

ICECAST

Published : 2011-06-22

License : None

INTRODUCTION

1. Icecast
2. What is streaming?

1. ICECAST



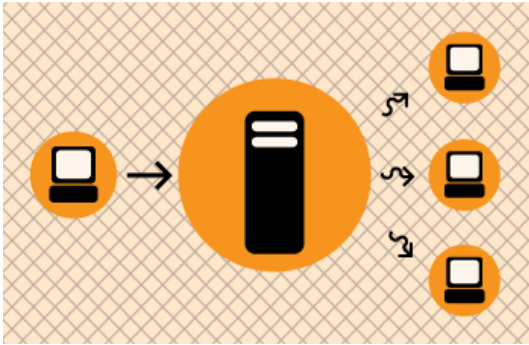
In the world of online audio and video there is a group of technologies and techniques referred to as **streaming**. Streaming, at its simplest, is the delivery of audio and video over the internet in **realtime**. Traditionally streaming has been used for delivering live internet radio, or for broadcasting events over the internet so you can watch them at home.

Streaming is often talked about using terms from the broadcasting industry as live internet audio and video are often seen as being analogous to broadcast television and radio. Hence you might talk about 'internet radio' to refer to live online audio. Sometimes people refer to sending live audio or video using streaming as 'broadcasting on the net' or as a 'internet broadcast'. These terms are helpful for helping us understand the intended purpose of streaming but don't take them too literally. However it is useful to refer to these models when explaining the role of Icast.

With broadcast television or radio there is transmitter that distributes the signal sent from the studio, to your television or radio.



With internet broadcasting (streaming) you replace the transmitter with a **streaming server**, which is really a kind of software.



Icecast is this kind of software. It enables you to distribute live audio and video across the internet in realtime. Note that this is only part of the equation. You also need a **stream encoder** that can send the original audio or video stream to Icecast. Icecast distributes the stream, it does not create the stream - that's the job of the encoder.

Icecast is a very mature technology and very robust. It supports the delivery of many different formats of streaming audio (including MP3 and Ogg Vorbis) and it supports the delivery of Ogg Theora for streaming video. Icecast runs on **Linux** and **Windows**. If you wish to install it on Linux you will need to have some experience with Linux; specifically you will need to know a bit about how to install software from the command line, and be comfortable editing text based configuration files. The Windows installation process is a little bit more straight forward but you still need to be comfortable editing text based configuration files.

Installing and running Icecast is not recommended for inexperienced users.

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).

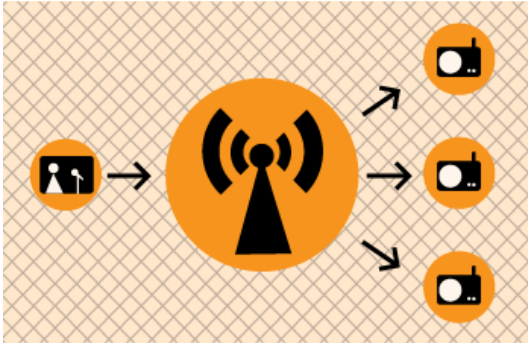
DELIVERY

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

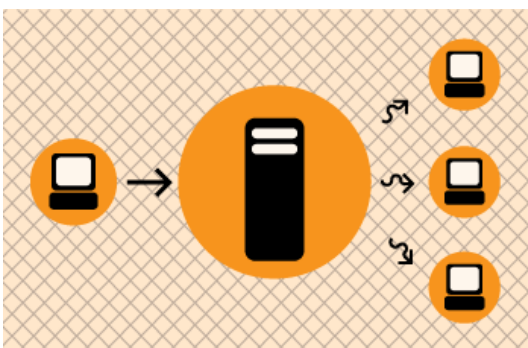
Perhaps a good way to understand what a streaming server does is to imagine a radio station. A radio station comprises of three components - a *studio*, a *transmitter*, and the *receivers* that your audience has.



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 Iccast on Windows
4. Installing Iccast from Source (Linux)

5. Installing on a Debian Linux

3. INSTALLING ICECAST ON WINDOWS

Software name : Iccast

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

Software version used for this installation : 2.3.1

Operating System use for this installation :Windows 2000

Recommended Hardware : 300 Mhz

Its not too tricky to install Iccast on Windows. You can install it on your own computer and have your very own streaming server to experiment with, or you can install Iccast on a Windows Server.

DOWNLOAD THE INSTALLER

The first thing to do is to visit the Iccast website and download the installer. So, point your browser at <http://www.icecast.org> and you will find the installer right on the front page:

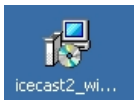
Icecast Release 2.3.1

We are pleased to announce the next release of Icecast.

