The Linux MP3−HOWTO

Table of Contents

Table of Contents

The Linux MP3−HOWTO

By Phil Kerr, `phil@plus24.com`

v1.40, April 2000

This document describes the hardware, software and procedures needed to encode, play, mix and stream MP3 sound files under Linux.

1.Introduction.

This document describes the hardware, software and procedures needed to encode, play, mix and stream MP3 sound files under Linux.

It covers:

Encoding MP3's from a live or external source.

Encoding MP3's from audio CD's.

Streaming MP3's over a network.

Listening to MP3's.

Mixing MP3's

2.Copyright of this document.

3.Where to get this document.

The most recent official version of this document can be obtained from the Linux Documentation Project http://www.linuxdoc.org/.

The homepage for this HOWTO is: http://www.mp3−howto.com

3.1 Translations

This HOWTO has been translated into the following languages:

Please note that translations may be slightly out of date from this document as, naturally enough, the translations take time.

Korean

http://kldp.org/HOWTO/MP3−HOWTO By Lee,So−min <animator@nownuri.net>

French

http://www.freenix.org/unix/linux/HOWTO/MP3−HOWTO.html By Arnaud Gomes−do−Vale <arnaud@carrosse.frmug.org>

Hungarian

http://free.netlap.hu/howto/MP3−HOGYAN.html By Andras Timar <atimar@itp.hu>

Italian

ftp://ftp.pluto.linux.it/pub/pluto/ildp/HOWTO/MP3−HOWTO By Mariani Dario <darkpand@uni.net>

Spanish

http://www.insflug.org/documentos/MP3−Como By Arielo <larocka@yahoo.com>

Dutch

http://nl.linux.org/doc/HOWTO/MP3−HOWTO−NL.html By Reggy Ekkebus <reggy@zeelandnet.nl>

Japanese

http://www.linux.or.jp/JF/JFdocs/MP3−HOWTO.html By Saito Kan <can−s@geocities.co.jp>

Many thanks to the above translators. If you can translate this HOWTO, please drop the author an email. Also please state the URL where the translation will be housed.

4.Acknowledgments.

In writing this HOWTO I have had to draw heavily on the *Sound−HOWTO* By Jeff Tranter, and the *Sound−Playing−HOWTO* By Yoo C. Chung.

Thanks also to the other HOWTO authors whose works I have referenced:

Linux System Administrators Guide By Lars Wirzenius.

Linux Network Administrators Guide By Olaf Kirch.

Multi Disk System Tuning HOWTO By Stein Gjoen.

Also a big thank−you to all who have sent in feedback, comments and bug−reports.

Special thanks to all my colleagues at WebSentric AG, especially Mark S. Fischer & Peter Conrad for their comments, feedback and support.

5.Disclaimer.

Use the information in this document at your own risk.

I disavow any potential liability for the contents of this document.

Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

All copyrights are owned by their owners, unless specifically noted otherwise.

Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

6.Hardware Requirements & Performance Issues.

Digital Audio processing is a resource intensive task that relies heavily on the processing and I/O capabilities of a system. I would strongly recommend a Pentium class machine as a minimum.

If you are going to be encoding from an analogue audio source via the line or microphone input, a PCI soundcard will give the best results. The I/O performance difference between an ISA and PCI based card is significant, over 132 MBytes/sec for PCI (quote taken from the *PCI−HOWTO*). Naturally, the better the quality of the soundcard in terms of its signal−to−noise ratio, the better the encoded MP3. I've been using the Soundblaster PCI128 and just switched over to a Soundblaster Live Value; both cards give good audio performance, but the Live has significantly better S/N ratios, good enough for semi−pro audio work. Remember the old data processing maxim:− garbage in − garbage out!

Creative have a Beta driver for the Soundblaster Live! which can be downloaded from:

http://developer.soundblaster.com/linux/

When recording analogue audio to a hard disk, more commonly referred to as direct to disk or d2d recording, the performance of the disk, and its interface is critical. If you are using an IDE based based system, mode 4 or UDMA is preferable as the transfer rate is sufficiently high enough to provide reliable data transfer without problems.

The ideal solution would be to use a SCSI based system as the drives and interface have far better throughput capabilities, a sustained 5mbits/sec for SCSI 1 through to 80mbits/sec for ultra/wide SCSI. IDE can peak at anything from 8.3 MB/s to 33 MB/s for Ultra−ATA but these speeds are peak, average transfer rates will be slower. If you can find, or afford, an AV SCSI drive, go for it. AV drives have had the read/write head system optimised for continuous data transference; other SCSI and IDE drives normally cannot sustain continuous data transfer as the write head heats up!

Naturally a drive that has cache will give more consistent results than one that doesn't, as the cache will act as a buffer if the heads do lift or it cannot handle the throughput.

If your drive isn't up to spec, your recording will suffer from dropouts and glitches, where the drive failed to record the signal. If you are recording one−off events, such as live performances invest in a good SCSI based disk system.

Another cause of d2d dropouts is a heavily loaded system. Background tasks can cause the system to momentarily glitch. Its recommended to run as few background services as you can, especially networked based services. For more information about setting network services, and startup scripts please refer to the *SAG* and *NAG* guides.

Virtual memory paging will also cause glitches, so run with as much physical RAM as you can, I'd recommend at least 32 Mb, but you may well need more.

For those wanting to extract the most out of their system, optimising the kernel probably wouldn't do any harm either.

While the hardware specifications above will give you a decent system to encode audio data, don't discount using older, lower spec kit if that's all you have access to.

It'll be a good challenge for a sys−admin to tweak a low−spec system to give good results, and the end result will probably be a happier Linux box.

Another important issue is the audio cabling. Cheap, poor quality cables and connectors will result in poor recording quality. If your soundcard has the option to use phono, sometimes referred to as RCA connectors, use them. Gold plated contacts will also help maintain audio quality, as will keeping audio cables away from data cables as there will be a chance of interference between them.

But don't forget, spending a fortune on the best audio cabling will be lost if the rest of the system hasn't been optimised.

For encoding MP3's from CD−ROM, the speed or type of drive will determine the time taken to read the raw information from it. A single speed drive will probably be too slow for all but the most patient.

Your CD−ROM must be connected to your soundcard if you want to hear what you are recording, either using the internal connector or by connecting headphone's to the headphone output, although you will not be able to listen to MP3's through the CD−ROM headphone socket!

