

So you want to start an Internet Radio Station:

A Free Guide to planning, building and operating an online Radio Station.

This document is provided free of charge for non-commercial use by Radio Free Peterborough.

<http://radiofreepeterborough.ca>



Document Version 0.2 - Released 24 July 2007

This updated version was graciously sponsored by Kate Story

This document will tell you (just about) everything you need to know in order to set up and run an Internet Radio Station using any reasonably modern sound-enabled home PC. However, we are not lawyers. By viewing and using the information contained in this document you assume all responsibility and liability for what you do with the information contained herein. Any harm that may come to you, your spare time, your computer, your pets, your neighbours or nearby civilian populations as a result of using or misusing this information is your fault. You undertake any and all actions documented herein at your own risk. Be sure that you know the legal implications of what you are going to broadcast. If you don't own the broadcast rights to it, whoever does might not like you broadcasting it. Otherwise, have fun!

This document is Copyright (C) 2007 Steve McNabb / Radio Free Network - <http://radiofreenetwork.ca>. All Rights Reserved. Permission to use this document for noncommercial purposes is granted at no cost and distribution of

this document is encouraged. As long as the document is distributed in its entirety with this notice intact, this document is completely free for noncommercial use and may be copied, printed, emailed, posted on web sites, and distributed in any other manner provided no payment or obligation is required for its use, and no modifications are made to the document.

Radio Free Peterborough encourages users of this document to keep us informed of your progress. We want to see small-scale Internet Radio change the world. So get busy.

You don't need a supercomputer and thousands of dollars to start a station: just some time and gumption. Visit our feedback page or send email to steve@radiofreepeterborough.ca to let us know about your project when it's online, or if you find any errors or omissions in this document.

Introduction:

So you want to be the coolest kid on your block and run your own Internet Radio Station out of your living room eh? Good for you. It's a lot of fun, and can be a very rewarding experience. Especially if your focus is local music. Radio Free Peterborough started as an experiment to see how much local music we could assemble in one place to broadcast.

Pretty soon, it became the focus of rabid enthusiasm and community support. You never know where a good idea will go until you make it happen. So let's get started. This document will try to provide technology alternatives that are free wherever possible. Although we are not affiliated with any of these products or projects, we have used most of them and confirmed that they work as advertised.

What level of technical expertise is required?

If you are comfortable using your computer to surf the web, send emails, listen to music and you have some experience downloading and installing software you should be ready to roll. Don't be intimidated by all the switches and buttons in the windows control panel, or other software setup screens. They look scary, but if you just work through them slowly and carefully, you should be able to feel your way along.

What do you need to get started?

We're going to assume you're starting from scratch here, but that you at least have a reasonably modern computer with a sound card and a "high speed" Internet connection. Although you can still serve audio over a modem, the bandwidth available is pretty minuscule so the bit rate (audio quality- see below) you'll be able to broadcast at will be so low it's not worth bothering.

If you've got a sound-capable computer on a cable modem or DSL connection,

you already have all the equipment you need to start your own online radio empire. Some of the frumpier Internet Service Providers frown on people running servers on residential connections, so you might want to check with them first.

Technology Overview:

So how does this Internet Radio thing work anyhow. Well, here's a stab at a not-too-technical description.

Typical Internet Radio involves three pieces of software. Part one, is the Stream Server. This is a network application that feeds audio data to your listeners. It doesn't generate any audio on its own - it just takes incoming audio data from the Stream Source (part two) and broadcasts it to your listener's Player Software (part three).

Part two is the Stream Source software. This is how the audio data gets from your sound card to the Internet. The Stream Source encodes the audio data as digital information that the computer can then hack into many tiny pieces and start shooting those pieces out onto the Internet where they are (hopefully) accepted and relayed by the Stream Server for distribution to listeners.

Part three is the Player Software. This is the "home radio" part of the setup. The Player goes out on the Internet, connects to the Stream Server and requests a particular stream. Once data starts to flow, the player assembles the "pieces" of audio data in the right order, and starts playing them back as audio through the sound card.

Streams are identified by their "mount point" - things like "/live" and "/hi-fi.ogg" (see how they start with a forward slash "/" character?). Ogg Vorbis stream mount points must always end in ".ogg" Assuming the stream is operational, and there are available "listener slots" free on the server the stream will start to buffer data, and should start to play audio within 5-20seconds after connecting. A complete radio station "address" URL would be something like:

`http://yoursite.ca:8000/some-stream.m3u`

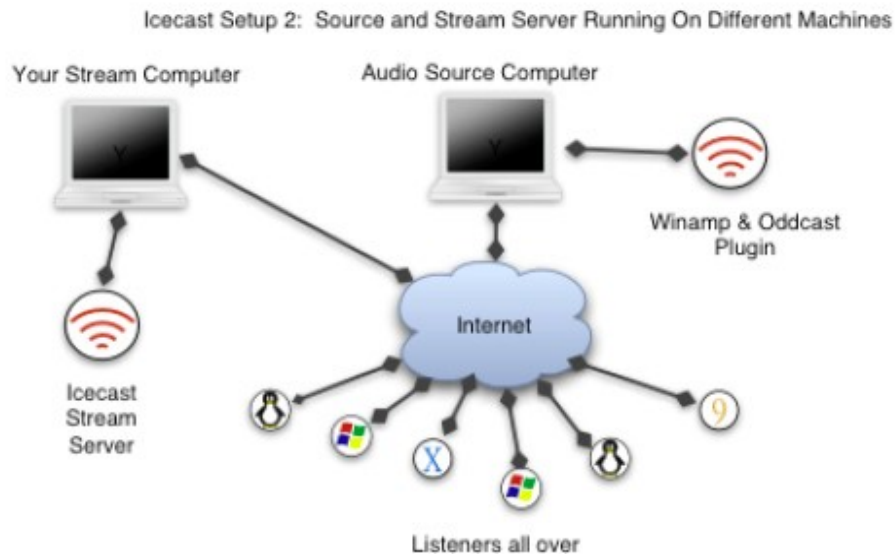
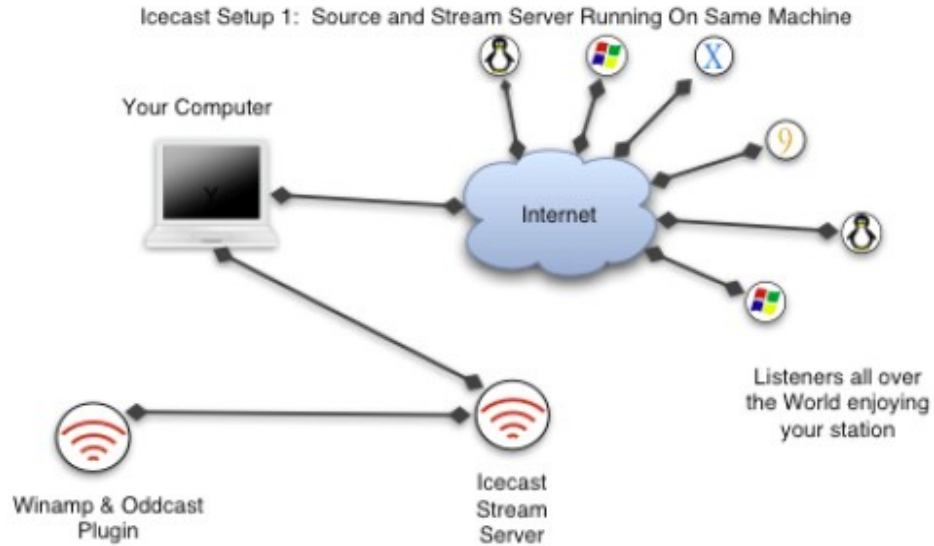
The "8000" tells your player that the audio Stream Server is set up to take connections on port #8000. A port is analogous to the entrance to a building: stuff can go in, stuff can go out. This is how your computer connects to anything on the Internet: it opens a port on your computer, connects it to a port on some remote machine and starts exchanging data. Port 8000 is the "standard" port used for streaming audio services. You can use any port above 1024, but there's no good reason not to use the default 8000 for most people.

To compare it to "real world" radio, the listener's computer and Player Software are analogous to your home radio. The Stream Source is like the mobile broadcast truck broadcasting "Live and on location" - from your living room. The Stream Server is analogous to the radio station's master control room where they hook up the signal from the mobile truck to the broadcast tower so it can be

rebroadcast to the masses.

Below you'll see a diagram that visually represents the two ways you can setup your Winamp & Oddcast -> Icecast network. Either the Stream Server and the Stream Source can run on the same machine, or on two different machines. Those two machines can be anywhere in the world, provided that there is a reasonably fast network between them. So, you could have the "main" server running on your cable connection in Peterborough, but run the Winamp/Oddsock on a computer running at your friend's house in downtown Boston. Your friend is having a huge party with live bands, and your friend wants to broadcast this music live to some friends around the world who are too far away to make it to the party. No problemo. You and your online station can spread that music anywhere it needs to be (assuming there is Internet connectivity there of course).

The following diagram shows you how you connect your audio streams to your listeners with the Stream Server and Stream Source running on the same machine, or networked over the Internet. In ideal circumstances, the audio will pump out the station after a delay of 20-45 seconds of encoding time. The computers are doing a lot of math to make all this happen, and it takes a moment or two for the process to get rolling.



Stream Formats:

There are two main flavours of audio stream. These two flavours are “mp3” and “ogg vorbis”. These are two different technologies used to compress the audio data so it is much smaller with little or no perceptible loss in quality (depending on the bit rate used for encoding). There are reams and reams of information online about the specific advantages of one over the other, so we’ll leave you to research that for yourself if you wish. In general, however, mp3 is more widely supported, and is great for mono, but ogg vorbis sounds a lot better in stereo, and is “Free” software - whereas mp3 is patented and may require a license in some countries. As far as we know as of the time of writing this document, no

license is required to broadcast Internet-based mp3 data in Canada.