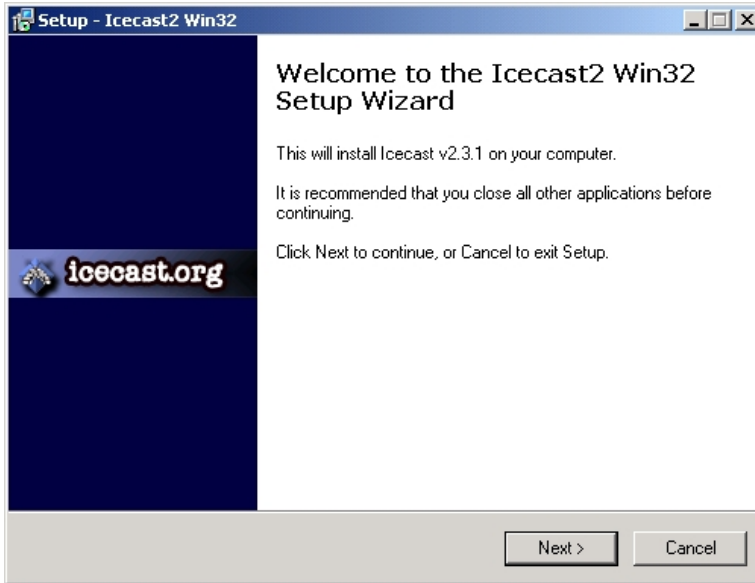
Downloads:

Source : [icecast-2.3.1.tar.gz](#)
SRPM : [icecast-2.3.1-0.src.rpm](#)
Windows Setup : [icecast2_win32_v2.3.1_setup.exe](#)

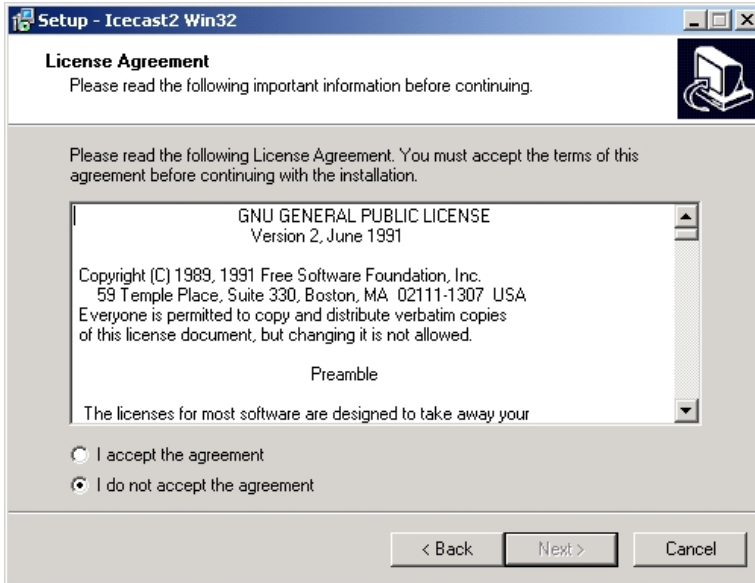
If you click on the Windows Setup link the installer will start downloading. Download it to your Desktop or somewhere else you will be able to find it. Once the download is complete then you will see the installer file on your desktop, or in the folder you chose to save it to. It should look something like this :



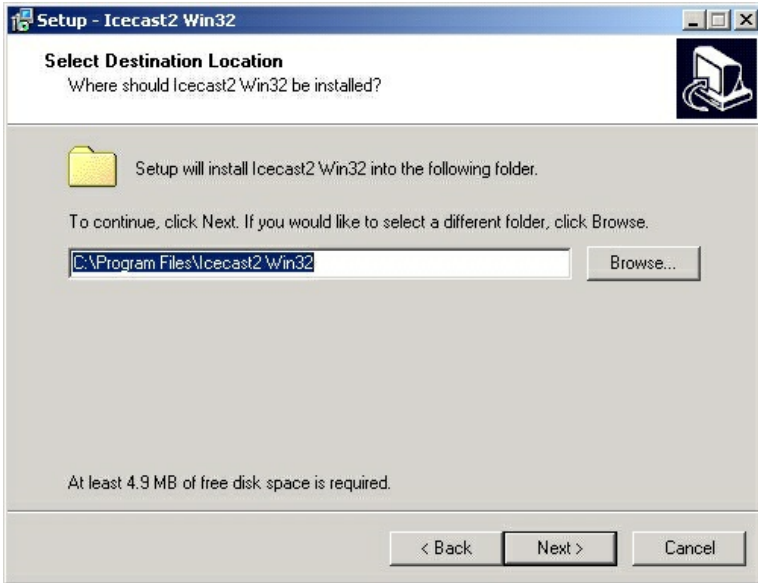
Now double click on the icon :



Press **Next >** and continue on to the license screen :



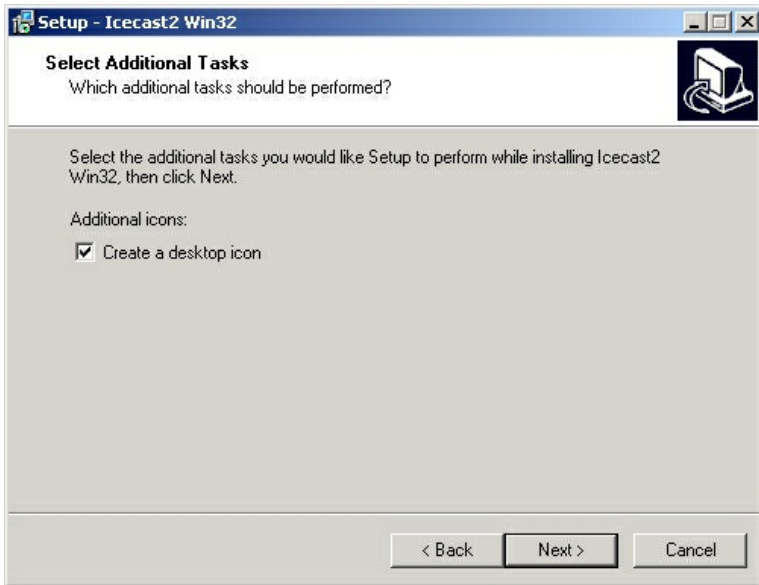
If you like you can read the license, or else just go right ahead and click on 'I accept the agreement' and press **Next >** :