For detailed instructions on setting up soundcards, now would be an excellent time to read the

Sound−HOWTO.

7.Software Requirements.

Converting audio to MP3's is normally a 2 stage process, first the audio is recorded into a WAV format, then the WAV is then converted into an MP3. Some utilities will do both processes in one go for you.

The format you wish to encode audio from, CD or direct audio, will determine what software tools you need to produce the WAV file.

If you are wanting to encode from audio input, you will need a program that will record from your soundcard's input and save the results in a WAV format. Below are some useful utilities (most of the comments are taken from the respective website of the app.)

7.1 Rippers & WAV Recorders

To grab from analog audio line−in. *Wavrec*

Wavrec is distributed as part of wavplay, which can be downloaded from:−

ftp://sunsite.unc.edu/pub/Linux/apps/sound/players/

To convert CD audio data to WAV format, sometimes known as CD ripping:

CDDA2WAV

http://metalab.unc.edu/pub/Linux/apps/sound/cdrom/

Cdparanoia

Cdparanoia is a Compact Disc Digital Audio (CDDA) extraction tool, commonly known on the net as a 'ripper'. The application is built on top of the Paranoia library, which is doing the real work (the Paranoia source is included in the cdparanoia source distribution). Like the original cdda2wav, cdparanoia package reads audio from the CDROM directly as data, with no analog step between, and writes the data to a file or pipe in WAV, AIFC or raw 16 bit linear PCM. Compared to cdda2wav, it's much slower but really gets the best results you can get even from CDs that are difficult to rip for scratches or other read−errors.

http://www.xiph.org/paranoia/index.html

RipEnc

RipEnc is a bourne shell script frontend to Cdparanoia, cdda2wav, tosha and Bladeenc, 8hz−mp3, l3enc. It utilizes CDDB lookups to automate the naming of songs as they are ripped. A manual naming option is also available. The entire CD can be ripped or you can pick the songs to rip. ID3 tags are also supported.

http://www.asde.com/~mjparme/index.htm

RipperX

RipperX is a GTK program to rip CD audio and encode mp3s. It has plugins for cdparanoia, BladeEnc, Lame Mp3 encoder, XingMp3enc, 8hz−mp3, lame, and the ISO v2 encoder. It also has support for CDDB and ID3 tags.

http://www.digitallabyrinth.com/linux/ripperX/

Grip

Grip is a GTK−based CD−player and CD−ripper/MP3−encoder. It has the ripping capabilities of cdparanoia built in, but can also use external rippers (such as cdda2wav). It also provides an automated frontend for MP3 encoders, letting you take a disc and transform it easily straight into MP3s. The CDDB protocol is supported for retrieving track information from disc database servers. Grip works with DigitalDJ to provide a unified "computerized" version of your music collection.

http://www.nostatic.org/grip/

7.2 Encoders

To convert the WAV file to MP3 format you will need an encoder:

Blade's MP3 Encoder

BladeEnc is a freeware MP3 encoder. It is based on the same ISO compression routines as mpegEnc, so you can expect roughly the same, or better, quality . The main difference is the appearance and speed. BladeEnc doesn't have a nice, user−friendly interface like mpegEnc, but it is more than three times faster, and it works with several popular front−end graphical user interfaces.

http://bladeenc.cjb.net

Lame

In the great history of GNU naming, LAME stands for LAME Ain't an Mp3 Encoder. LAME is not an mp3 encoder. It is a GPL'd patch against the dist10 ISO demonstration source. LAME is totally incapable of producing an mp3 stream. It is incapable of even being compiled by itself. You need the ISO source for this software to work. The ISO demonstration source is also freely available, but any commercial use (including distributing free encoders) may require a license agreement from FhG (Fraunhofer Gesellschaft, Germany).

http://www.sulaco.org/mp3/

Gogo

This is a very fast MP3 encoder for x86−CPU, which is based on LAME ver 3.29beta and optimized by PEN@MarineCat, Keiichi SAKAI, URURI, kei and shigeo. (You will also need to download NASM to compile the source, which can be found http://www.web−sites.co.uk/nasm/)

http://homepage1.nifty.com/herumi/gogo_e.html

7.3 Players

To play the MP3's you will naturally need a player:

Xmms (Formerly known as X11Amp)

This player has most of the features as Winamp from Windows 95/98/NT but it will of course feature some specials only available for the linux version.

http://www.xmms.org

Xaudio

Xaudio is a very fast and very robust multiplatform solution for Digital Audio playback, especially targeted at MPEG Audio (MP1, MP2 and MP3) decoding.

http://www.xaudio.com

AlsaPlayer

AlsaPlayer is a new type of PCM player. It is heavily multi−threaded and tries to exercise the ALSA library and driver quite a bit. It has some very interesting features unique to Linux/Unix players. The goal is to create a fully pluggable framework for playback of all sorts of media with the focus on PCM audio data. *Full speed (pitch) control, positive *and* negative! First Linux− and only GPL player that does this!! MP3's and CD's do varispeed :)*

http://www.alsa−project.org/~andy/

mpg123

What is mpg123? It is a fast, free and portable MPEG audio player for Unix. It supports MPEG 1.0/2.0 layers 1, 2 and 3 (those famous "mp3" files), and it has been tested on a wide variety of platforms, including Linux, FreeBSD, NetBSD, SunOS, Solaris, IRIX, HP−UX and others. For full CD quality playback (44 kHz, 16 bit, stereo) a Pentium (or fast 486), SPARCstation10, DEC Alpha or similar CPU is required. Mono and/or reduced quality playback (22 kHz or 11 kHz) is even possible on slower 486 CPUs.

http://dorifer.heim3.tu−clausthal.de/~olli/mpg123/

Freeamp

FreeAmp is an extensible, cross−platform audio player. It features an optimized version of the GPLed Xing MPEG decoder which makes it one of the fastest and best sounding players available. FreeAmp provides a number of the most common features users have come to expect in a clean, easy to use interface.

http://www.freeamp.org/

7.4 Streaming Servers

Streaming servers allow you to 'broadcast' MP3's across a network, whether this is your intranet or the internet itself.