Bit rates and Sample rates:

Two technical terms you're going to see a lot of are "bit rate" which is generally 32, 56, 96 or 128 Kbps (Kilobits per second) and Sample rate which is usually represented in kiloHertz. For CD quality sound, we want 44 kHz. That means that for each second of audio there are 44,000 'chunks of sound' that get played back Very Fast when the stream is playing. The bit rate defines how much data should be allowed to stream out per second. The higher the bit rate, the more data can pass out through your network card onto the Internet, the better the end product sounds. There's no good reason to use bit rates above 96Kbps unless you are on a superfast connection with a few thousand dollars worth of high-fidelity stereo equipment plugged into your computer.

Bit rates also help you figure out how many listeners you can expect to host without making your Internet connection unusable for your daily surfing needs. See "**Bit rate and bandwidth calculations**" in the **Procedures** section below.

Planning:

What you're planning to do with your station is going to define how you'll have to set everything up to make it all work. For example, if you are planning to do "On Air" live d.j. voiceovers, you'll need a microphone & headphones. Any old microphone will do, as long as it will fit into (or can be adapted to) a "mini phone" jack - the little plugs on the back of your sound card.

Unless you are going to be broadcasting just your own voice through the microphone, you're going to need some digitally-encoded audio files: mp3 or ogg. You're better off to assemble the catalogue of stuff you want to play before you do anything else. Then once you get your tech worked out, you're ready to roll right away.

Software To Download:

Now that you know a bit about how the technology works, let's get started with the software we're going to need - If you remember from above, the three pieces of software we'll need are a player (Winamp), a stream server (Icecast2) and an audio source (Oddcast). The first piece of software we're going to download is the player. Once we know the player is installed and working correctly, we can start to build up the rest of the station around it.

Make a special directory to store all the station-related programs you're going to download in one place. You're going to have enough head-scratching to do without looking all over hell's half acre for your files. First we're going to get a player. We're downloading the player first so we can make sure that the sound card is producing audio properly. Windows users, point your web browser at

1. First, point your web browser at <http://www.winamp.com>.
2. Next, find the "download" button
3. Scroll down through the feature comparison table until you find the "Get Basic" (Free) and click
4. Grab the "Full" or "Bundle" version - the Bundle version has some extra goodies that you may or may not want. Regardless, you don't need them for broadcasting
5. Find the downloaded Winamp and double click it to install it.

Mac OSX users, iTunes will handle mp3 stream playback just fine, but doesn't support ogg vorbis out of the box (at least not yet). A suggested alternative for ogg on OSX is theogg plugin for quicktime. You can find instructions at this URL : <http://jsp.vs19.net/osx/oggtunes.php>

Linux/Unix users, you should install xmms. For console-only junkies, try out MPD -<http://musicpd.org/> Newer versions of mpg123 also support mp3 streaming.

Once you have Winamp or your player of choice installed and able to play music out of your speakers or headphones, we're ready to move on to the next step: the stream server software Icecast2.

You should download the appropriate version of Icecast for your operating system from [http:// www.icecast.org](http://www.icecast.org).

For Windows users:

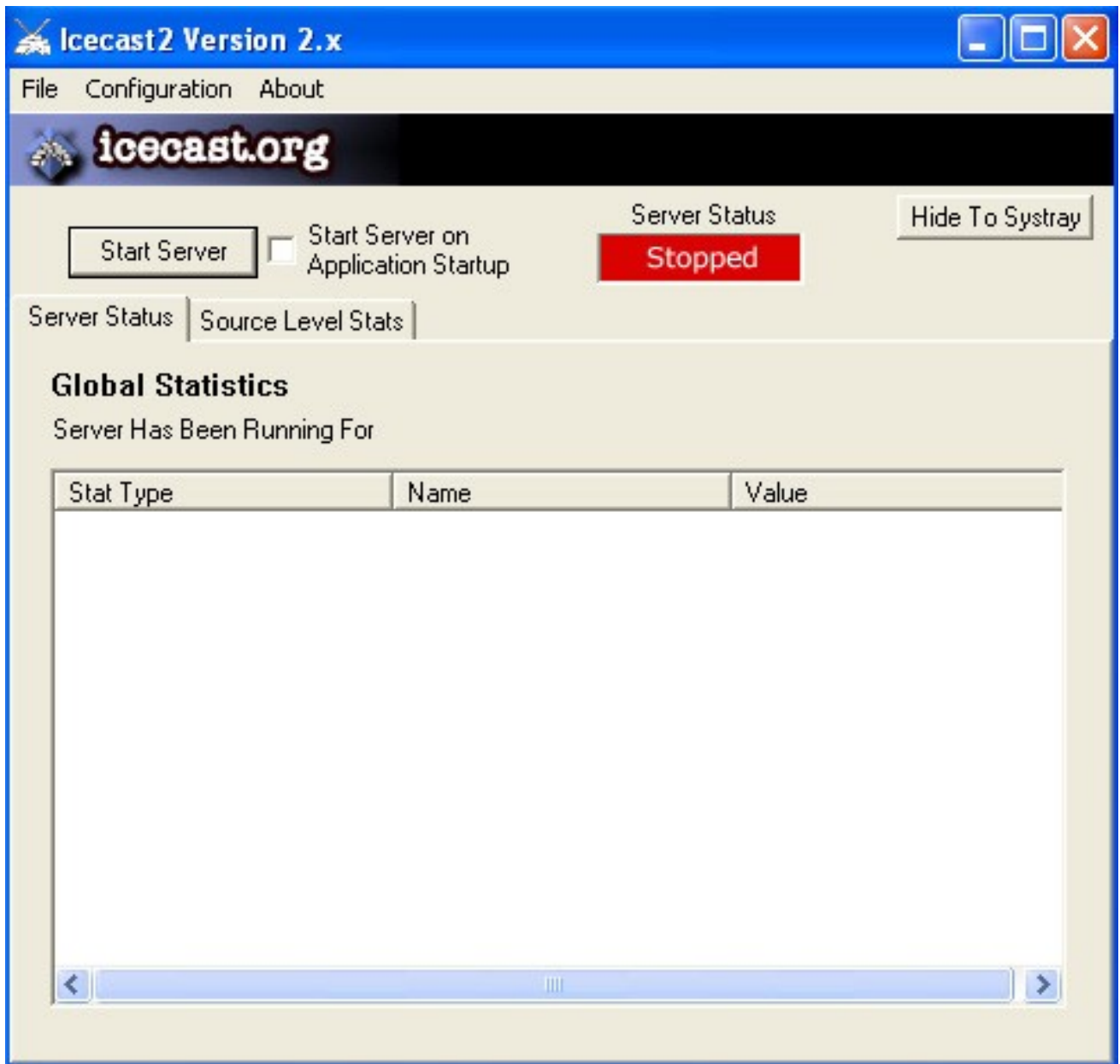
1. visit <http://www.icecast.org/download.php>
2. click "icecast2_win32_v2.3.1_setup.exe" next to "Windows NT/2000/XP"