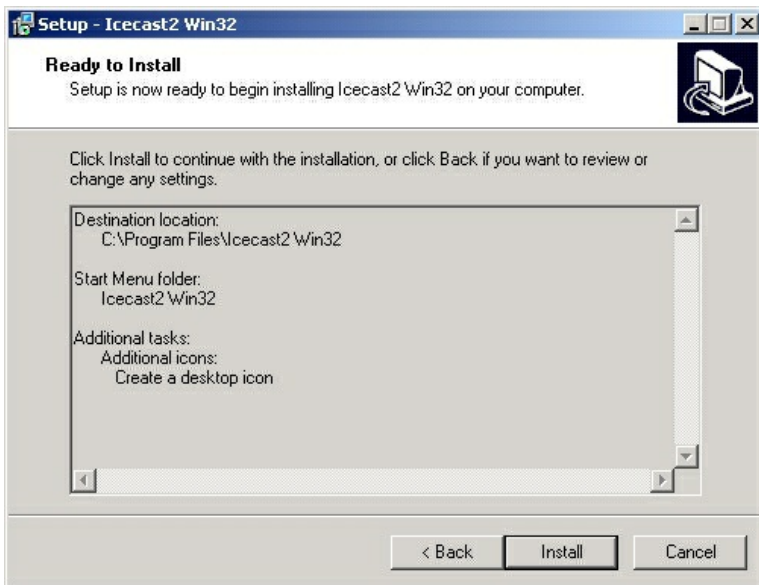
Ok, now we can save the program anywhere we like, you can change this by hitting the 'Browse' button and choosing another directory to save Icecast. For now we will stick with the defaults and press **Next >** :



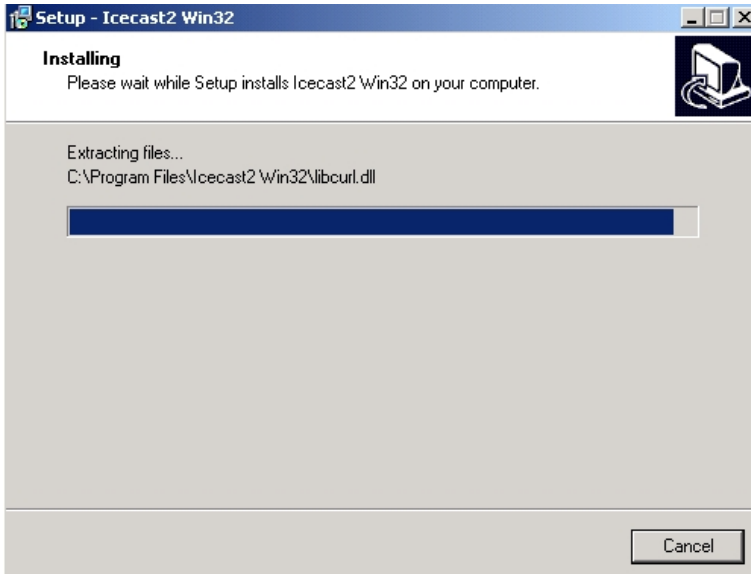
We can decide what the name of the software should be called in the Windows 'Start Menu' for quickly accessing Icecast. We will leave this setting at its default :



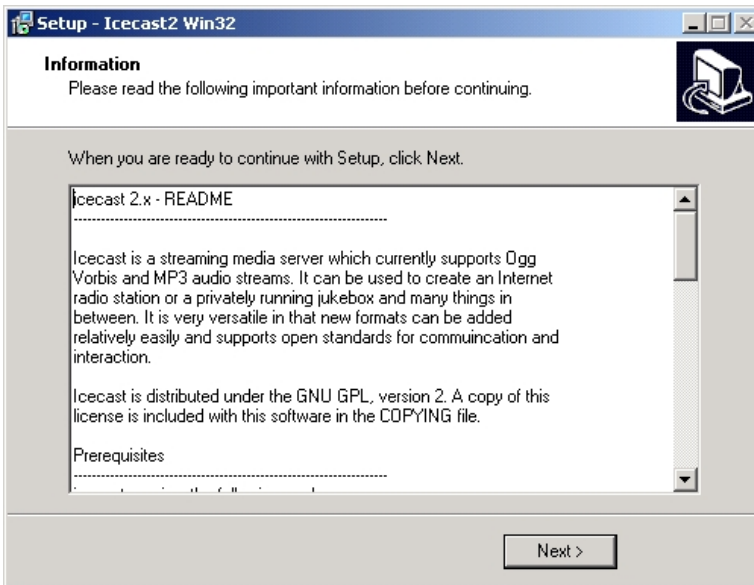
In the above window you are simply deciding if you want an icon to appear on the Desktop so you can quickly start Icecast this way. We will leave it at its default setting and click **Next >** :



