Streaming, including webcasting, livecasting, web tv, or net.radio, the term that has prevailed is "streaming media" (more often it is just referred to as "streaming").

The difference between Progressive Download and True Streaming is not always obvious. If, for example, a user has a very fast connection to the internet then Progressive Download may appear to behave in exactly the same way as True Streaming.

Within Download and Play, Progressive Download, and True Streaming there are two distinct varieties : static file streaming and live streaming.

Static file streaming

The delivery of pre-recorded media files over the internet in real time. Typically when we refer to archives of online media we are discussing static file streaming. A large archive of video art encoded into streaming files is an example of a collection of static file streaming files. This content is also known as on-demand or archived content. This content can be delivered by Download and Play, Progressive Download, or True Streaming processes.

Live streaming

The delivery of live audio and/or video over the internet. This allows the user to experience an event as it occurs in realtime. There are many examples of this such as online radio or viewing live performances. Only True Streaming supports live streaming.

This manual will mainly deal with True Streaming of both live and archived content.

ENCODING

To stream static files over the internet the files must be compressed and encoded into a 'streaming format'. Ripping CDs into MP3 is doing just this and is done using encoding softwares. In this process a CD audio recording is converted by the encoding software from a CD (Compact Disc) audio format into a 'streaming audio format' (MP3). This will involve compressing the data, which reduces its quality and file size, and converting the data into a 'streaming format' (for example MP3).

Compression

There are two forms of compression - lossy, and lossless compression.

When the encoding process compresses the source file so that it can be delivered over internet connections in real-time this process degrades the quality of the audio and video. The more a file is compressed, the lower bandwidth required to be able to play the file, but the more the quality is reduced. A compromise has to be achieved whereby the level of compression achieves an acceptable audio and video experience, while reducing the amount of data enough to enable delivery over the internet.

Live encoding is similar except that an audio or video (or both) input is encoded instead of a file. With this process the encoding software delivers the encoded data in a continuous stream to the streaming server.

Lossless compression

This is the process of compressing data information into a smaller size without removing data. To visualise this process imagine a paper bag with an object in it.

When you remove the air in the bag by creating a vacuum the object in the bag is not affected even though the total size of the bag is reduced

Lossy compression

Sometimes called 'Perceptual Encoding', this is the process of 'throwing away' data to reduce the file size. The compression algorithms used are complex and try to preserve the qualitative perceptual experience as much as possible while discarding as much data as necessary.

Lossy compression is a very fine art. The algorithms that enable this take into account how the brain perceives sounds and images and then discards information from the audio or video file while maintaining an aural and visual experience resembling the original source material. To do this the process follows Psychoacoustic and Psychovisual modelling principles.

Codecs

The algorithms used to compress and encode audio and video and create the file format are known as "codecs". The word is made from two common terms compress and decompress. The encoding software uses a codec to 'compress' the streaming content for delivery over the net, and the player softwares use a codec to 'decompress' the content for replay.

The codec is mathematical wizardry and is the heart and soul of streaming. It is on this ground that we have the various technology providers battling it out for dominance. Each codec has its own unique way of doing things depending on what the developer believes is best, and some come with their own minefield of licencing issues too.

Bitrate

Compression is the process of reducing the amount of data

Multiple bitrate encoding allows the server and player to negotiate the best quality (highest bandwidth) stream to be delivered from a single static file or live stream. Hence the player is delivered the best quality stream possible over the user's internet connection. Multiple bitrate? encoding produces only one encoded stream.

Choosing the bitrate(s) will require the consideration of several factors, including: The target audience's connectivity; The desirable frame size of the video (if including video); The amount of movement in the frame (if including video); The level of video contrast (if including video); The type of audio encoded (e.g. ambient noise/voice/stereo music); The amount of camera movement (if including video); The quality of the camera and camera lens (if including video).

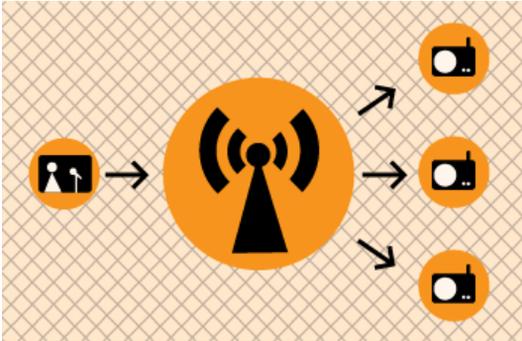
DELIVERY

Streaming static ("archived") files can be achieved using a normal web server. This is often the cheapest way to deliver content on a small scale. This method cannot be used for live streaming and does not allow for the advanced features of True Streaming such as multiple bitrate encoding. This method will also enhance the likelihood for time-outs ('buffering') and cannot deliver the same amount of simultaneous player connections as True Streaming.