There are versions for Windows (2000, XP, etc) and Linux. There are instructions for compiling Icecast under OSX at <http://icecast.darwinports.com/>

Download and install the Icecast2 package by double-clicking on it in the folder where it was downloaded. Once you have Icecast2 properly installed, you should see a nice little Icon for Icecast2 - an igloo with antennae poking out of it.

Fire up Icecast2 and get yourself a cup of coffee - this next part is a little bit confusing, but you can handle it.

Start up the Icecast program , and you should see a screen appear with a red "Stopped" server status box. It should look something like this:



Before we can start up the server, we have to tell Icecast what options we want to use.

Select "Configuration" from the menu bar across the top of the Icecast window and click "Edit Configuration". This will bring up the Icecast.xml configuration file in a text editor. This xml file tells Icecast which options and settings to use to run your station. That's what all those pointy brackets are there for. It's very important that you not touch ANY of the pointy bracketed sections - only edit the content between them. For instance, to restrict your server to 5 listeners, you'd change the default setting in the Icecast.xml by changing the line:

```
<clients>100</clients>
```

to

```
<clients>5</clients>
```

If you muck up the start `<tag>` or end `</tag>` of any element (like `clients`) then Icecast won't start, and you'll have to start all over again with a fresh copy of `Icecast.xml`. Caveats aside, let's look at the settings. We're not going to examine all the settings, only the ones we really need to change to get a station up and running. There is plenty of documentation on the Icecast web site at <http://www.icecast.org> that covers all of the settings in detail. The first setting to look at is the "hostname" setting. Look for a line that looks like this:

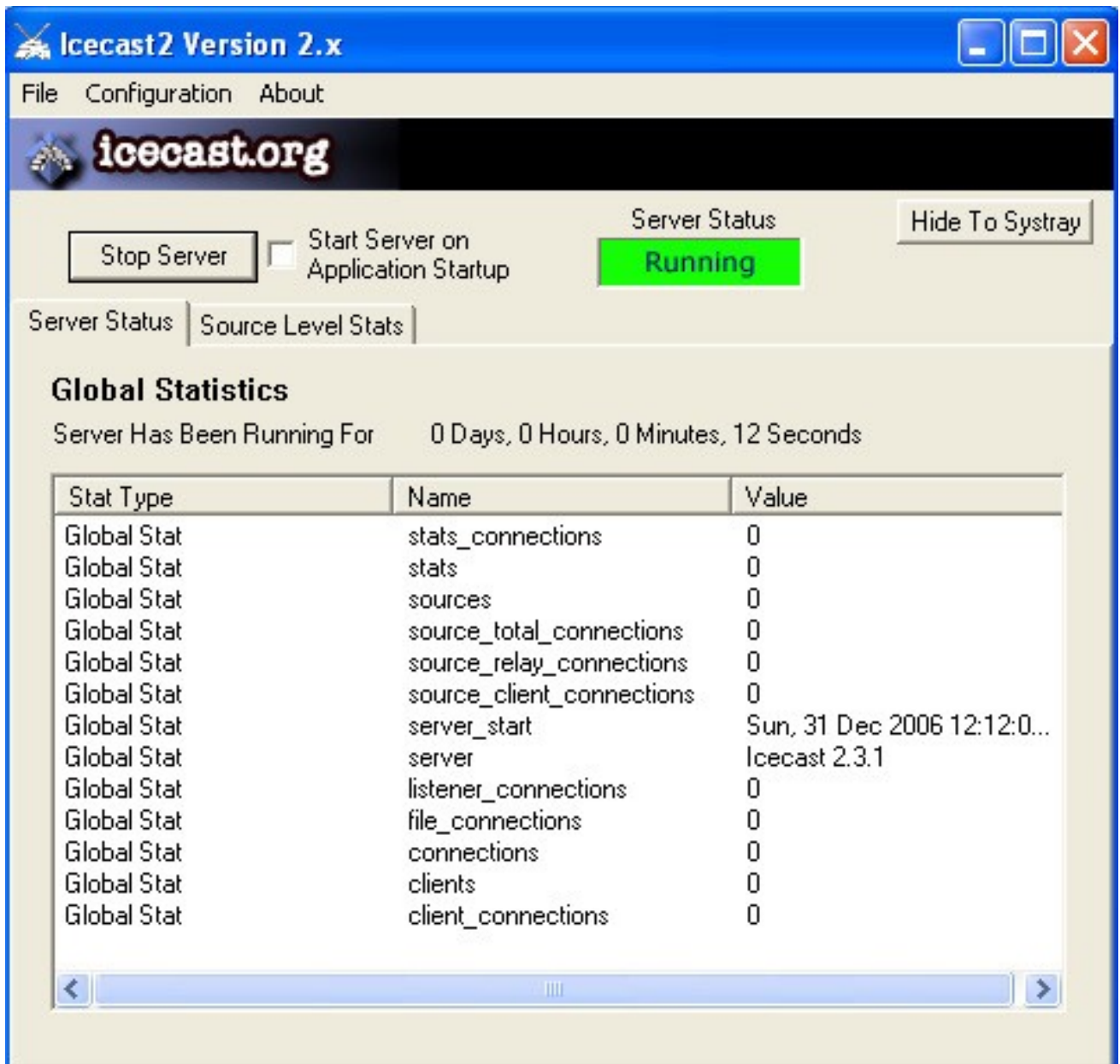
```
<hostname>something here</hostname>
```

If you have a domain name pointing at your address see ***how do I get a cool domain name?*** in the **Procedures** section below), insert the domain name here. If you are behind a router (like a linksys wireless router) you'll also have to configure the router's port forwarding setup to forward port 8000 on the machine running your Icecast stream server. You can also use your IP address (see ***how do I find out my IP address?*** in the **Procedures** section below) for the hostname. If you only want to use the station from inside your own home network, you could use "localhost" or "127.0.0.1" as your IP address. Don't use localhost or 127.0.0.1 if you want anyone from the outside world to be able to hear your station.

Change your source and relay passwords in the `<authentication>` section to a password of your choosing rather than the default 'hackme' . If you don't want people to be able to download the files they are listening to (like a free music store) then change `<fileserve>1</fileserve>` to `<fileserve>0</fileserve>`. A good idea anyhow unless you have A LOT of bandwidth. You do not need to modify the `<mount>` information here. Just leave it as-is. You only need to fiddle with those sections if you want some of the more advanced, per-stream features. If that's what you want, the Icecast documentation would be the place to start. You'll be adding mount point information via the Oddcast source program in a minute. The mount point serves as a way to "connect" to the data being sent by the source (Oddcast) to the listener (Winamp) via the stream server (Icecast).

Now save the `Icecast.xml` file, close and restart Icecast.

Now click "Start Server" and if all is well, and the `Icecast.xml` file is correct, Icecast should show you a nice green "Running" server status indicator. Now click "Hide to Systray" to get Icecast out of your way. Congratulations. You're now 2/3 done. You should see something like this:



Next we need a stream source. For Windows users, go grab the Oddcast Winamp DSPplugin from <http://www.oddsock.org/tools> and install it.

1. visit <http://www.oddsock.org/tools/oddcastv3/>
2. grab the "Winamp" version "Download Current Version - Winamp 3.1.12"

For Linux users, get Ices for mp3 streaming or Ices2 for ogg streaming. Both are available at <http://www.icecast.org> - just rtfm. If you want to do mixed live streaming and mp3/ogg tracks, Darkice is probably your best bet under Linux.