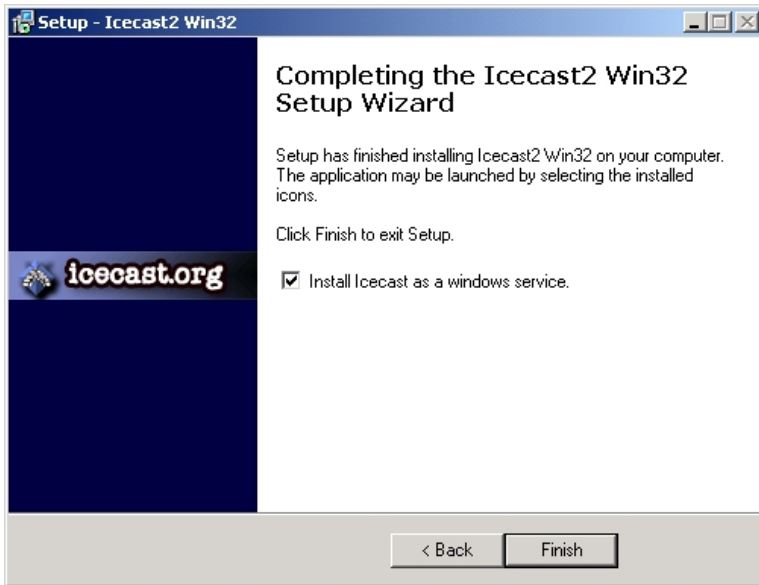
The above window is just a summary of your decisions made on the previous screens. Just check it and if all is ok then press **Install** :



The progress bar will whiz across the screen and when the installation is completed you will see this :



For the Windows installer this information is not so important, click on **Next >** :



This is pretty much the only real decision you have to make. A **Windows Service** means that you wish to install Icecast to run 'in the background'. This is known as a 'daemon' or as a 'service'. A server, like a streaming server, is usually a computer running streaming software (like Icecast) as services. If you have a server and this is your aim then leave the default checked, otherwise (if for example you want to experiment with Icecast on your Desktop or Laptop computer) unclick this option and press **Finish**

RUNNING ICECAST

To start Icecast you can double click the handy icon on your Desktop :



and you should see something like this :



4. INSTALLING ICECAST FROM SOURCE (LINUX)

This section covers the installation of **Icecast** on a **Linux** machine using the **sources**. This means you will need to compile the application using the command line.

Right on the front page of the **Icecast** website is a 'Download' link: click on this and you will be taken to the download section of their site (<http://www.icecast.org/download.php>).

At the time of writing the latest version is **2.3.1**, however it is more than likely you will be working with a different version. No matter, unless you are reading this in 2040 the install process should pretty much be the same. You might notice that there are different version of the software available - there are some versions available for **Redhat** for example, and there is also a **Windows** version. The file we want is what is known as the **source**. This is a single compressed downloadable file which contains all the installation files.

Download the **source** files. This is sometimes listed under 'all' in the 'Platform' sections of the **Icecast** site, but in any case you will recognise it because source files normally end in a **.tar.gz** suffix. In my example I will download the file (from <http://www.icecast.org>)

```
icecast-2.3.1.tar.gz
```

Ok... so I hope you have downloaded this file to somewhere you can access... normally I download installation files to my **home directory** in a directory called **src** . You can put your files anywhere you like as long as you can access them, although be a bit careful as you don't want to clutter up the file-system with installation files everywhere... I recommend putting it in your **home directory**. A **home directory**, by the way, is the directory with your **username** on it which is under the **/home** directory. To get to your **home directory** type the following in the terminal window:

```
cd /home/username
```

Where 'username' is your **username**, for example, my username is 'adam' so I would type:

```
cd /home/adam
```

or try this:

```
cd ~
```

The **cd** command means **change directory** and you use this command all the time to navigate around the file system. If you are in your **home directory** then look for a directory named **src** . To do this type :

```
ls -al
```

This command **ls** means 'list', and typing this will show you all the files in your home directory. If you don't see a **src** directory, then type the following:

```
mkdir src
```

The **mkdir** command means **make directory**, and typing the above will create a directory called 'src': now you need to change directories (**cd**) to the new directory. This is where you should put the sources if you have nowhere better to put them. If you have not yet downloaded the sources then download them to this directory, if you have already downloaded the sources then you can move the sources to this directory by using the **mv** command (assuming you know where the sources are)...

```
mv /home/username/icecast-2.3.1.tar.gz /home/username/src
```

In the above example I accidentally downloaded the sources to the **/home/adam** directory and I

am moving them to my new src directory.