To enable live streaming and to gain full functionality and efficiency a streaming media server is required. This server is usually standard server hardware but with the necessary streaming server softwares installed. It is quite normal to install a streaming server on the same machine as an existing web server.

Streaming servers

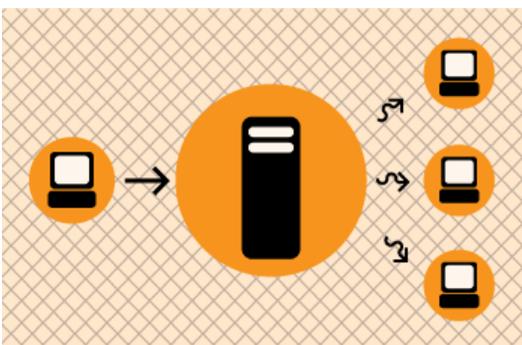
Perhaps a good way to understand what a streaming server does is to imagine a radio station. A radio station comprises of three components - a *studio*, a *transmitter*, and the *receivers* that your audience has.



Above is a basic diagram showing how a transmitting radio station works. The radio studio is the source of the audio. In this space there are usually mixing desks, cd-players, minidisc players, turntables etc. Then from the studio an audio signal is sent to the transmitter. This can be sent from the studio to the transmitter by either a cable (sometimes called a "landline") or by a microwave link. Then the transmitter sends the audio via FM so that radio receivers (tuners) can pick it up and play it.

Radio works this way because it is trying to distribute the studio audio to as many people as possible. If you imagine the radio studio without the transmitter then the station would have a fairly reduced audience! Essentially only people that could fit into the studio would be able to listen. So the transmitter works as a distributor, allowing more people to connect via their radio receivers and hence the potential audience is enlarged.

This is a close analogy to why streaming exists and how it works. If you were just playing audio on a computer in your room then the audience isn't going to be so big.. so, we utilise streaming to distribute this audio to more people.



The analogy is obvious....the computer replaces the radio studio, the streaming server replaces the transmitter, and your listeners connect by computers to the server rather than with radio receivers to the broadcast signal. The analog can be taken quite a long way. Having a bigger radio transmitter is like having more bandwidth available at the streaming server - both allow more people to connect.

INSTALLING

3. Installing from Source

3. INSTALLING FROM SOURCE

Software name : Multiple Streaming Engine (MuSE)

Homepage : <http://muse.dyne.org>

Software version used for this installation : 0.9

Operating System used for this installation : Debian 3.1 (Sarge)

Recommended Hardware : 200 MHz CPU or greater

Requirements : internet connection, root password (also known as the admin or superuser password)

Installing from source' means that you download the actual **source code** for an application and you compile it into a **binary**. A "binary" is often referred to as an "executable" , "application" or "program". The source code is what the programmers actually create, its a series of text files that contain program code. These files don't actually do anything until you **compile** them. "Compiling" is the process by which you convert the text files containing the program code into a binary. In this case you take the MuSE *source code* and *compile* it into a *binary*. It is then this binary that you then use.



DOWNLOADING THE SOURCES

Please visit <http://muse.dyne.org> and download the latest source distribution of MuSE, at the time of writing the latest version is 0.9, so I will download the "MuSE-0.9.tar.gz" file. You will need to first visit <http://muse.dyne.org> and then click on the download link:

:: [description](#)

:: [features](#)

:: [developers](#)

:: [download](#)

:: [documentation](#)

:: [Make a Donation](#)

You will now see a list of **ftp** sites where you can get the source code:

free download of sourcecode



[ftp] [Dyne.org software archive](#)
[ftp] [Gigaload dyne mirror](#)
[ftp] [Tekkaman Porgfa mirror](#)
[http] [Gigaload dyne mirror](#)
[http] [Ljudmila medialab mirror](#)
[http] [GNU Generation mirror](#)
[rsync] [rsync.dyne.org::muse](#)

Click on the link labeled:

[ftp] Dyne.org software archive

Now you can see a list of files and directories looking something like this:

Index of ftp://ftp.dyne.org/muse

[Up to higher level directory](#)

 AUTHORS	3 KB	27/12/05	00:00:00
 COPYING	18 KB	14/12/03	00:00:00
 ChangeLog	10 KB	27/12/05	00:00:00
 INSTALL	8 KB	17/04/04	00:00:00
 KNOWN-BUGS	1 KB	16/12/04	00:00:00
 NEWS	6 KB	16/12/04	00:00:00
 README	8 KB	27/12/05	00:00:00
 TODO	1 KB	16/12/04	00:00:00
 USAGE	4 KB	16/12/04	00:00:00
 binary		28/06/06	22:40:00
 muse-commandline.html	7 KB	14/12/03	00:00:00
 muse.i	5 KB	14/12/03	00:00:00
 old-binaries		28/06/06	22:40:00
 opensource-radio-streaming.pdf	19385 KB	15/10/04	00:00:00
 releases		27/12/05	00:00:00
 sources		15/04/05	00:00:00

