Linux MP3 CD Burning mini−HOWTO

Greg Wierzchowski

greg.wierzchowski@usa.net

2001−05−29

Revision History

Revision 1.5	2001−11−19	Revised by: GW
Fixed omission in Disc−At−Once burning section.		
Revision 1.4	2001−11−17	Revised by: GW
Added Disc−At−Once burning section.		
Revision 1.3	2001−09−02	Revised by: GW
Added another example of decoding MP3 files with lame.		
Revision 1.2	2001−07−12	Revised by: GW
Minor layout changes; Added Translations subsection into Credits.		
Revision 1.1	2001−06−12	Revised by: GW
Minor cleanup; Regexp fix for MP3 to WAV name conversion example.		
Revision 1.0	2001−05−29	Revised by: GW
Initial Release.		

A complete recipe for creating audio CDs from MP3 files.

Table of Contents

1. Introduction....1
1.1. Copyright and License....1

2. Preparing the Tracks....2

3. Burning Your CD....3

4. Burning a DAO CD....4

5. Credits....5
5.1. Translations....5
5.2. Other Credits....5

1. Introduction

This mini−HOWTO was created because of my experience with burning music CDs and lack of some specific information about sound normalization on the Internet. I usually burn music CDs as a mix − different songs from different sources.Very often volume level between songs varies greatly. This is the first obstacle. Second, many of the files on the Internet are not CD−compatible (16 bit, stereo, 44.1 kHz) and have to be converted. There are many programs to burn music CDs from MP3 files, and many of them do the conversion transparently. But I haven't seen a single tool that also normalizes the volume, so that's why I worked out my own CD−burning recipe.

I'm assuming you wish to burn a CD with the collection of songs you obtained from different sources, all varying quality, but you want to get the best−sounding CD possible. This mini−HOWTO outlines the steps that may help you.

1.1. Copyright and License

This document is copyright 2001 by Greg Wierzchowski and is released under the terms of the GNU Free Documentation License, which is hereby incorporated by reference. Send feedback to *greg.wierzchowski@usa.net*.

---

2. Preparing the Tracks

Note

All commands assume bash shell

1. Collect all MP3 files in one directory.
2. If any filenames contain spaces, first convert them to underscores:
```
for i in *.mp3; do mv "$i" `echo $i | tr ' ' '_'`; done
```
3. Convert them to WAV with the command:
```
for i in *.mp3; do mpg123 -w `basename $i .mp3`.wav $i; done
```

Mpg123 should be present in any Linux distribution, but if you don't have it, get it at http://www.mpg123.de/.

NOTE I noticed that with some MP3 files mpg123 output was distorted. At first I thought that MP3's were bad, but then I checked with another player and they sounded OK. So I searched for another MP3 player that could write WAV files to disk, and found this one: *MAD mp3 decoder* at http://www.mars.org/home/rob/proj/mpeg/. With *madplayer*, the command line is:

```
for i in *.mp3; do madplay -o `basename $i .mp3`.wav $i; done
```

There is yet another way to do the conversion. Some MP3 files apparently give both **mpg123** and **madplay** trouble with decoding. The **lame** encoder, which has a decoding mode, seems to handle difficult cases very well (**lame** can be found at http://www.mp3dev.org/mp3/) :

```
for i in *.mp3; do lame --decode $i `basename $i .mp3`.wav; done
```

NOTE: The **`basename $i .mp3`.wav** command replaces MP3 extensions with WAV. There are 101 ways to do that, here's the alternative: **`echo "$1" | sed 's/\.mp3$/.wav/"`**

4. Run "**file *.wav**" and check the output for any files different from 16 bit, stereo 44100 Hz.
5. If there are files with different characteristics, convert them to the above specs. For example, to convert file track01.wav to obtain sample rate 44.1 kHz, you could use:
```
sox track01.wav -r 44100 track01-new.wav resample
```

Sox is so popular, that it's probably installed by default with any Linux distribution, and can be obtained from http://www.spies.com/Sox/. However, the command-line options are somewhat cryptic for the casual user (me). Look at http://www.spies.com/Sox/sox.tips.html for some tips on usage.