EXTRACTING THE DOWNLOADED FILES

Now we will **uncompress** the file we have just downloaded... to do this type (assuming you have the

2.3.1 version of **Icecast**, otherwise substitute the below file name for the one that you have):

```
tar -zxvf icecast-2.3.1.tar.gz
```

This will throw a whole lot of garbage into the terminal window, which is a list of all the files that have just been extracted from the archive. To learn more about the **tar** command use this:

```
man tar
```

Now if you type the **ls -al** command you will hopefully see a new directory:

so now you need to **cd** to the new directory (in my case the directory is named **icecast-2.3.1**).

```
cd icecast-2.3.1
```

Then you need to type the following:

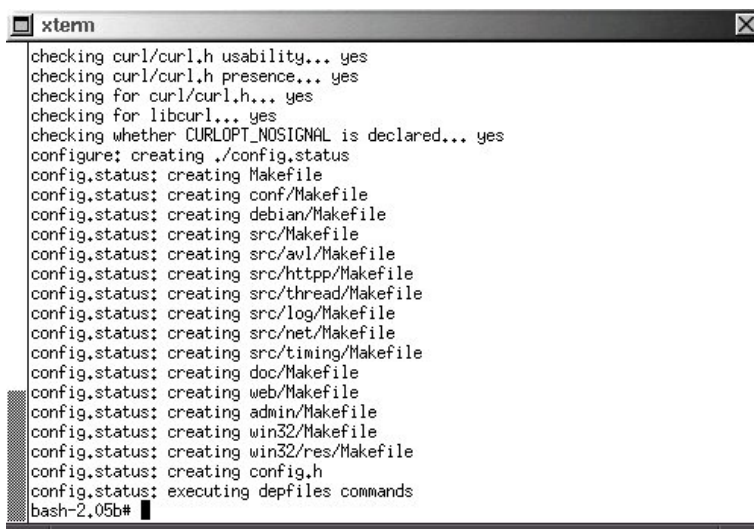
```
./configure
```

This is the standard **configure command** that you will pretty much use every time you install something from **source on Linux**. If you are lucky you will see a whole lot of text scrolling through the terminal... this is the **configure script** checking that everything is ok on your computer before it allows you to install the software. If all goes well you should have a nice clean configure process with no errors.

Now type the following:

```
make
```

The **make command** compiles a **binary** from the **sources**, basically what this means is that the **configure** script has worked out how to create the software for your system, then the **make** command uses this information to create (compile) the software. If the **make** process has gone well then you will end up with a nice clean terminal once again with no errors.



```
xterm
checking curl/curl.h usability... yes
checking curl/curl.h presence... yes
checking for curl/curl.h... yes
checking for libcurl... yes
checking whether CURLOPT_NOSIGNAL is declared... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating conf/Makefile
config.status: creating debian/Makefile
config.status: creating src/Makefile
config.status: creating src/avl/Makefile
config.status: creating src/http/Makefile
config.status: creating src/thread/Makefile
config.status: creating src/log/Makefile
config.status: creating src/net/Makefile
config.status: creating src/timing/Makefile
config.status: creating doc/Makefile
config.status: creating web/Makefile
config.status: creating admin/Makefile
config.status: creating win32/Makefile
config.status: creating win32/res/Makefile
config.status: creating config.h
config.status: executing depfiles commands
bash-2.05b#
```

ok! so... if you have had problems and this is not what you see then you are going to need help... I'm sorry I can't say much here other than perhaps you need to go to your favourite search engine and start searching for answers. The best method I have found for doing this is to copy the exact error message you receive and type this into the search field of your favourite search engine and see what results you get. Chances are you will find someone else has had the same problem and has documented the solution or perhaps someone else has offered advice in a news group etc. In any case, the problems that could occur are quite broad and it's not possible to predict them all here.

If all has gone well then there is one more command to type:

```
make install
```

This command installs the compiled software in the appropriate place on your system. To check this has all gone well type the following:

```
icecast
```

This is the command that starts your newly installed Icecast2 server...

5. INSTALLING ON A DEBIAN LINUX

Software name : Icecast

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

Software version used for this installation :

Operating System use for this installation :

Recommended Hardware : 200 Mhz processor (CPU) minimum, internet connection

The easiest way to install programmes on Linux machines is to use the **package manager** of the operating system. A package manager is a programme that oversees the installation and removal of other programmes. It can install the programmes you want automatically for you. On Linux distributions based on Debian (for example Ubuntu) the package manager is called **APT** (Advanced Package Manager). You can install the icecast server with apt by typing the following into a terminal. This means you will first need to know how to open a terminal (also known as 'shell' or 'command line'). When you have a terminal open type the following to install Icecast :

```
apt-get install icecast2
```

If you are not sure whether your linux has APT, it is safe to try this command, so just go ahead. If you get an error message, your Linux distribution is **not** based on Debian Linux, or it could also be that you are not connected to the internet (APT downloads the installer files from the internet). If you have trouble, first check your internet connection and then try :

1. install from source
2. use a different package manager. For example, on Gentoo Linux the package manager is called **Port**, so you would have to type "port install icecast2".

GETTING STARTED

6. Starting Icecast (Windows)
7. Paths and Logfiles
8. Starting Icecast in Linux

6. STARTING ICECAST (WINDOWS)

To start Icast you can double click the handy icon on your Desktop :