Click on the bottom directory called "sources". You will now see a list of files like this:

Index of ftp://ftp.dyne.org/muse/sources

[Up to higher level directory](#)

 oo LATEST IS 0.9.2	17/04/02 00:00:00
 MuSE-0.1.1.tar.gz	216 KB 22/09/00 00:00:00
 MuSE-0.1.2.tar.gz	206 KB 22/09/00 00:00:00
 MuSE-0.1.2.tar.gz.sign	1 KB 22/09/00 00:00:00
 MuSE-0.2.tar.gz	224 KB 22/09/00 00:00:00
 MuSE-0.2.tar.gz.sign	1 KB 22/09/00 00:00:00
 MuSE-0.3.1.tar.gz	234 KB 26/10/00 00:00:00
 MuSE-0.3.1.tar.gz.sign	1 KB 02/12/00 00:00:00
 MuSE-0.3.tar.gz	227 KB 10/10/00 00:00:00
 MuSE-0.3.tar.gz.sign	1 KB 10/10/00 00:00:00
 MuSE-0.4.1.tar.gz	341 KB 24/01/01 00:00:00
 MuSE-0.4.1.tar.gz.sign	1 KB 24/01/01 00:00:00
 MuSE-0.4.tar.gz	445 KB 02/12/00 00:00:00
 MuSE-0.4.tar.gz.sign	1 KB 02/12/00 00:00:00
 MuSE-0.4pre1.tar.gz	558 KB 08/11/00 00:00:00
 MuSE-0.4pre2.tar.gz	554 KB 08/11/00 00:00:00
 MuSE-0.4pre3.tar.gz	560 KB 11/11/00 00:00:00
 MuSE-0.5.1.tar.gz	349 KB 08/03/01 00:00:00
 MuSE-0.5.1.tar.gz.sign	1 KB 08/03/01 00:00:00
 MuSE-0.5.2.tar.gz	396 KB 23/03/01 00:00:00

Scroll down to the latest version of the software (in this case the name of that file is "MuSE-0.9.tar.gz") and click on that file name. If all is working correctly the download should begin automatically from your browser.

COMPILING

Once it has downloaded you need to open a **terminal**, use the `cd` command to change working directories so you are in the same directory as your downloaded source. If you haven't used a terminal before it would be good first to read up something about this otherwise you are not going to get very far. If you are not sure what a terminal looks like, it looks something like this:

```
eset@focus: ~/home/eset
```



Assuming you have a terminal open you need to change directories to where you downloaded the MuSE sources. Forgot where you put them? A common error - use `slocate` or `locate` to find the file, you may have to do an **updatedb** first. For example, I would type

```
slocate MuSE-0.9.tar.gz
```

and this would output to the terminal where my file is located and I would then use `cd` to get there.

Okedoke... now unpack the archive by typing `tar zxvf` followed by the name of the file you just downloaded, I would type the following:

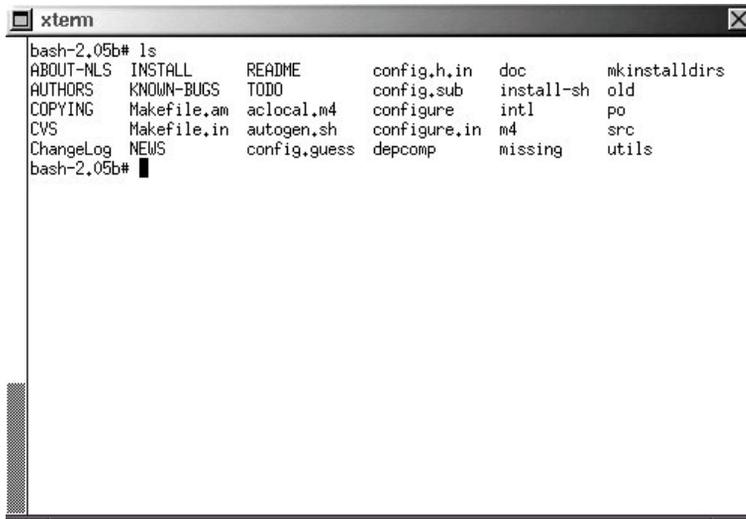
```
tar zxvf MuSE-0.9.tar.gz
```

And the file would unpack nicely to a new directory within the directory I am currently in. Typing a `ls` will show this. In my case there is a new directory called "MuSE-0.9" . This directory could be named differently to what you have, depending on what version of the software you downloaded.