Icecast

Welcome! icecast is a Mpeg Layer III Audio broadcasting system brought to you by the linuxpower.org team. Icecast comes bundled with iceplay, and icedir. iceplay is a playlist streamer that will allow you to send pre−encoded files to your icecast server.

http://www.icecast.org/

Fluid

Fluid Streaming Server is a program for streaming media over networks and in its current form using the mp3 format.

http://www.subside.com/fluid/ (old site) http://fluid.sourceforge.net/ (new site)

7.5 Mixing

LiveIce

LiveIce is the source client for Icecast which encodes an mpeg stream for broadcast as it is created. Unlike clients such as Shout and IceDJ this permits the broadcast of live audio, rather than prerecorded mp3's.

LiveIce is bundled with Icecast, newer versions together with documentation may be found at the website below:

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

eMixer

eMixer is an easy−to−use front−end to mpg123 that allows you to play and mix two mp3 streams together. The ability to mix two mp3s makes eMixer act like a cross−fader, effectively giving the user DJ−like capabilities from the computer console. eMixer is also very able in a "real time" party environment. eMixer is based on the original mp3 mixing code upon which liveice's mixing component is built.

http://emixer.linuxave.net/

7.6 Misc

Volume normalization

Wavnorm

If you have encoded live audio, or have encoded from older cd's you may find variations in the overall sound level.

To change the encoded volume levels of the MP3's you will need to normalise them using wavnorm.

http://www.zog.net.au/computers/wavnorm/

Sox is a very handy sound conversion utility which I'd recommend having, and you will need it if you wish to use wavnorm.

ftp://sunsite.unc.edu/pub/Linux/apps/sound/convert/

You may also need a mixer program; Xmixer works well and is included with most distributions.

8.Setting up your system.

This section will describe the basics of setting up your Linux system to record audio from either an analogue or CD−ROM source.

I'm basing this section around my Intel based Linux system which is running Redhat, but should be reasonably distribution neutral. I'll be working on the Sparc platform version shortly. (if you have any success in using this HOWTO on other hardware, please get in touch).

Naturally a reasonable prerequisite is a working soundcard. At this point in the HOWTO, I invite you to read the excellent *Linux Sound HOWTO*, by Jeff Tranter. After which a good read of the *Linux Sound Playing HOWTO*, by Yoo C. Chung. Both of the above mentioned HOWTO's cover the details of getting a sound system working under Linux far better than I could.

8.1 Setting up for Analogue Audio Capture

Firstly, set up your audio. There are a multitude of ways to route audio before it gets to your Linux box, some common ones are:

Line out to Soundcard Line in. Most audio devices have a Line output sockets. Line level is a standard that specifies what voltage the audio device will send out. If I remember correctly it is 500mV for domestic and Semi Pro devices, and 750mV for Pro audio devices. I would guess that the standard set for most soundcards will be 500mV, but some of the newer Pro audio may be to the higher standard It shouldn't make too much difference unless you are recording at very high levels.

The Line level output is normally used to connect HI−FI equipment to an amplifier, so things such as Tape Decks, Radio Tuners, CD players, DAT machines and Mini−Disc players should connect without problem. Turntables can be more of a problem, see below for more information.

You could capture audio from VCR's as well. Most VCR's will either have Line out for sound, or you can Get a Line out from a SCART socket if your VCR has one.

Amplifier Tape out to Soundcard Line in, Soundcard Line out to Amplifier Tape in. This configuration is essentially replacing a traditional tape recorder connected to your HI−FI amplifier with your Linux system. The Soundcard Line out to Tape in allows monitoring of the recording levels.

Mike to Soundcard Mike in. The voltages generated by microphones is very much smaller than those used in Line level devices. If you were to plug a Microphone into the Soundcard Line in, chances are you would never record anything.

WARNING, doing the reverse, plugging a Line level device into the Soundcards Microphone input, can damage your soundcard!!

Turntable to Mike in.

Many thanks to Mark Tranchant for the following.

*The raw output from a record deck cartridge is very low level. However, you cannot plug it directly into a microphone input and expect good results. The output requires equalization, as records are mastered with less bass and more treble to optimize the physics of the moving needle. This equalization is carefully defined and referred to as RIAA equalization. You *need* to run the output through a phono preamp first, and then into a line input.*

Music keyboards & synths should be connected to the Soundcards Line in, with guitars connecting to Line in via a DI (Direct Injection, used to convert the signal to Line level) box.

Before you plug in anything into your soundcard, make sure the volume levels are turned down to minimum, or if using microphones they are either turned off or away from speakers.

8.2 Setting up for CD−ROM Audio Capture

Setting up your Linux system to extract audio data from CD−ROM is reasonably straight forward.

If you can hear a track playing from your CD−ROM through your speakers or amplifier, connected to your soundcard, then there's a reasonable chance you should be able to record from it.

8.3 Additional Setting up

Log in as per normal to your system, then using a mixer program set the recording levels that are loud enough to give you a decent recording level, but aren't too loud and distorting. I normally just judge this by ear, after a while you'll get to know what levels are best for your kit.

I recommend either turning off all unnecessary services or switching to the single user runlevel, especially when encoding from an audio source. This is to ensure that the bare minimum of services are running and thus minimising system glitches when recording.

I've set up a separate SCSI drive, exclusively to record the audio to, which I'll refer to as /mp3. I've done this mainly for the performance gains in using a SCSI drive. Also, recording onto a dedicated drive, where you are almost certain the head isn't going to suddenly skip to another part of the drive as you are writing audio data to it, is a good thing :)

For details on setting up a Linux system with multiple disk drives, a good read of the *Multi−Disk−HOWTO*, by Stein Gjoen may be useful.

9.Encoding from Audio.

Firstly, make sure you have enough space on your drive. At CD quality, 44.1 Khz 16 Bit stereo, 1 minute takes nearly 10 Mb (5 MB per channel).

I normally record at DAT quality, which is 48 Khz 16 Bit stereo.

Using wavrec I use the following syntax:

/usr/local/bin/wavrec −t 60 −s 48000 −S /mp3/temp.wav

The first part is an explicit path to wavrec. The '−t 60' specifies the length of time to record for, in seconds.

The third option, −s 48000 refers to the sample rate in samples/sec. (48000 is the rate for DAT, 44100 is CD)

The last option is the path to the output file.

To see the full set of options, run waverec −help, or see it's man page.

This will produce your WAV file Next you will need to encode it into MP3 format.

Use bladdenc with the following command line.

/usr/local/bin/bladeenc *[source file] [destination file]* −br 256000

The −br option sets the bit rate, in this case I've set the rate to the maximum rate of 256k bits/s. The path to bladeenc may also be different on your system to the one I've used in my example.