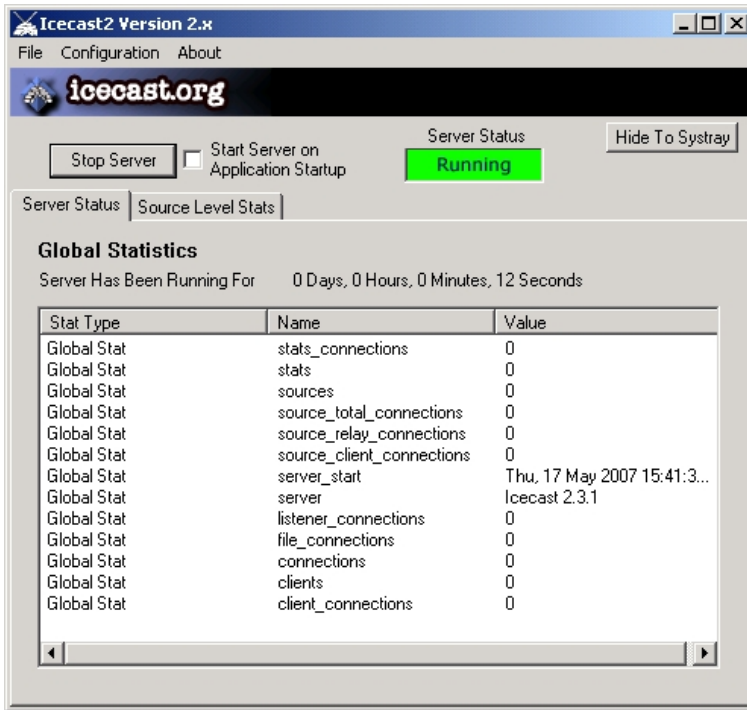
and you should see something like this :



Next you simply press the big **Start Server** button :



And Icast should start and look something like this :



If you want to try streaming to your new Icecast server then you can run a stream encoder on the same machine. If you want to do this then download and install **M3W** the stream encoder (<http://www.informatik.fh-muenchen.de/~ruckert/m3w/>).

7. PATHS AND LOGFILES

Sometimes I find that Icecast throws an error when it tries to create the log files. The error looks like this:

```
FATAL: could not open error logging
FATAL: could not open access logging
FATAL: Could not start logging
```

In this case I open my favourite text editor and start looking at the 'icecast.xml' file. I find these lines:

```
<logdir>/usr/local/var/log/icecast</logdir>
  <webroot>/usr/local/share/icecast/web</webroot>
  <adminroot>/usr/local/share/icecast/admin</adminroot>
```

So I change these values to point to my home directory like so:

```
<logdir>/home/adam</logdir>
  <webroot>home/adam</webroot>
  <adminroot>/home/adam</adminroot>
```

You would of course use the name of your own directory. Also change the values in the following part of the configuration file to read as follows:

```
<logging>
  <accesslog>access.log</accesslog>
  <errorlog>error.log</errorlog>
  <loglevel>4</loglevel> <!-- 4 Debug, 3 Info, 2 Warn, 1 Error -->
</logging>
```

And finally, in your home directory type the following:

```
touch access.log error.log
```

The touch command simply creates some blank files that icecast2 can then write to. Now I find that Icecast works fine. If you get a text editor and alter this password etc while Icecast2 is already running then you will need to stop Icecast2 (press ctrl and c simultaneously to do this) and then restart Icecast2 with the original command you used.

8. STARTING ICECAST IN LINUX

In Linux, Icast is controlled completely from the command line. This is very handy if you need to run it on a remote server which is very often the case.

So to start Icast the basic command is:

```
icecast -c icecast.xml
```

The trick is to find your 'icecast.xml' file. If you installed from source then you will find the file in the Icast2 source directory within the conf directory. The file will be called 'icecast.xml.dist'. If its not there then run updatedb and then do a **locate** or **slocate** to try and find the file. So let's say I am in the Icast2 source directory and the 'icecast.xml.dist' file is on the 'conf' directory, well maybe it would be easier if I moved it up to the directory I am working in and changed the name to 'icecast.xml'. I will use the cp command because then I leave a copy of the original just in case... so to do this I type:

```
cp conf/icecast.xml.dist icecast.xml
```

now I can run the command:

```
icecast -c icecast.xml
```

with luck it runs... the '-c' parameter here means 'config', so it is a way of telling the icecast command to use the config file 'icecast.xml'

If you find that there is an error then see the next section of configuring Icast2.

However if it is running then all you need to know is the default password is hackme.

For the purposes of running MuSE with the Icast server that you have just set up, use the following (all of which can be altered by editing the 'icecast.xml' file):

```
port : 8000
```

```
password : hackme
```

```
host : localhost
```

Then use any mount point you like, for example /firststream will work just fine.

If you do this, when you test the stream open xmms and enter into the 'open location' field (see section above) the following URL:

```
http://localhost:8000/firststream
```

and you should hear whatever is coming through your soundcard.

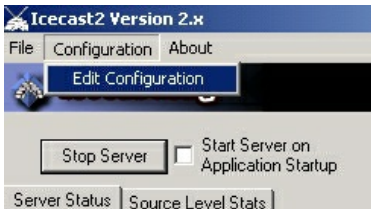
CONFIGURING

9. Configuring Icast (Windows and Linux)

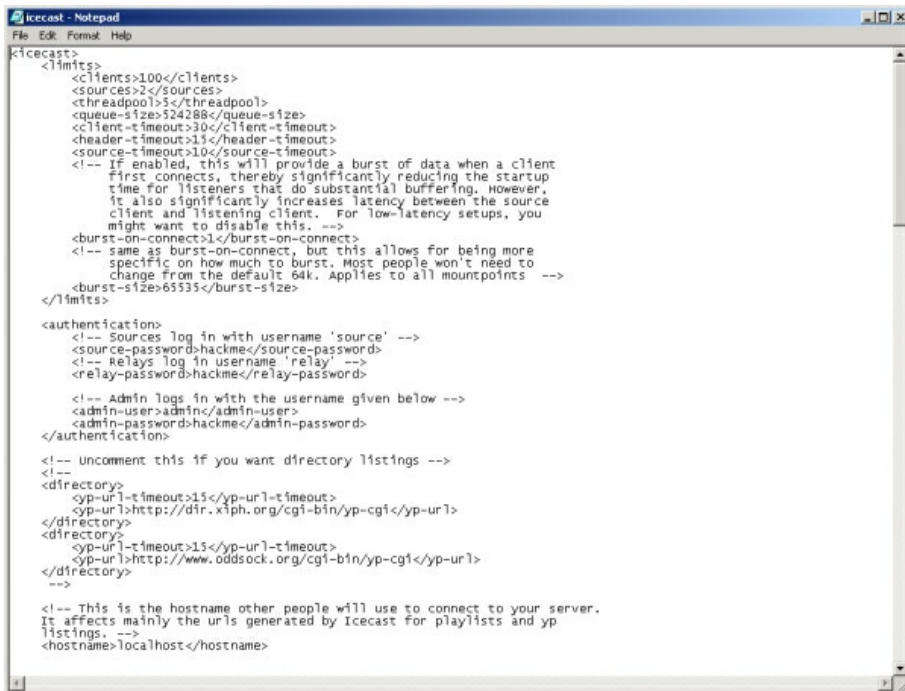
10. Creating Accounts

9. CONFIGURING ICECAST (WINDOWS AND LINUX)

On Linux you will need a text editor to change the configuration file. You will also need to know where on your operating system the configuration file is stored. On Windows its a bit easier. If you wish to edit the Icast configuration file in Windows you must first make sure that Icast is open but **not** running. Then you just need to click on **Configuration** on the menu bar and choose 'Edit Configuration' :



You will then see the configuration editor open in **notepad**:



Icast2 can be configured on Linux and Windows by editing the config file with a text editor. This document follows a certain syntax which is known as xml, so the Icast2 configuration file should be named 'Icast.xml'. If you have a file called 'Icast.xml.dist' then this is the example configuration file that is installed with Icast2. Rename the 'Icast.xml.dist' file to just 'Icast.xml'. Configuration File Format xml is a format that uses tags similar to HTML. The important thing to know is that each 'open tag' must have a 'close tag' otherwise the file will not be accepted by the server. A tag is something that is contained within an open and closed angle bracket like so:

```
<my-tag>
```

This example is called an 'open' tag, and each of these must have a corresponding closed tag.

Closed tags start with an open angle bracket followed by a forward slash, like so:

```
</my-tag>
```

So a tag that starts with '<' and ends with '>' is an opening tag, and a tag that starts with '</' and ends with '>' is a closing tag. Values can also be contained within these tag pairs. These values are the crucial parts of the configuration file that you will change to customise the configuration of Icecast2 to your needs. To change values of the configuration file change the text within the xml tags. For example, if you need to change the source-password, you will change the entry from this:

```
<source-password>hackme</source-password>
```

To something like this:

```
<source-password>mypassword</source-password>
```

COMMENTS

Comments are frequently used in the configuration file. Comments are simply remarks put in the file by the Icecast2 programmers to assist you in understanding the different sections of the configuration file. Each comment starts with ". This is the same syntax that HTML uses for comments. HTML Comments get ignored by the browser (they have no effect on the layout of a web-page), and consequently comments in this configuration file get ignored by the Icecast2 server. The below is an example of a comment you will find in the configuration file:

```
<!-- Uncomment this if you want directory listings -->
```

CHANGING THE CONFIG FILE

When you first start using Icecast2 there are just a few items within the configuration file that you need to worry about.

AUTHENTICATION

```
<authentication>
  <!-- Sources log in with username 'source' -->
  <source-password>hackme</source-password>
  <!-- Relays log in username 'relay' -->

  <relay-password>hackme</relay-password>

  <!-- Admin logs in with the username given below -->
  <admin-user>admin</admin-user>
  <admin-password>hackme</admin-password>
</authentication>
```

The fields above control the login processes made by any type of software attempting to make a connection to the Icecast server. The exception is that media players (like Winamp / XMMS etc) do not need to have a login, they just connect without any necessary authentication. You only (for now) need to be concerned with the source-password section of this file. source-password is the password that encoders must use to make a connection to the Icecast server. In order to send a stream the encoder must first identify itself to the Icecast server by sending the correct password. If the password is correct then the server will accept the connection and you can start streaming. If the password is incorrect, or if there isn't a password sent by the encoder, then the Icecast2 server will not accept the connection and you will not be able to stream. There is only one password for encoding software, so if there are, for example, 10 people sending separate streams to the server then each must use the same password which is listed here. The default source-password for all **Icecast2** servers is "hackme" so it's a good idea to change this to something else as a lot of people know this default.

HOSTNAME AND PORT

You don't need to know what a hostname and port are for now, you just have to know what these fields should be set as.

```
<hostname>localhost</hostname>

<listen-socket>
  <port>8000</port>
  <!-- <bind-address>127.0.0.1</bind-address> -->
</listen-socket>
```

Every Linux machine has what is known as a hostname. To find out what this is simply type hostname into the terminal and enter the result into the field in this section. For example if I type hostname in the terminal and I get the result "darkstar.net" then my hostname field in this section would look like this:

```
<hostname>darkstar.net</hostname>
```

The default port for Icecast2 is "8000", just leave it as it is for now. Note: you will need to remember these two values (hostname and port) as they need to be put into the corresponding fields of your encoder.

10. CREATING ACCOUNTS

written for **Icecast 2.3.1**

Icecast accounts in Linux and Windows are created by directly editing the configuration file with a text editor. Most probably you will be working with a config file on a server so it will be necessary to use a text editor in the terminal. Try **nano**, **pico**, or **emacs**, **vi**, **vim**. If you have no experience with terminal based text editors I suggest you try the first two.

If you want one password for all encoders (the **Icecast** default setting) you need to look at the following section:

```
<source-password>hackme</source-password>
```

In this situation anyone can create a stream on any **mountpoint** on your server using this **password** ('hackme'). If you want to change the **password** you simply edit between the tags and include a new plain text **password**.

You can however create individual **mountpoints** for individual users, requiring them to authenticate with their own **password**. For these 'accounts' you must stipulate the **mountpoint** to be used. The options for controlling the parameters of these type of **mountpoints** are quite extensive, and you can limit the number of listeners, the **bitrate**, where to save the stream as an archive, and even an 'intro' file to be played before a listener hears the stream (amongst other things).

Each account of this type is listed within `<mount></mount>` tags. If you want to start a very simple account where you give someone a **username** and **password** and a user can stream to the server then the configuration you would add would be as simple as this:

```
<mount> <mount-name>newuser.mp3</mount-name> <password>hackmetoo</password> </mount>
```

Unfortunately if you add **mount** 'accounts' like the above to the **configuration file** you will need to force **Icecast** to re-read the configuration file before the changes take place. You don't need to restart **Icecast** to do this but you do need to send it a **SIGHUP** signal from the command line. You do this by issuing the following command:

```
killall -HUP icecast
```

You can also add any number of the following parameters (the **mount-name** is always required):

- mount-name
- username
- password
- max-listeners
- max-listener-duration
- dumpfile
- intro
- fallback mount
- fallback-override
- fallback-when-full
- public
- stream-name
- stream-description
- stream-url
- genre
- bitrate
- type
- subtype
- hidden
- burst-size
- mp3-metadata-interval
- authentication-type
- on-connect
- on-disconnect

COMMON ACCOUNT SETTINGS

What we refer to as **account settings** are really settings specific to individual **mountpoints**. The following are some of the more useful settings:

password

the **password** required before any stream will be accepted by the server on that **mountpoint**. The password will be entered into the **password field** of the **encoder**, if the encoder is a command line encoder, then the **password** will be either entered in the command line or added to a text config file. If the encoder has a graphical user interface (GUI) then there is usually a form field in the interface for the password.

max-listeners

the maximum number of listeners that the **mountpoint** can support concurrently (at the same time). If you set the value to **50** for example, then you will be able to have up to **50** listeners or viewers at any one moment.

dump-file

the **path** and **filename** where the **stream** on this **mountpoint** should be **archived** (saved).

intro

this stipulates that an **intro** must be streamed on the **mountpoint** before the listener hears the **stream**. For example, if you had a live stream for an event, you may wish to play a short introduction announcement when the listener connects, followed by the stream. In this case you would use this option and include a link to the **intro** file to be played.

APPENDICES

11. Streaming
12. Related Links
13. License

11. STREAMING

12. 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.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>

MuSE.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>

MuSE.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/>

MuSE.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>

MuSE.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, MuSE.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/>

MuSE.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

MuSE.PureData and open source audio workshops (etc)

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

Tetsuo Kogawa

MuSE.MiniFM

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

<http://www.silenttv.net>

13. 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

BASIC CONFIG

© adam hyde 2006, 2007, 2008

Modifications:

Zita Joyce 2008

ACCOUNTS

© adam hyde 2006, 2008

Modifications:

Zita Joyce 2008

CREDITS

© adam hyde 2006, 2007, 2008

GLOSSARY

© adam hyde 2007

LINUX (SOURCE)

© adam hyde 2006, 2007, 2008

Modifications:

Ben Baker-Smith 2010

Zita Joyce 2008

LINUX (DEBIAN or UBUNTU)

© maxigas maxigas 2007

Modifications:

adam hyde 2007, 2008

Ben Baker-Smith 2010

WINDOWS

© adam hyde 2007, 2008

Modifications:

Zita Joyce 2008

INTRODUCTION

© adam hyde 2006, 2007, 2008

Modifications:

Zita Joyce 2008

LINKS

© adam hyde 2005, 2006, 2007

Modifications:

Stella Brennan 2005

Thomas Middleton 2008

Zita Joyce 2008

LOGS (LINUX)

© adam hyde 2006, 2008

Modifications:

Zita Joyce 2008

STARTING (LINUX)

© adam hyde 2006, 2008

Modifications:

Zita Joyce 2008

STARTING (WINDOWS)

© adam hyde 2007, 2008

WHAT IS STREAMING?

© adam hyde 2005, 2006, 2007

Modifications:

Ruckert Martin 2006

Stella Brennan 2005

Thomas Middleton 2008

Zita Joyce 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