Now `cd` to the new directory. Again in my case I would type:

```
cd MuSE-0.9
```

This would place me in the MuSE source directory. If I type `ls` I get this:



```
xterm
bash-2.05b# ls
ABOUT-NLS  INSTALL      README      config.h.in  doc          mkinstalldirs
AUTHORS     KNOWN-BUGS  TODO        config.sub   install-sh   old
COPYING     Makefile.am aclocal.m4  configure    intl         po
CVS         Makefile.in autogen.sh  configure.in  m4          src
ChangeLog   NEWS        config.guess  depcomp     missing     utils
bash-2.05b#
```

cool! these files are all the files you need (hopefully) to install MuSE

Now proceed by typing :

```
./configure --with-gtk2 --with-x --x-includes /usr/include --x-libraries /usr/lib
```

This should start the **configure** process. Now, having said you need all the files in this directory to install MuSE, this is indeed true but without some specific extra software MuSE is not much more than a simple mp3 player. So check the output of your **configure script**... it will look something like this:

```
=====
config.status: creating src/resample/Makefile
config.status: creating intl/Makefile
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing default-1 commands

== MuSE build configuration =====
:: Building on i686-pc-linux-gnu (kernel Linux267)
:: Linking OggVorbis libraries for OGG encoding
:: Linking SndFile libraries for WAV/AIFF/VOC/etc encoding
:: Linking Lame libraries for MP3 encoding
```

```

:: Building the GTK-2 user interface
:: Building the NCURSES user interface
:: DEBUG informations OFF
:: PROFILING informations OFF (see man gprof)
:: CFLAGS = -pipe -D_REENTRANT -O3 -fomit-frame-pointer
:: CXXFLAGS = -pipe -D_REENTRANT -O3 -fomit-frame-pointer
bash-2.05b#
:: LIBS = -lncurses
=====

```

now type **make**, may the source be with you! The bit you need to look at is the end part which gives a nice summary of what **libraries** are installed that MuSE will work with. In the case of the output above the configure script has reported that I have all the possible libraries installed, i.e. I have **OggVorbis, SndFile, Lame, GTK-2, and NCURSES**

DEPENDENCIES

If I read the "README" file (use your favourite editor from the section above to open this file) in the MuSE directory, the file tells me this:

```

=====
- LAME (optional)
  Lame can be installed but is no more needed, in particular
  libmp3lame and the header lame.h must be properly installed.

- OGG VORBIS (optional)
  You can compile and install libogg and libvorbis on your machine
  before compiling MuSE; the configure script will recognise them and
  include support for decoding and mixing of .ogg files.

- GTK+ (optional)
  if libgtk and all the related libraries are present, MuSE will
  compile the GTK+ graphic user interface for interactive use and
  additional fun.

- NCURSES (optional)
  if libncurses is present, MuSE will compile a text console
  interactive interface to be used into ASCII terminals.

- SNDFILE (optional)
  if libsndfile is present then you'll be able to play uncompressed
  sound files like wav, aiff, snd, voc, pvf, mat, au, sf etc.
=====

```

So I can do everything possible with MuSE because I have all the libraries installed. However if the output from your configure script says something is missing then you need to consider whether you should install it or not. The basic minimum I would hope that you have installed is: **GTK, Lame**

If you have these two libraries installed you can at least have a graphical user interface on MuSE (provided by GTK) and you can stream MP3 (provided by the Lame) library. If you don't have these libraries get them from here:

Lame

<http://lame.sourceforge.net/>

The libraries required if you wish to stream mp3.

GTK

<http://www.gtk.org/>

Note : I would be surprised if you didn't have GTK installed. Its possible, but unlikely as most distributions of Linux have this library as it contains many libraries needed by other softwares commonly found on Linux. If you don't have GTK it can be very tricky to install so I would recommend you get some help for this.

Ogg

If you also want to stream Ogg then you must first install "libogg" and "libvorbis" (<http://www.xiph.org/ogg/vorbis/index.html>)

If you need any of these libraries and the configure script says you don't have them installed download them first, run through the install process as outlined in the section above, and after each install run the *ldconfig* command to update the library database. Then once you have done this for each one, start again with the *.configure* process in this section.

Now, lets assume your configure script now gives you the output you want, indicating that all the required libraries are installed. Cool!

Now you just want to type the *make* command as so:

```
make
make install
```

If all has gone well you should just need to type :

```
muse
```

and the application will open as so:



INTERFACE

4. The Tool Bar

4. THE TOOL BAR

Software name : MuSE
Software version : 0.9

Now at the top we have a tool-bar:



BUTTON DESCRIPTIONS



The first button is the button that opens the configuration panel. It is in this section that you enter the information such as that outlined above (**hostname**, **port** etc) as well as the **format** (**bitrate**, **ogg/mp3** etc) of the stream. This is the part of the application that determines most whether you stream or not, so we will go over this in detail in a moment.



This button adds a new **channel** to MuSE, you won't need this until you are more experienced.



This button means that we want to stream using the input from the **sound card**, you will want to turn this on, but we will get to that soon.



This button means you want to stream using **playlists**, this is covered in a later section and for live streaming using the sound card you will want to turn this button off.