To see the full set of options, run bladeenc −help, actually this is an invalid option, but will display the list of options.

The same encoding using Lame (as well as Gogo as it is based on Lame) would need the command line

/usr/local/bin/lame *[source file] [destination file]* −b 256

10.Encoding from CD−ROM.

In a similar way to encoding from audio, encoding from CD is a 2 stage process. Firstly the audio data is extracted from the cd and converted into a wav file. Then the wav file is converted into MP3.

There are basically 2 types of encoders, console based and X based. Both do the same job, but the X based are easier to use (and look nicer).

Again, before you start to encode, check you will have enough drive space on your system.

10.1 Command Line encoding

I've written a very simple Perl script that will rip and encode tracks from a CD.

```
#!/usr/bin/perl

if ($ARGV[0] ne "") {

$count = 1;

do {

$cdcap = system("cdparanoia", $count, "/mp3/cdda.wav");
$track = "$ARGV[1]/track".$count.".mp3";
$enc = system("bladeenc  /mp3/cdda.wav $track −br 256000");
$count++;

}
until $count > $ARGV[0];
exit;
}

else {
print "Usage cdriper [no of tracks] [destination directory]\n\n";
}
```

Please note: The above script is very basic and has nothing fancy, like error checking or CDDB. Improve at your leisure :)

The main lines of interest are:

```
$cdcap = system("cdparanoia", $count, "/mp3/cdda.wav");
```

This line calls the CD ripper, cdparanoia. Cdparanoia converts raw CD audio data to WAV format.

I'm using Cdparanoia, but if you wish to use CDDA2WAV, the command line would be:

```
$cdcap = system("cdda2wav", $count, "/mp3/cdda.wav");
```

The salient options are $count, which is the number of tracks to rip, and then the path for the outputted WAV file. In my example this will go to a tmp directory on my MP3 SCSI drive.

The WAV file is then converted into a MP3 file using Bladeenc.

I've written this Perl script in order to rip a CD without having to rip and encode each track, and without having to use the batch mode of Cdparanoia. This cuts down on free disk space needed as Cdparanoia's batch mode will rip the whole disk, and take up anything upto 600 Meg.

If you wanted to use Lame or Gogo, replace the encoder line with:

```
$enc = system("lame  /mp3/cdda.wav $track −b 256");
```

or

```
$enc = system("gogo  /mp3/cdda.wav $track −b 256");
```

Here is a dump of the available option for each of the encoders.

Bladeenc

```
BladeEnc 0.91    (c) Tord Jansson          Homepage: http://bladeenc.mp3.no
===============================================================================
BladeEnc is free software, distributed under the Lesser General Public License.
See the file COPYING, BladeEnc's homepage or www.fsf.org for more details.

Usage: bladeenc [global switches] input1 [output1 [switches]] input2 ...

General switches:
  −[kbit], −br [kbit]  Set MP3 bitrate. Default is 128 (64 for mono output).
  −crc                 Include checksum data in MP3 file.
  −delete, −del        Delete sample after successful encoding.
  −private, −p         Set the private−flag in the output file.
  −copyright, −c       Set the copyright−flag in the output file.
  −copy                Clears the original−flag in the output file.
  −mono, −dm           Produce mono MP3 files by combining stereo channels.
  −leftmono, −lm       Produce mono MP3 files from left stereo channel only.
  −rightmono, −rm      Produce mono MP3 files from right stereo channel only.
  −swap                Swap left and right stereo channels.
  −rawfreq=[freq]      Specify frequency for RAW samples. Default is 44100.
  −rawbits=[bits]      Specify bits per channel for RAW samples. Default is 16.
  −rawmono             Specifies that RAW samples are in mono, not stereo.
  −rawstereo           Specifies that RAW samples are in stereo (default).
```

```
  −rawsigned           Specifies that RAW samples are signed (default).
  −rawunsigned         Specifies that RAW samples are unsigned.
  −rawbyteorder=[order]Specifies byteorder for RAW samples, LITTLE or BIG.
  −rawchannels=[1/2]   Specifies number of channels for RAW samples. Does
                       the same as −rawmono and −rawstereo respectively.

Global only switches:
  −quit, −q            Quit without waiting for keypress when finished.
  −outdir=[dir]        Save MP3 files in specified directory.
  −quiet               Disable screen output.
  −nocfg               Don't take settings from the config−file.
  −prio=[prio]         Sets the task priority for BladeEnc. Valid settings are
                       HIGHEST, HIGHER, NORMAL, LOWER, LOWEST(default) and IDLE
  −refresh=[rate]      Refresh rate for progress indicator. 1=fastest, 2=def.
  −progress=[0−8]      Which progress indicator to use. 0=Off, 1=Default.

Input/output files can be replaced with STDIN and STDOUT respectively.
```

Lame

```
LAME version 3.50 (www.sulaco.org/mp3)
GPSYCHO: GPL psycho−acoustic model version 0.74.

USAGE   :  lame [options] <infile> [outfile]

<infile> and/or <outfile> can be "−", which means stdin/stdout.

OPTIONS :
    −m mode         (s)tereo, (j)oint, (f)orce or (m)ono  (default j)
                    force = force ms_stereo on all frames. Faster and
                    uses special Mid & Side masking thresholds
    −b <bitrate>    set the bitrate, default 128kbps
                    (for VBR, this sets the allowed minimum bitrate)
    −s sfreq        sampling frequency of input file(kHz) − default 44.1
  −−resample sfreq  sampling frequency of output file(kHz)− default=input sfreq
  −−mp3input        input file is a MP3 file
  −−voice           experimental voice mode

    −v              use variable bitrate (VBR)
    −V n            quality setting for VBR.  default n=4
                    0=high quality,bigger files. 9=smaller files
    −t              disable Xing VBR informational tag
    −−nohist        disable VBR histogram display

    −h              use (maybe) quality improvements
    −f              fast mode (low quality)
    −k              disable sfb=21 cutoff
    −d              allow channels to have different blocktypes
  −−athonly         only use the ATH for masking

    −r              input is raw pcm
    −x              force byte−swapping of input
    −a              downmix from stereo to mono file for mono encoding
    −e emp          de−emphasis n/5/c  (obsolete)
    −p              error protection.  adds 16bit checksum to every frame
                    (the checksum is computed correctly)
    −c              mark as copyright
    −o              mark as non−original
    −S              don't print progress report, VBR histograms

  Specifying any of the following options will add an ID3 tag
```

```
     −−tt <title>     title of song (max 30 chars)
     −−ta <artist>    artist who did the song (max 30 chars)
     −−tl <album>     album where it came from (max 30 chars)
     −−ty <year>      year in which the song/album was made (max 4 chars)
     −−tc <comment>   additional info (max 30 chars)


MPEG1 samplerates(kHz): 32 44.1 48
bitrates(kbs): 32 48 56 64 80 96 112 128 160 192 224 256 320

MPEG2 samplerates(kHz): 16 22.05 24
bitrates(kbs): 8 16 24 32 40 48 56 64 80 96 112 128 144 160
```