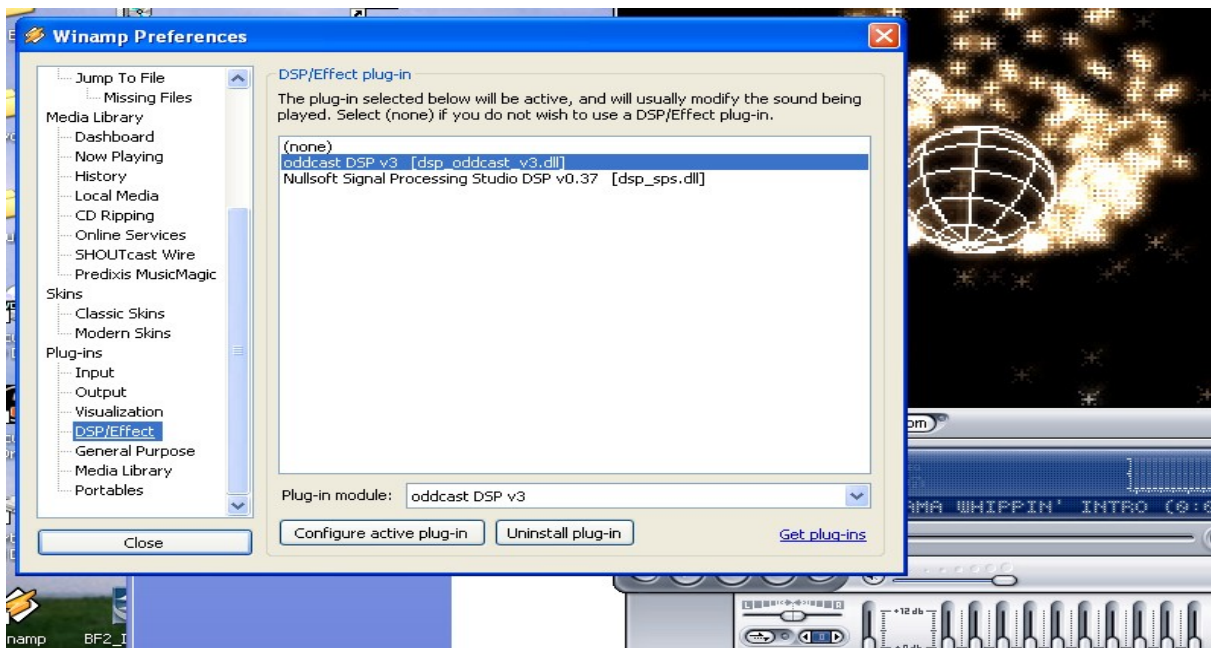
OSX users, although you may be able to get Ices or Darkice to compile, you might prefer to just drop the \$60 on “Nicecast”. Nicecast will let you redirect the audio output of any application to an Icecast server. Just set “Icecast2” for the protocol.

Since the vast majority of home PC’s run a Windows variant of some kind, the specific instructions from here on in are for Windows users running Icecast, Winamp and Oddcast.

So now we hopefully have our Stream Server(Icecast), Player(Winamp) and Stream Source(Oddcast) all installed. We’re pretty much done. All that’s left is to do our final setup.

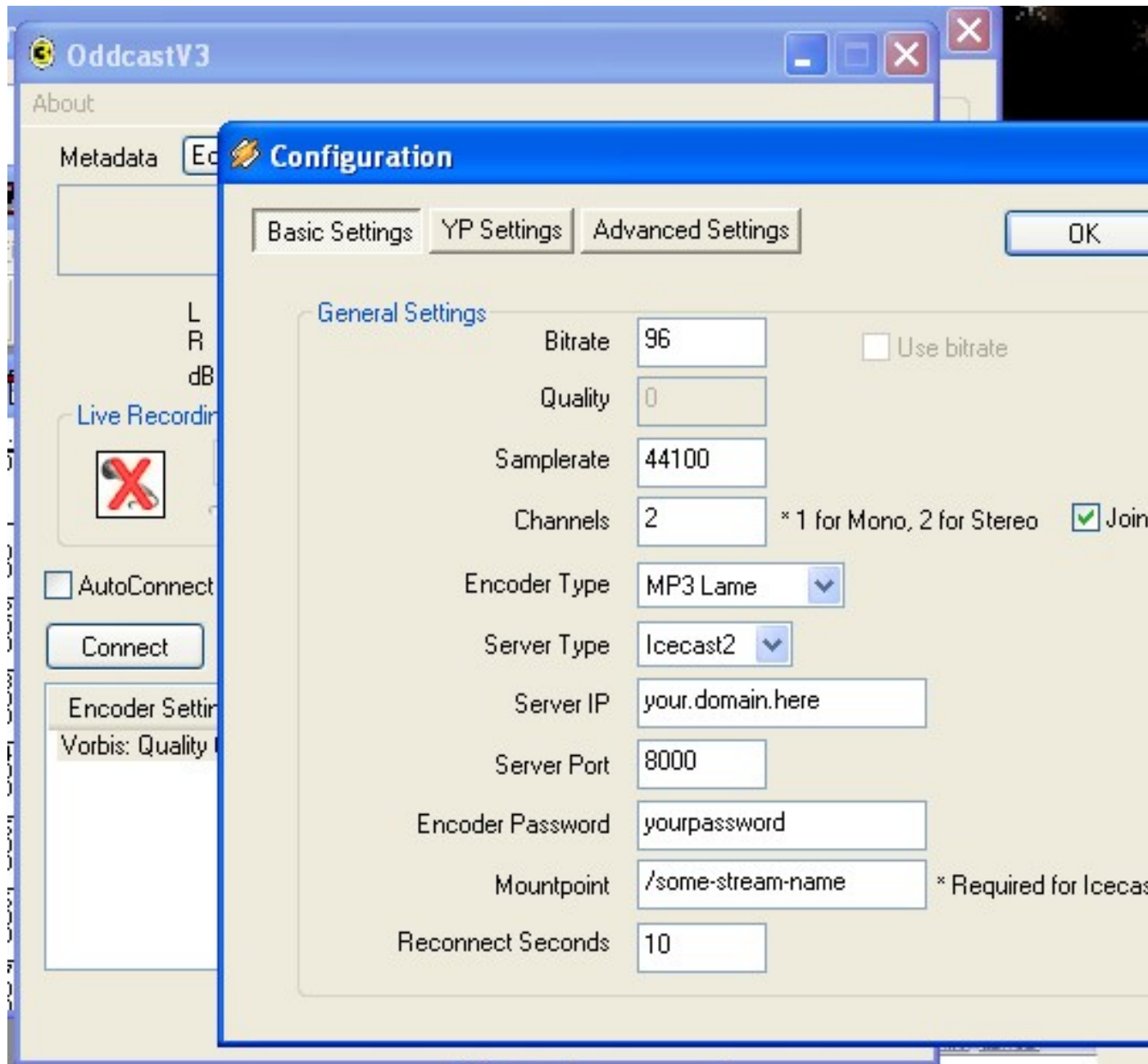
We need to tell Winamp that we want to use the Oddcast DSP plugin. This plugin reroutes the audio data from your sound card into a network-transportable data stream and sends it to the Icecast server.

To set this up, right-click on the main Winamp player window (where the play/fast forward/etc controls are). Select Options->Preferences from the menu, then DSP/Effect from under the Plugins branch of the menu on your left. You should see oddcast listed in the panel on your right. Select it, and the oddcast panel should pop up. You can now hit the “Close” button on the winamp preferences panel.



On the oddcast preferences tabs, find the one with server setup information on it. Enter the IP address of your Icecast server, or localhost if it’s set up on the same

machine, and you've set up Icecast.xml to bind to localhost.



Next, set the server type to "Icecast2" and pick a name for your mount point. Bear in mind that if you want to create an Ogg Vorbis mount point, it must end in ".ogg" - so /stuff.ogg would be a valid Ogg mount name, but /stuff would not. If you are using mp3, this doesn't apply. You can name your mount point anything you want, as long as there are no spaces in the name, and it begins with a "/" - so /test/somestuff-here and /my-yodeling-station are all valid non-ogg stream names. If you wanted them to be Ogg Vorbis streams, they would have to be /test.ogg /somestuff-here.ogg and /my-yodeling-station.ogg

Set the password to whatever you set Icecast.xml's source password option to earlier and try to connect Oddcast to the Icecast server. If all is well, you should see a little green light on Oddcast's "Connect" button. You should be online. Assuming you left the port setting at 8000 in Icecast.xml, you should be able to point your web browser at <http://your.server.ip.address:8000/status.xsl> and see some information about your stream. Hopefully you should see the title of the stream, the number of listeners (probably 0 for the moment) and the song title.

Now you have to tell Windows which audio source you want to "broadcast" from your computer. You can only broadcast from one source at a time without an external hardware mixer. To access your recording options, select Audio Options from the Windows control panel. All versions of Windows call it something different, but it should be something like Audio Options or Audio Devices or some such. Open that up and you should see a control for the volume. Look nearby for an "Advanced" button. Click that, and you should get a long panel of sliders. One for each audio-producing device on your machine: MIDI, Wave Audio Playback, CD, Line out, etc. In the upper left corner of this window is a menu item "Options".

Click Options and select Properties. This should open a panel with your sound card properties. Now change the radio button titled "Adjust volume for" from "Playback" to "Recording". Now all those sliders are for setting **Input** volumes instead of the usual playback volume sliders. These are analogous to the faders you would see on the soundboard in a regular radio station. Only one of these will be enabled at a time, and that is what will go out through the Oddcast stream to your listeners. To play audio back from a regular audio CD, pop it in the CDROM drive, press play and then tick the box under the "CD" recording level slider. To break in with the microphone and talk, check the box under microphone and start talking into your mic. You should probably test the microphone setup with something simple like windows sound recorder to make sure it's working properly first before pulling your hair out with Icecast and Winamp/Oddcast.

It's a bit of a clunky setup if you wanted to talk much and play music too, since you can't mix the two together this way, but hey - it works, it's free, and it's reasonably easy to set up. If you wanted to get all fancy pants and have a professional-style setup, you could patch the line-out from your sound card into a small mixer with your microphone(s) and blab to your heart's content while mixing in music. Just patch the line out of your mixer into the line in on the sound card and set the recording input to 'Line in' by checking the button under its slider. If everything went well with install and setup, you should now be ready to broadcast whatever you want to the world.

Once your station is up and running, you can visit the default status page at <http://your.ip.or.domain.name:8000/status.xsl> - this will show you all the active mount points with their listener statistics, as well as provide a rudimentary interface for administration of the streams.

By visiting the status.xsl page you can see how many people are connected to

your stream. You can also find the link there to cut-and-paste for addition to your web site. You should now be online and pumping out some audio goodness for the world to share. If you get stuck, double-check that everything is set up as you think it is. If you get really stuck, go for a walk, and/or ask someone a little more nerdy to help you out.