This button opens the **volume meters** so you can monitor the volume of the outgoing stream.



This button opens the panel with the credits... check it out, its quite funny and it's worth knowing who provided you this great software!

USING

5. Live streaming with MuSE
6. Streaming playlists with MuSE

5. LIVE STREAMING WITH MUSE

Software name : MuSE²

Software version : 0.9



One of the many conferences streamed by RIXC (<http://www.rixc.lv>) using MuSE.

For this section I am going to make the assumption that you already have access to a **streaming server** and that someone has given you the details. Specifically you need to know:

- The **hostname** (IP address or URL) of the streaming server
- The **login type** of server being used (shoutcast/ Icecast / Icecast2 etc)
- The **password** of the server
- The **mountpoint** you should use (more on this later)
- The **port** you will stream to (usually 8000)

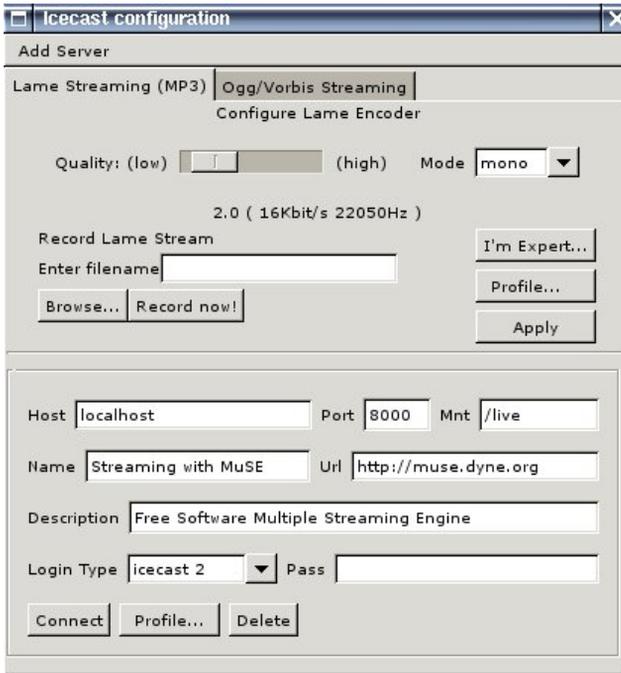
Ask the person that administrate the streaming server for this information. If on the other-hand you intend to run your own server, as explained later in this manual, then you will get these details yourself.

OPENING THE CONFIGURATION WINDOW

Press the **configuration button** (first button)



so, under **Linux** you will see this:



CONFIGURING

We need to enter here all the information relevant to the stream we want to begin. Lets start with streaming **MP3**...so the panel that is to the front here is the panel for setting up a **MP3 stream**. if you want to do an **Ogg stream** then click on the panel "Ogg/Vorbis Streaming" at the back, and jump forward in this section to where this is covered... for now we will just do MP3.

For now leave the top part of this form alone... the top sets the **bitrate** and **quality** of the outgoing stream, we will first just get the stream going then we will tweak this part a bit. ok... we have some information fields that need filling in, these all have to do with the **streaming server** that you are streaming to. We will assume that you are using an **Icecast2** server. If you are using a different streaming server then everything will be the same except you will need to know what kind of **Login Type** the server is... I will let you know more about this when we come to fill in this field.

The next section gives a field by field account of the MuSE configuration panel.

Host

Enter here the **URL** or

IP address of the server you will be streaming to. If you are running a server on your own machine (where you are also encoding from) then you need to enter "localhost" in this field. Other wise you need to enter everything including the "http://" section. For example, if my server was located at "http://icecast.streaming suitcase.com" then I would enter exactly that into this field, i.e.:

<http://icecast.streaming suitcase.com>

The host information will be given to you by the person running the server.

Port

This is usually by default port **8000**. The person running your server will give you this port number and you should enter it in this field, if they haven't given you a port number try 8000

first and if this doesn't work then call them and ask them for it.

Mount

A **mountpoint** is the unique point on the server that your stream will connect to. This means if you are running a streaming server and there are (for example) 20 people sending individual streams to the server then each can be identified by the mount point. It is also used in the **playlist** file you will create later so that you can link to the stream from your web-page. It doesn't matter what the name of your mountpoint is, although sometimes the server administrator will give you a specific one to use. To make sure you won't conflict with another stream don't use obvious mountpoint names like "test" etc... also make sure your point point has a preceding " / " as indicated in the example screen shot above.

Name

The name that will appear in the media player of your listeners when they connect to your stream... using something more descriptive than just "my stream" or some such.

URL

This is not the **URL** of the streaming server, its just a URL that sometimes also gets displayed in the clients player. Its for displaying the URL of the associated web page.

Description

Pretty much the same as the name field except you can fit in more information.