Gogo

```
GOGO−no−coda ver. 2.24 (Feb 12 2000)
Copyright (C) 1999 PEN@MarineCat and shigeo
          Special thanks to Keiichi SAKAI, URURI, Noisyu and Kei
This is based on LAME3.29beta and distributed under the LGPL
usage
gogo inputPCM [outputPCM] [options]

 inputPCM is input  wav file
if input.wav is `stdin' then stdin−mode
outputPCM is output mp3 file (omissible)

options
−b  kbps     bitrate [kpbs]
−br bps      bitrate [ bps]
−silent      dont' print progress report
−off         {3dn,mmx,kni(sse),e3dn}
−v {0,..,9}  VBR [0:high quality 9:high compression]
             You should combine this with −b option
for only RAW−PCM input
−offset bytes skip header size
  −8bit       8bit−PCM [dflt 16bit−PCM]
  −mono       mono−PCM [dflt stereo−PCM]
  −bswap      low, high byte swapping for 16bitPCM
  −s kHz      freq of PCM [dflt 44.1kHz]
−nopsy       disable psycho−acoustics
−m  {s,m,j}  output format s:stereo, m:mono, j:j−stereo
−d  kHz      change sampling−rate of output MP3
−emh {n,c,5} de−emphasis
−lpf {on,off} 16kHz filter [dflt use if <= 128kbps; not use if >= 160kbps]
−test        benchmark mode
−delete      delete input file, after encoding
```

RipEnc

RipEnc performs the same task as the code above, but is written in shell and is easier to use :)

Here's what it looks like.

```
RipEnc version 0.7, Copyright (C) 1999  Michael J. Parmeley
<mjparme@asde.com>, RipEnc comes with ABSOLUTELY NO WARRANTY
```

```
There is currently NO encoding process running in the background
Your encode.log file is 982607 bytes long.

<Enter 'd' for details, 'v' to view the encode log, or 'del' to delete the encode log>


1) Change working directory....................[/megajukebox/tmp]
2) Choose encoder..............................[lame]
3) Choose ripper...............................[cdparanoia]
4) Choose id3 tool.............................[none]
5) Toggle between Manual and CDDB naming.......[manual]
6) Setup XMCD_LIBDIR variable for CDA..........[/var/X11R6/lib/xmcd]
7) Set preferred naming convention.............[artist−name_of_song.mp3]
8) Rip whole CD?...............................[no]
9) Set small hard drive option?................[no]
10) Please select your Cd−Rom device...........[/dev/cdrom]
11) Set the Bitrate for the encoded MP3's......[256]
12) List the files in your working directory
13) Start
14) About
15) Exit
?
```

10.2 GUI Based Encoders

GUI based encoders offer all the functionality of console based encoding, but wrap it all up in a nice easy to use interface. Grip and RipperX are similar in operation, both offer you the ability to select one, several or all tracks on a CD and convert them. They also offer CDDB support which allows you to retrieve the album and track information from a server and saves you having to enter the information by hand.

10.3 Encoder Performance

In the encoding sections I've mentioned 3 different encoders, bladeenc, lame and gogo. The main difference is their performance in encoding (although there are differences in the available options which were listed earlier).

A little example. I ripped a track from a CD and then encoded it with the different encoders. All encoders were run with the same system conditions and all produced stereo out mp3's.

```
[dj@megajukebox]$ ls −l cdda.wav
−rw−rw−r−−   1 dj       dj       59823164 Feb 10 00:56 cdda.wav

[dj@megajukebox]$ bladeenc cdda.wav −br 256

BladeEnc 0.91    (c) Tord Jansson          Homepage: http://bladeenc.mp3.no
===============================================================================
BladeEnc is free software, distributed under the Lesser General Public License.
See the file COPYING, BladeEnc's homepage or www.fsf.org for more details.

Files to encode: 1
```

```
Encoding:  ../test.wav
Input:     44.1 kHz, 16 bit, stereo.
Output:    128 kBit, stereo.

Completed. Encoding time: 00:05:58 (0.78X)

All operations completed. Total encoding time: 00:05:58

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

[dj@megajukebox]$ lame cdda.wav −b 256
LAME version 3.50 (www.sulaco.org/mp3)
GPSYCHO: GPL psycho−acoustic model version 0.74.
Encoding ../test.wav to ../test.wav.mp3
Encoding as 44.1 kHz 128 kbps j−stereo MPEG1 LayerIII file
    Frame          |  CPU/estimated  |  time/estimated | play/CPU |   ETA
 10756/ 10756(100%)| 0:02:28/ 0:02:28| 0:02:29/ 0:02:29|    1.9074| 0:00:00

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

[dj@megajukebox]$ gogo cdda.wav −m s −b 256
GOGO−no−coda ver. 2.24 (Feb 12 2000)
Copyright (C) 1999 PEN@MarineCat and shigeo
          Special thanks to Keiichi SAKAI, URURI, Noisyu and Kei
MPEG 1, layer 3 stereo
inp sampling−freq=44.1kHz out sampling−freq=44.1kHz bitrate=256kbps
inp sampling−freq=44.1kHz out sampling−freq=44.1kHz bitrate=128kbps
input  file `../test.wav'
output file `../test.mp3'
{  10751/  10755} 100.0% (  2.94x)  re:[00:00:00.03] to:[00:01:35.42]
End of encoding
time=  95.430sec
```

It would appear that Gogo has a much optimised algorithm for encoding than Bladeenc and Lame.

11.Streaming MP3's

A streaming server allows you to transmit MP3 files over a TCP based network. This can be the Internet itself or your local network / intranet.