6. Normalize your WAV files, to avoid drastic differences in volume levels. I use a program by Chris Vaill (<cvaill@cs.columbia.edu>), called **normalize** – it can be obtained from http://www.cs.columbia.edu/~cvaill/normalize/

I use the following syntax (–m is for mix mode, where all files should be as loud as possible):

```
normalize -m *.wav
```

3. Burning Your CD

There are many programs to create CDs from WAV files. I use **cdrecord** for command−line burning and **XCDROAST** for gui. For **cdrecord**, you have to know what SCSI device your CD−writer is. If you're using ATAPI writer, use SCSI emulation (kernel module ide−scsi). Let's assume, that your ATAPI cdwriter is on the second IDE bus as a master. Thus, it will have `/dev/hdc` device file. To instruct the kernel that you want to treat it as a SCSI device, add the following line to `/etc/lilo.conf`:

```
append=" hdc=ide-scsi"
```

Also, if your kernel doesn't automatically load ide−scsi module, add **insmod ide−scsi** into your `rc.local` (or equivalent) file. Once you have our CD−writer recognized as a SCSI device, run **cdrecord −−scanbus** to find out what's the "dev" parameter to cdrecord. On my system, the output looks like the following:

```
scsibus1:
1,0,0 100) 'IOMEGA ' 'ZIP 250 ' '51.G' Removable Disk
1,1,0 101) 'HP ' 'CD-Writer+ 7100 ' '3.01' Removable CD-ROM
```

So, the **cdrecord** command line will contain **dev=1,1,0** to specify the device. Here is the complete command on my system:

```
cdrecord dev=1,1,0 -eject speed=2 -pad -audio *.wav
```

NOTE

The −pad argument is neccessary, because all audio tracks on the CD must be adjusted for the proper data length, which is not always the case with mp3 files.

4. Burning a DAO CD

DAO, Disc−At−Once, is as of now the only method for burning a CD without a 2−second pause between the tracks. It's useful for burning party mixes. The program for burning CDs in DAO mode is **cdrdao**, available from SourceForge, http://sourceforge.net/projects/cdrdao/.

The **cdrdao** program uses description files called *TOC* (Table Of Contents, of course). There are two ways to create such file. First is to use a shell script, distributed with **cdrdao** source (in `contrib` directory, called **generate_toc.sh**. It takes a list of `.wav` files as an argument and produces a `cd.toc` file. Second way is to simply create such file yourself in a text editor of your choice. Here is a self−explanatory example:

```
CD_DA

TRACK AUDIO
AUDIOFILE "mix-01.wav" 0

TRACK AUDIO
AUDIOFILE "mix-02.wav" 0

TRACK AUDIO
AUDIOFILE "mix-03.wav" 0

TRACK AUDIO
AUDIOFILE "mix-04.wav" 0

TRACK AUDIO
AUDIOFILE "mix-05.wav" 0
```

The *0* (zero) after the wave filename means start from the beginning of the file. There can be a second number providing the length (time) of file to record. The **xcdroast** creates similar *TOC* files, there are also examples in `testtocs` directory of **cdrdao** source.

The **cdrdao** by default uses the device `/dev/cdrecorder`, which should be a link to the cdwriter device. Assuming your cd recorder device file is `/dev/scd0`, create the link (as root) as follows:

```
ln -s /dev/scd0 /dev/cdrecorder
```

Then, assuming that the *TOC* file is named `cd.toc` the command to burn the cd is simply:

```
cdrdao write cd.toc
```

5. Credits

Special thanks to all the people who contribute to the Linux community and who made this HOWTO possible.

5.1. Translations

- Im Eunjea – Translated this document to Korean, URL is http://kltp.kldp.org/eunjea/mp3_burning/.
- Mendel L Chan – Translated this document to Chinese, URL is http://www.linux.org.tw/CLDP/mini/MP3−CD−Burning/.

5.2. Other Credits

- Greg Ferguson – Initially converted this document from HTML to SGML.
- Rob Russell – Corrected my name conversion example.
- Terry Davis – Suggested submitting my HOWTO to linuxdoc.
- Chris Vaill – Created **normalize** program.
- Jamie Kellogg – Submitted a solution to decode with **lame** for troublesome files.