LoginType

This is the type of authentication your server uses, here is a table to help you choose the correct one:

SERVER TYPE	LOGIN TYPE
Darwin	icecast 1
Icecast 2	icecast 2
Icecast 1	icecast 1
Shoutcast	shoutcast

Pass

Enter your **password** for the server here. This should be given to you by the system administrator for the streaming server.

Now start your stream by starting some audio from your mixing desk (or whatever you have plugged into your sound card), make sure your software mixer is turned up, then press "connect" on this form and away you go!

6. STREAMING PLAYLISTS WITH MUSE

Software name : MuSE²

Software version : 0.9

Instead of working with a live input you can choose to stream **pre-recorded material** with MuSE. To do this please first configure MuSE as discussed in the last section and then follow these two simple steps.

LOADING THE PLAYLIST

So, lets look at the original window:



If you want to play pre-recorded material, then you will need to create a **playlist** with MuSE, which is easy... the first step is to make sure the big 'speaker' button is pressed down.



Once you have pressed the speaker button, then 'right-click' on the white colored window space and choose 'add file'... then browse to a **MP3** file on your **harddisk** (or any format audio file) and add it to the playlist



Do this as many times as you want... then when you are ready we can start streaming.

STARTING THE STREAM

If you have chosen to play a playlist, then highlight one of the tracks and press the play button, and return to the preferences window by clicking on the button at the far right, and press 'connect' on the configuration panel... thats it!!!

APPENDICES

7. Related Links
8. Somebody Should Set The Title For This Chapter!
9. License

7. RELATED LINKS

MANUALS

Center for Independent Journalism

Some great manuals on Sound Editing, Sound Effects, Turning Sound Online, Sound Editing (advanced), Setting Up a Station and a Studio, Designing a Website
<http://www.cijmalaysia.org/resources.htm>

Spot

Good site for some tips and tricks, especially if you use Dyne::bolic
<http://spot.river-styx.com/linkage.php>

ENCODERS

MuSE (Linux / OSX)

The best audio encoder for Linux.
<http://muse.dyne.org/>

Oddcast (Windows)

Great plugin for encoding Ogg and MP3 using Winamp.
<http://www.oddsock.org/tools/oddcastv3/>

ostream (Windows / Linux)

A cool ogg encoder that runs on Linux, Microsoft Windows, and MacOSX.
<http://radiostudio.org/ostream>

oggment (Linux)

An interesting tool which can encode a single audio source into a real stream and a ogg stream simultaneously... made by August Black (radio/software/media artist)... there are some other interesting tools on this site too
<http://oggment.sourceforge.net/ogg-real.php>

oggcaster (Linux)

A plug-in for xmms to stream ogg files to icecast2... looked interesting but i couldn't get it to compile... I think there was quite some activity on the development side, so it might be fixed soon... if doing encoding this way is your thing then don't forget the excellent xmms-oddcast plug-in (for icecast2) and the liveice plug-in (for icecast1 only) - both of which are available from the xmms (<http://www.xmms.org>) website in the plug-ins directories

<http://xmms-oggcaster.sourceforge.net/>

Soma (Linux)

A very interesting set of tools... sound daemon, player, encoder and scheduler... ate

a whole lot of my cpu and wasn't so nice with the crossfades, otherwise i might have used it...

<http://www.somasuite.org/>

darkice (Linux)

A very light weight, easy to deploy command line live mp3 streamer. You may also wish to look at the GUI interface, which you need to install seperately, it's called darksnow.

<http://darkice.sourceforge.net/>

liveice (Linux)

Encoder for icecast1. Command line

<http://star.arm.ac.uk/~spm/software/liveice.html>

IceS (Linux)

Command line encoder for icecast1 and icecast2

<http://svn.xiph.org/releases/ices/>

Oddcast (Windows)

A great plugin for XMMS that allows you to stream MP3 or Ogg. It can be tricky to install as it requires WXWindows.

<http://www.oddsock.org>

MPEG4IP (Linux)

A great how to for mpeg4ip streaming to a Darwin Server

<http://www.linuxjournal.com/node/6720/print>

PLAYERS

VLC (Windows / Linux / OSX)

The all purpose cross-platform media player. Does much more too...

<http://www.videolan.org/vlc/>

zinf (Windows / Linux)

A many featured player but couldn't see the advantage over xmms (my preferred gui player) except that it has a built in streaming encoder/server.

<http://www.zinf.org/>

soundplay (beOS)

This is good if you have beOS (i thought it was for linux at first)... comes with a built in streaming server....

<http://www.xs4all.nl/~marcone/soundplay.html>

snackAmp (Linux/ Windows)

A hugely featured gui player... almost too many features... has a built in web-server for remote control, and a built in streaming server... I would have used it for the project I was researching if it could work as a command line interface.