The connection principal is very similar to that of a web server, files are streamed when a client (the MP3 player) connects to the server.

Setting−up a streaming server is reasonably straight forward, I'll focus on Icecast first, then Fluid.

11.1 Icecast

After downloading and untaring, a good look around the doc/ directory would be a good thing, the HTML manual is very helpful and comprehensive.

If you have downloaded the source code, follow the instructions regarding compiling pertinent to your system.

Icecast will not work correctly unless you correctly set the servername in the config file, icecast.conf, which is located in the etc directory. It must match the name that resolves to your IP address exactly.

If you see the following line when Icecast starts−up you have problems:

```
−> [05/Jan/2000:17:21:04] WARNING: Resolving the server name [your.server.name] does not work!
```

Edit icecast.conf which is located in the etc directory and locate the line containing the entry for "server_name" and enter your servers name. If you are unsure you can find out by using the *hostname* command, or by cat'ing /etc/hosts.

Once you've made the neccesary changes you'll need to either copy the conf file to the bin directory, or start icecast with the −c option and specify the location, like so:

```
./icecast −c ../etc/icecast.conf
```

If everything has been configured correctly, you should see something similar to the following:

```
[dj@megajukebox bin]$ ./icecast −c ../etc/icecast.conf −d /home/dj/mp3/icecast/
Icecast Version 1.3.0 Starting...
Icecast comes with NO WARRANTY, to the extent permitted by law.
You may redistribute copies of Icecast under the terms of the
GNU General Public License.
For more information about these matters, see the file named COPYING.

[05/Jan/2000:18:36:30] Icecast Version 1.3.0 Starting..
[05/Jan/2000:18:36:30] Using stdin as icecast operator console
[05/Jan/2000:18:36:30] Tailing file to icecast operator console
[05/Jan/2000:18:36:30] Server started...
[05/Jan/2000:18:36:30] Listening on port 8000...
[05/Jan/2000:18:36:30] Using [megajukebox] as servername...
[05/Jan/2000:18:36:30] Max values: 1000 clients, 1000 clients per source, 10 sources, 5 admins
−> [05/Jan/2000:18:36:30] [Bandwidth: 0.000000MB/s] [Sources: 0] [Clients: 0] [Admins: 1] [Uptime
```

The −d option sets the directory for log files and templates.

Below is the list of command−line options:

```
        −c [filename]

Parse as a configuration file. Please note that any command line
parameters you supply after this override whatever is in file. Also note that
icecast.conf in the current directory is already parsed when you specify
this file, so anything in icecast.conf not overridden by the new configuration
file will be used by the server.

        −P [port]

This is the port used for all client, source, and admin connections. It's set
to 8000 by default.
```

```
        −m [max clients]

Allow this number of client connections. When this number is reached, all
client connections will be refused with 'HTTP/1.0 504 Server Full'

        −p [encoder password]

This sets the password that the encoder must use to be allowed to stream
to the server. Note that if you have compiled the server with crypt()
support, this argument must be an encrypted string.

        −b

This will send the icecast server into the background (i.e daemon process).
To use the admin commands now, you have to connect to the server as an
admin, using some sort of telnet client.

        −d [directory]

Make all log files created by icecast, and all templates that icecast looks
for be relative to this directory.
```

So, thats the server started, but you now need to connect an MP3 source to the server.

You can choose from two applications which deliver MP3 data to the server, Shout and LiveIce.

Shout

Shout provides Icecast with a static playlist of MP3's to stream and is included with Icecast.

You create the playlist if the MP3 files you want to stream with the following:

```
find [MP3 directory] −name *.mp3 −print > playlist
```

At it's most basic level, to start the shout service, issue the following:

```
[dj@megajukebox bin]# ./shout megajukebox −P hackme −p playlist
```

The −P option specifies the password needed to add a mount−point to Icecast, this is the aptly set as *hackme*..... I strongly suggest you change it otherwise someone may :) The −p option specifies the location of the playlist file. Below is a list of all of the command line options:

```
[dj@megajukebox bin]# ./shout
Usage: shout <host> [options] [[−b <bitrate] file.mp3]...
Options:
        −B <directory>       − Use directory for all shout's files.
        −C <file>    − Use file as configuration file
        −D <dj_file> − Run this before every song (system())
        −P <password>        − Use specified password
        −S              − Display all settings and exit
        −V              − Use verbose output
        −X <desc>    − Use specified description.
```

```
-a              - Turn on automatic bitrate (transfer) correction
-b <bitrate> - Start using specified bitrate
-d              - Activate the dj.
-e <port>    - Connect to port on server.
-f              - Skip files that don't match the specified bitrate
-g <genre>   - Use specified genre
-h              - Show this text
-i              - Use old icy headers
-k              - Don't truncate the internal playlist (continue)
-l              - Go on forever (loop)
-m <mount>   - Use specified mount point
-n <name>    - Use specified name
-o              - Turn of the bitrate autodetection.
-p <playlist>        - Use specified file as a playlist
-r              - Shuffle playlist (random play)
-s              - (Secret) Don't send meta data to the directory server
-u <url>     - Use specified url
-v              - Show version
-x              - Don't update the cue file (saves cpu)
-z              - Go into the background (Daemon mode)
-t              - Enable title streaming
```

LiveIce

LiveIce can work in 2 modes, it can pass a playlist to Icecast or can pass live audio from the soundcard.

After untaring and reading the README concerning building the package, make sure you have mpg123 installed and available as LiveIce requires it.

There are two ways of configuring LiveIce editing the config file with vi/emacs/or whatever or by using the TK based configuration tool, which is a pretty way of editing it :)

The best place for describing the internals of liveice.cfg can be found at LiveIce's homepage where Scott covers all of the options.

This is a copy of my config file with LiveIce set to mixer mode (stream from a list of MP3's)

NOTE: I've added comments to the file, so if you cut and paste make sure the comments haven't wrapped around to a new line otherwise LiveIce will not work :)

```
# liveice configuration file
# Automatically generated

SERVER megajukebox              # Your server name * MUST BE THE NAME THE SERVER RESOLVES TO *
PORT 8000                       # The port Icecast is running on

NAME Megajukebox                # Information regarding the name of your server which is sent to
                                # to directory servers.
                                # Examples 'Sarah FM' or 'ThisTown: Loud and Heavy Jazz - Interne

GENRE Live                      # Information regarding the genre.  Examples 'Talk' or 'Dance'

DESCRIPTION                     # Information regarding the station.  Example 'The best for regga
```

```
URL http://megajukebox:8000     # The URL and port of the server.

PUBLIC 0                        # Set this to 1 if you want Icecast to announce your station and
                                # on a directory server, otherwise leave 0

XAUDIOCAST_LOGIN                # can be either ICY_LOGIN or X_AUDIOCAST_LOGIN.  X_AUDIOCAST is b

MOUNTPOINT /techno              # Sets the mountpoint name of the stream for Icecast.  Only used
                                # otherwise defaults to icy_0

PASSWORD hackme                 # Icecast's admin password

SAMPLE_RATE 44100               # The sample rate of the stream
STEREO                          # Can be MONO or STEREO

NO_SOUNDCARD                    # See below

HALF_DUPLEX                     # Sets the soundcard duplex mode.  Can be HALF_DUPLEX or FULL_DUP
USE_GOGO                        # Sets the encoder to use.  Check the README for the list
BITRATE 128000                  # Sets the bit rate of the stream (see below)
VBR_QUALITY 1                   # Sets the variable bit rate quality.

MIXER                           # See below

PLAYLIST /megajukebox/playlist  # Location of the playlist (see details on the find command later

TRACK_LOGFILE track.log         # Filename and location to dump list of MP3's streamed
```

Once you have your config file you start LiveIce like so:

```
[dj@megajukebox liveice]$ ./liveice
/megajukebox/playlist
1
opening connection to megajukebox 8000
Attempting to Contact Server
connection successful: forking process
opening pipe!...
writing password
Setting up Interface
Soundcard Reopened For Encoding
Input Format: 16Bit 44100Hz Stereo
Output Format: 256000 Bps Mpeg Audio
IceCast Server: megajukebox:8000
Mountpoint: /techno
Name: megajukebox − this and that radio − broadcasting 24/7
Genre: Techno
Url: http://megajukebox
Description: a load of digital noise −> but i know you like it :)

 Press '+' to Finish
adding /megajukebox/demotunes/track_1.mp3
adding /megajukebox/demotunes/track_2.mp3
adding /megajukebox/demotunes/track_3.mp3
adding /megajukebox/demotunes/track_4.mp3
/megajukebox/demotunes/track_4.mp3
Adding New Channel 1
Adding New Channel 2
Channel 1 selecting
 /megajukebox/demotunes/track_1.mp3
Channel 2 selecting
```

```
 /megajukebox/demotunes/track_1.mp3
Playing track_1.mp3
searching for Id3v2
searching for Id3v1
copying the data
fixing the nulls
adding the url
closing input file
Using log track.log
```

The last line is a peak meter.

These are the keyboard controls for mixer mode:

```
Action                          Channel 1 Key   Channel 2 Key
~~~~~~                          ~~~~~~~~~~~~~   ~~~~~~~~~~~~~
Select next track on channel    1               a
Select prev track on channel    q               z
Start/Stop channel              2               s
Reset channel                   w               x
Increase volume on channel      3               d
Decrease volume on channel      e               c
Increase speed on channel       4               f
Decrease speed on channel       r               v
Sticky mode On/Random/Off       5               g
Preview channel                 t               b
Random Track                    u               m
```

The above liveice.cfg is for mixermode. To use LiveIce in audio mode change the line relating to MIXER to NOMIXER and set NO_SOUNDCARD to SOUNDCARD and restart LiveIce.

Forgetting to set the correct options will lead to some interesting warning ;)

```
   946:Error: Line In mode *and* no soundcard??????? Eeejit!
```

Once you have it all correctly set up and have plugged in an external source, you should be able to stream =:)

```
[dj@megajukebox liveice]$ ./liveice
/megajukebox/playlist
0
Initialising Soundcard
16Bit 22050Hz Stereo Full Duplex
opening connection to megajukebox 8000
Attempting to Contact Server
connection successful: forking process
opening pipe!...
writing password
Setting up Interface
Soundcard Reopened For Encoding
Input Format: 16Bit 22050Hz Stereo
Output Format: 32000 Bps Mpeg Audio
IceCast Server: megajukebox:8000
Mountpoint: /daves_band_live_at_the_club
Name: megajukebox − Dave and the Dynamite − Live at the Roxy
Genre: Live/Rock
Url: http://megajukebox
Description: megajukebox::Louder than a frog in a trashcan..... and almost as musical
```

```
 Press '+' to Finish
Lvl: L:   8704 R:  11776
```

The last line is a signal level meter, if the input signal is too high you will get a *clip* warning. If you do turn down the gain of the input source.

The keen eyed amongst you may of noticed that in liveice.cfg the first comment lines point out that the file is automatically generated. If you are using the TK based GUI liveiceconfigure.tk and you've made manual changes, you will lose them when you save. Either use the GUI or learn vi/emacs :)

11.2 Fluid

After untaring the bundle cd to the directory, then read the README :)

Fluid has three basic modes of operation, transmit, relay and forward. I'll only focus on transmit.

The config files associated for transmit are located in config/MP3TX.cfg. To test the server run with the following, at this point the default config settings should be ok:

java Fluid TX

Naturally enough you'll need Java of some form installed first. You can use either the Blackdown port of JDK available from http://www.blackdown.org or if you are using Redhat, Kaffe.

Fluid comes with a few sample MP3 files, so if everything is working you should see something similar to this (I've started the server using Kaffe in this example, you may have to start it using java):

```
[dj@megajukebox Fluid−Beta2J]$ kaffe Fluid tx
−−−−−−− Fluid Streaming Server Beta 2 −−−−−−−
This program is ShareWare(tm) and it will not
be crippled in any way because of it. However
if you do like the program and will use it
commercial purposes, we ask of you to contact
us at the address below for pricing info:

 Eldean AB                   E−mail:
 Sjoangsvagen 7              fluid@subside.com
 S−192 72 Sollentuna
 SWEDEN

   Fluid is Copyright Subside (C) 1998
       written by Lars Samuelsson
         http://www.subside.com
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

* Transmission mode *
Reading config from: config/MP3TX.cfg
Reading playlist: playlist.m3u
Server started on port: 2711
Accepting administrator login on port: 2710
P| Dr. Nick − Hello Everybody
```

If you get this far, it looks like things are working, but I'm sure you'll want to stream more than the demo files!

You'll need to compile a playlist of the MP3's you want to stream. This will be a static list users will not be able to alter this list or make requests. This playlist is named playlist.m3u and is located by default in the root directory.

To compile a playlist of all MP3's in a particular directory (or disk) use the following command:

```
find [MP3 directory] −name *.mp3 −print > playlist.m3u
```

By default the server uses port 2711, which is where your listeners will connect to, if you need to change this this can be done in the config file.

The server can be remotely administered by telneting to it's admin port, by default port 2710 like so:

```
[dj@megajukebox Fluid−Beta2J]$ telnet localhost 2710
Trying 127.0.0.1..megajukebox
Connected to localhost.localdomain.
Escape character is '^]'.
jaguar
You are connected to the −Fluid− Streaming Server
Type "help" for a command reference
help
The following commands are available:
 help conn curr exit
curr
Information about the currently broadcasted song:
Title:   Beer Talk
Artist:  Homer Simpson
Album:   The Simpsons
Year:    1996
Comment: Borrowed this as an example
Genre:   Comedy
```

The reference to "jaguar" is the admin password, this is the default. There is no prompt for the password so please don't sit there waiting for one! I suggest that you change the password from the default otherwise you will invite a hack! This can be changed in the config file, which looks like this:

```
[dj@megajukebox config]$ cat MP3TX.cfg
2711
2710
5
4096
32
1000
jaguar
playlist.m3u
current.txt

# −−− The lines are −−−
# 1. PORT number (the server will use)
# 2. PORT number (for maintaining the server remotely)
# 3. Maximum number of connections (the server will accept)
# 4  Packetsize when reading/sending (in bytes)
# 5. Bitrate of the mp3s in kbit/s (all mp3s must have same bitrate)
```

```
# 6. Delay between songs (in milliseconds)
# 7. Password for remote administration
# 8. Playlist name (list in .m3u format)
# 9. Name of the file to write song info to (from ID3−tag)
```

The reference to the playlist being in m3u format means that it is in the same format as produced by the find command mentioned earlier.

11.3 Bandwith considerations

Streaming audio can consume vast quantities of bandwidth if the MP3 servers' bit−rate is set too high.

Consider this scenario. A T1 link has a capacity of approx. 1.55 Mb/Sec. If you stream your MP3's at 128K/Bps stereo, each connecting player will use 256K/Bps, so only 6 users could connect to your MP3 server at any time without problems. And at 256K/Bps, you will not get too many modem users connecting!

So you must make a decision at what to set your stream rates not only on what your server's internet connection is rated at, but what your users will be connecting at. 24K/Bps Stereo will give a reasonable quality signal that 56K modem users will be able to connect to, and for the same T1 line would allow approx. 32 simultaneous connections.

If your server is running on an Intranet, bandwidth issues will still have to be considered especially if your network is running 10M/Sec.

But please let either your ISP or sys admin know you are going to stream otherwise you may be in for a shock. Some ISP's will charge you for bandwidth over a certain limit and sys admins like to know why their network is now running slow :)

11.4 Copyright Issues

I think it's reasonable to assume that record companies will not like you streaming material without their permission or payment of some kind! So what can you stream?

This is an area where you will need to be aware of the legal ramifications, because it will be you who will be liable.

Below are two links, one for the Electronic Frontier Foundation who are advocates of freeing restrictions surrounding the technology. The other link is to the Recording Industry Association of America, which seeks to protect the rights of artists from piracy.

I strongly suggest visiting both of the sites, and any others relevant to where you are physically based.

http://www.eff.org/cafe/

http://www.riaa.com/weblic/weblic.htm

12.Listening to MP3's.

So, hopefully, you should now have some MP3 files ready to listen to, and have the choice of paying from file or stream.

12.1 Playing from File

Playing from file is reasonably straight−forward with all players. The only big difference is some are command−line based and some are X based.

Playing an MP3 file from file requires you to pass the mp3 file as a parameter, like so:

```
[dj@megajukebox]$ mpg123 /mp3_files/SampleFile.mp3

or

[dj@megajukebox]$ xaudio /mp3_files/SampleFile.mp3
```

If you want to play a series of files, pass them in as a list:

```
[dj@megajukebox]$ alsaplayer /mp3_files/SampleFile1.mp3 /mp3_files/SampleFile2.mp3
```

To play all the tracks in a directory, just wildcard the file selected, like so:

```
[dj@megajukebox]$ xmms /mp3_files/*.mp3
```

12.2 Playing from MP3 Streams

Playing from a MP3 stream is quite easy, just replace the file with the streams url and port number:

```
mpg123 http://localhost:8000

or

freeamp http://megajukebox:2711
```

12.3 Mixing

eMixer

eMixer gives you the ability to mix MP3's in a simllar manner to a DJ's mixing desk.

Newer versions support 2 sound cards so you can output your mix on one card and monitor or cue the next track on another.

As usual once untaring, read the readme on how to build the package.

You will need to create a playlist of MP3 files, do this with the find command mentioned in the Streaming section.

You will need mpg123 installed before you can run eMixer.

Here are the contol keys (taken from the readme)

```
KEYBOARD CONTROLS
"up, down"              scroll thru playlist
"page up, page down"    scroll thru playlist screen full at a time
"enter"                 start/stop track
"tab"                   change channel
"}","]"                 toggle between volume and speed controls/windows
"space"                 restart active track
"left, right"           fader controls
"insert"                decrease volume/speed in channel one
"home"                  increase volume/speed in channel one
"delete"                decrease volume/speed in channel two
"end"                   increase volume/speed in channel two
"< , / , >"             left, centre & right positions of fader
" + , = "       (NEW)   switch between faders
" q "                   start/stop channel channel one
" w "                   start/stop channel channel two
" p "                   toggle between playmodes − single, loop, continous, random
" a "                   stop all channels
" f "                   file menu
" u "                   util menu
" h "                   help menu
" ~, ` "                cancel menu drop down
" s "                   turn SIM Play on
                        (SIM Play starts the same track in both channels
                        simultaneously )
```

13.Feedback.

New hardware and software is being released all the time. If you are using newer versions of the hardware and / or software listed within this document, or can add to anything within this area, please send your information for inclusion to *phil@plus24.com* and I'll include it in the next release.

While I'd like to reply to every question, please note that on occasions I will not be able to reply quickly due to work.

Happy MP3'ing!!