Your listener's player may "crap out" after a short time playing - if this happens, try having your listener bump up Winamp's buffer size (see Winamp's help) and turning off any other programs that may be eating up CPU power and/or network bandwidth. If the problem persists, your network may not be fast enough to handle the bitrate you have selected. Try a lower rate (say, 48k or 32k) and see if that stabilizes it. Sometimes it takes a bit of fiddling around to get the right combination of bitrates and bandwidth.

Finding Help:

If you are having trouble getting your setup working Just Right, try looking at the Icecast log files. These log files will be located in the directory where you installed Icecast, in a subdirectory called "logs". If you installed Icecast in the default location, your logs would be found in C:\Program Files\Icecast2Win32\logs\

When Icecast isn't set up quite right, it will usually tell you what is wrong by writing a message to the error.log file. For example, if you used "fishes" for your source password in the icecast.xml file, but typed "fishess" in the password field of oddcast, the error log would tell you that the passwords don't match.

Many communities also have computer user's clubs that are easy to find online. Make a posting on a user's group bulletin board, or show up at a meeting. If you show that you've done your homework, and that you're just stuck on some technical point, you'll be surprised at how helpful people can be.

Procedures:

Bandwidth and Bit rate calculations - how many listeners can I support?

The bit rate of your connection is likely not the same in both directions. Most consumer Internet connections are asymmetrical (fast downloading, but slower uploading). Use your outbound connectivity for all bit rate calculations. For example, if you had a typical "highspeed" DSL connection, you'd likely get 800 Kbps outbound. So, if you had a high-fidelity mp3 stream at 96Kbps, you'd probably want to limit your station (via icecast.xml) to 6 listeners. That'd use up 576 Kbps at full capacity, leaving 224 for your other uses. This methodology assumes that you're rarely going to have 6 listeners all at once. When you do, your network will be slower. Don't leave Icecast set up to handle more data than your connection can reasonably deal with or your surfing will slow to a crawl.

Some other example calculations:

4 x 32 Kbps stream listeners, and 3 x 96Kbps stream listeners
 $(4 \times 32 \text{ Kbps}) + (3 \times 96 \text{ Kbps}) = 416 \text{ Kbps}$

10 x 56 Kbps stream listeners
 $10 \times 56 \text{ Kbps} = 560 \text{ Kbps}$

etc.

You can have a lot of lower-quality streams, or fewer high-quality ones, or mix and match like Radio Free Peterborough does to accommodate listeners with both high and low bandwidth connections.

How do I find my IP address to give to people?

Your IP address identifies your computer (or network) to the rest of the “outside” world. This is how the servers you visit know where to send the data back to. An IP address consists of four groups of numbers from 1-255 separated by dots. Examples are 142.77.1.1 and 33.13.223.1

If your ip address were 123.123.123.123 and you had set up your Stream Source to mount on /live (using the default port 8000) then your stream information would be at:

<http://123.123.123.123:8000/status.xml>

You’d find the link to listen to the stream on that page.

If you are behind a router (like a linksys wireless gateway for instance):

Log in to the router’s administration system (usually found at 192.168.1.1) with your web browser and find your IP address in the page that tells you the status of your Internet connection. Don’t forget to set up port forwarding on your router to point port 8000 at your Stream Serve rmachine running Icecast.

If you are connected directly to a cable/DSL modem via Ethernet:

Open the Network Settings panel from the Control panel, and look for the TCP/IP protocol bound to your Ethernet card. Click that and you should get a configuration panel that will tell you your IP address. You can also search for the program “Winipcfg” on most windows machines. Select your network card from the dropdown list, and you should see your IP address.

How do I get a cool domain name like radiofreepeterborough.ca so people don’t have to remember my IP address to hear my stream?

Visit <http://www.dyndns.org> and set up a free account. Then create a hostname

and keep the record up to date. If you want to have a “.ca” address or “.com”, you’ll have to pay for it. R.F.P. doesn’t recommend any particular commercial dns service providers - shop around for the best deal.

How do I publish my stream in the public stream directories to let other people know about it for free?

All Icecast servers come with a built-in ability to announce themselves to a set of international server directories. If enabled, these directories list your stream along with many many others. This enables people who are looking for Internet Radio to find streams by genre, title, country, etc. Publishing your stream WILL get you many more listeners. Make sure you are all set up and have been running properly for a few days before enabling the <Public> directive in the Icecast.xml by setting it to <Public>1</Public>. The public stream servers are free of charge for both stations and listeners. Cool eh?

The End. Or, the Beginning.

I hope you found this information useful. If you have any additions or corrections, suggestions or comments don’t hesitate to contact me by email at steve@radiofreepeterborough.ca

Please keep us informed about any Internet Radio projects that you set up as a result of this document. We want to know that all this hard work is finding some practical application in The Real World. My esteemed thanks to everyone for their continued support of the Radio Free Peterborough Project and other Radio Free Network activities, and for believing that little projects really can make a big difference.

Visit us online at <http://radiofreepeterborough.ca> and use our contact form to drop us a line and let us know what you're up to.