<http://snackamp.sourceforge.net/>

sonic-rainbow (Linux)

This looks nice and simple but i havenever tried it... it plays video too

<http://sonic-rainbow.sourceforge.net/>

impish (Linux)

This looked like a very sophisticated command line player but alas it has no crossfade and I couldn't get it to compile.

http://www.geocities.com/kman_can/

xmms (Linux)

This is my favorite player if only because it does so much

<http://www.xmms.org>

mplayer (Linux / Windows / OSX)

This is more a video player but also support audio. You can run it with or without a gui

<http://www.mplayerhq.hu/>

EMBEDDED PLAYERS

Flam

Flash Mp3player

<http://www.flamplayer.com/>

Make your own flash mp3 player

<http://web.uvic.ca/hrd/halfbaked/howto/audio.htm>

http://www.sonify.org/home/feature/remixology/026_mp3player/

Flow

Flash Video Player (open source)

<http://flowplayer.sourceforge.net/>

STREAMING SERVERS

Icecast

The best audio server on the net.

<http://www.icecast.org/>

An excellent Icecast Howto from Kerry Cox

<http://www.gnuware.com/icecast/>

Quicktime / Darwin Streaming Server

An **excellent** how-to for Quicktime Streaming Server, it also covers embedding Quicktime, compression, relaying streams etc

<http://www.soundscreen.com/>

A great quick how-to for Darwin and VLC (as a server) on Linux

<http://viz.aset.psu.edu/ga5in/StreamingMedia.html>

gini

Looks good... haven't tried it

<http://gini.sourceforge.net/>

Oyez

Python streaming server

<http://ubertechnique.com/seth/oyez/doc/Oyez.html>

oggserv

A php based daemon

<http://oggserv.sourceforge.net/>

litestream

An excellent streaming server with source clients and re-streamer... ultra stable

<http://www.litestream.org>

ample

A small streaming server

<http://ample.sf.net>

edna

This allows streaming of files stored on your 'server'

<http://edna.sf.net>

gnump3d

Another streaming server, also supports ogg

<http://www.gnump3d.org>

SOFTWARE REPOSITORIES

Dyne

The home of really good streaming software for Linux.

<http://www.dyne.org/>

Oddsock

A great home for streaming tools for Windows and Linux.

<http://www.oddsock.org/>

Freshmeat

If you don't know it you are guaranteed to spend the next x hours browsing it.

<http://freshmeat.net/>

SourceForge

Repository for many softwares.

<http://www.sf.net>

Version Tracker

Good for MacHeads.

<http://www.versiontracker.com/>

Tucows

Who said I don't look after newbies...an oldie but still useful, especially for Windows users.

<http://www.tucows.com/>

GNU

Good old GNU roll your own software repository for GPL software.

<http://savannah.gnu.org/>

OPEN CONTENT REPOSITORIES

Archive.org

Archive.org, otherwise known as the 'Internet Archive' is a place where you can store all your archived streaming files for free. The material must be licenced under Creative Commons or released to the public domain.

<http://www.archive.org>

ourmedia.org

Free hosting of media content for 'grass roots' media. Also for creative commons or public domain content

<http://www.ourmedia.org/>

v2v

Independent and activist media content storage (in ogg theora)

<http://www.v2v.cc/>

cc mixer

Sharing of audio samples licenced under creative commons

<http://ccmixter.org/>

DIGITAL AUDIO AND CODECS

Information on Formats

Good article on codec comparisons

<http://www.cdburner.ca/digital-audio-formats-article/>

ARS Technica Codecs

Good article on the difference between average, constant and variable bitrate codec settings

<http://arstechnica.com/guides/tweaks/encoding.ars>

FREE CODECS

xiph

The home of free codecs.

<http://www.xiph.org/>

ogg vorbis and theora

Information about free audio codecs and third party applications.

<http://www.vorbis.com/>

Very good posting on OGG

<http://forum.doom9.org/showthread.php?s=&threadid=77314>

An embeded ogg java player

<http://www.flumotion.net/cortado/>

Good Howto for streaming Theora to Icecast2

<http://www.oddsock.org/guides/video.php>

XVID

Open Source codec mostly used for ripping and playing DVDs. <http://www.xvid.org/>

Dirac

Upcoming free video codec developed by the BBC

<http://dirac.sourceforge.net/>

MP3 Patents

Good detailed information on MPEG related patents

<http://swpat.ffii.org/patents/effects/mpeg/index.en.html>

LICENCING CONTENT

Creative Commons

Alternative content licencing models.

<http://creativecommons.org/>

LICENCING SOFTWARE

Free Software Foundation

<http://www.fsf.org/>

What is FLOSS?

http://en.wikibooks.org/wiki/FLOSS_Concept_Booklet

PD STREAMING EXTERNALS

A complete list:

<http://pd.iem.at/pdb/query.html?ext=y&alb=y&qry=complete>

Peer2Peer Streaming

If you want a peer2peer Ogg streamer (ogglive~) for Windows and Mac OSX versions of PD then go here:

<http://www.nullmedium.de/dev/ogglive~/>

Speex, MP3, P2P

Speex Codex external (Speex~) for Linux and MP3 streaming externals (Mp3cast~) for Windows and Linux (includes a MP3 peer2peer external - mp3live~) versions of PD available here:

<http://ydegoyon.free.fr/software.html>

OGG

If you want Ogg streaming for PD then you can get the pdogg~ externals in an installed version of pd-extended or by installing the pd-externals package, both are available from here:

<http://pure-data.sourceforge.net/download.php>

MAX/MSP STREAMING EXTERNALS

MP3

Shoutcast external for streaming MP3 from MAX/MSP:

<http://www.akustische-kunst.org/maxmsp/>

OGG

If you want Ogg streaming for MAX then check here:

<http://www.nullmedium.de/dev/oggpro/>

TRANSCODING AND RIPPING

Mencoder

A good guide to transcoding using Linux and Mencoder

http://gentoo-wiki.com/HOWTO_Mencoder_Introduction_Guide

Ripping Guides

A good DVD ripping guide for Linux

<http://www.bunkus.org/dvdripping4linux/> a good how to for creating MP3s (for the slightly geeky)

<http://howtos.linux.com/howtos/MP3-HOWTO-10.shtml>

VIDEO EDITORS (GPL)

Jahshaka

The best I've seen for Linux

<http://www.jahshaka.org/>

There is also a very good beginners tutorial for Jahshaka here:

<http://users.pandora.be/acp/jah/>

Pitivi

Under Development

<http://www.pitivi.org/>

Cinelerra

Good but you need lots of screen space

<http://heroinewarrior.com/cinelerra.php3>

Cinelerra Tutorials

<http://www.ftconsult.com/twiki/bin/view/Cinelerra/CinelerraManualTOC>

<http://supreetsethi.net/drupal/?q=node/20>

Kino

Great editor

<http://www.kinodv.org/>

General

Some good notes on editing video on Linux

http://users.dslextreme.com/~craig.lawson/linux_notes/video.html The same site has

a great section on CD and DVD authoring under Linux

http://users.dslextreme.com/~craig.lawson/linux_notes/cds.html

MISCELLANEOUS

History of MP3

A brief overview of the most popular audio format on the net.

http://www.mp3-mac.com/Pages/History_of_MP3.html

Low Tech

Great spot to learn more about software and hardware if you are in the UK.

<http://lowtech.org>

Your Machines

Cultural event and ongoing research/writings about free software.

<http://www.yourmachines.org/>

OggPlay

Play Ogg on your mobile phone

<http://symbianoggplay.sourceforge.net/>

Googles beta video search

Using the VLC plugin for replay of content....

<http://video.google.com/>

swf.open4all

Experiment with collecting and transcoding (to flash) most commonly linked video content

<http://swf.open4all.info/>

WORKSHOPS

The Streaming Suitcase does various workshops including streaming, PureData, open source audio, linux etc. Workshops and resources by these people are also highly recommended:

Julian Oliver

Blender and gaming workshops (etc)

<http://selectparks.net/>

<http://selectparks.net/~julian>

<http://selectparks.net/modules.php?name=News&file=article&sid=395>

Derek Holzer

PureData and open source audio workshops (etc)

http://www.umatic.nl/info_derek.html

Tetsuo Kogawa

MiniFM

<http://anarchy.translocal.jp/MiniTV>

<http://www.silenttv.net>

8. SOMEBODY SHOULD SET THE TITLE FOR THIS CHAPTER!

STREAMING

9. LICENSE

All chapters copyright of the authors (see below). Unless otherwise stated all chapters in this manual licensed with **GNU General Public License version 2**

This documentation is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this documentation; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

AUTHORS

CREDITS

© adam hyde 2006, 2007

Modifications:

Aleksandar Erkalović 2007

GLOSSARY

© adam hyde 2007

LINUX (SOURCE)

© adam hyde 2006, 2007

Modifications:

Salvatore Lionetti 2010

Thomas Middleton 2008

INTRODUCTION

© adam hyde 2006, 2007

Modifications:

Thomas Middleton 2008

LINKS

© adam hyde 2005, 2006, 2007

Modifications:

Stella Brennan 2005

Thomas Middleton 2008

Zita Joyce 2008

LIVE STREAMING

© adam hyde 2005, 2006, 2007

Modifications:

Stella Brennan 2005

Thomas Middleton 2008

PLAYLIST STREAMING

© adam hyde 2006, 2007

Modifications:

Thomas Middleton 2008

WHAT IS STREAMING?

© adam hyde 2005, 2006, 2007

Modifications:

Ruckert Martin 2006

Stella Brennan 2005

Thomas Middleton 2008

Zita Joyce 2008

TOOLBAR

© adam hyde 2006, 2007

Modifications:

Thomas Middleton 2008



Free manuals for free software

GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such

an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS