Command-Line Guide to Audio Files in Ubuntu

This guide focuses on showing you how to manipulate and convert various audio files using tools included in the Ubuntu repositories. It contains mostly everything you need to know to convert various formats to the free format **Ogg Vorbis**, including explanations on how to convert **FLAC** to **WAV** and vice-versa, convert **FLAC**, **WAV**, **APE**, **AC3**, **WMA** or **SHN** to **Ogg Vorbis**, edit tags from command-line for free audio formats, and much more.

The guide is divided into the following sections, so that you can click on any of them to jump directly to the chapter which is of interest to you:

As a side note, this tutorial does **not** include references on how to edit or record sound (for example with Audacity or some other audio editor). Also, all the methods explained here make use of the command-line and command-line (or text user interface) tools. At the end of the guide there are several links to helpful links, a list of music players for Linux (both for GUI and CLI), as well as links to reviews for those.

Note: This tutorial works on both Debian and Ubuntu, and probably other distributions too, except for commands which pertain to the package manager (e.g. ***apt-get****). Commands which are prefixed with sudo should be ran as root in Debian (type* ***su****, enter the root password and then type the rest of the command without the* ***sudo*** *part.*

Introduction to Audio Manipulation on Linux

Free formats

In this tutorial I will focus mostly on manipulating and converting files to free formats, which in our case will be FLAC (Free Lossless Audio Codec) and Ogg Vorbis (free lossy codec), but support for MP3 is also included where it is the case. These formats are not patented and are free to use without the need to pay for using them. More details on this can be found on the official Xiph.org website.

I strongly encourage the use of free formats over those which are proprietary. First of all, using free formats will make your life easier on Linux, since all distributions include audio players which come with support for these by default, without the need to install additional codecs. There is also a drawback to this though: most portable players out there support MP3, but not Ogg. This isn't a very big issue because there are also portable players who support Ogg too in addition to MP3, and there are also solutions like Rockbox, which replaces the firmware on those players who can't read Ogg Vorbis or FLAC and adds support for those too.

The difference between lossless and lossy

Lossless formats are the ones which preserve the quality of the sound encoded at the expense of file size. Examples of such formats would be FLAC or WAV. On the other hand, lossy formats use lossy compression algorithms which create smaller files, but discard some of the audio data. However, on bitrates of 128 kbps and above the human ear cannot distinguish wether it's a lossy or a lossless sound they hear. Lossy formats include Ogg Vorbis and MP3. Use lossy to save disk space (for example for your personal music collection or for music on a portable player), and use lossless if you really need to preserve the original quality of the sound.

The Setup: Tools That We Need

In this section we will install all the tools you need for this tutorial, however notice that not all of these are needed to perform only a specific task, like converting from FLAC to Ogg Vorbis for example. In each section I will include again only the tools needed for that particular task, so it's not absolutely necessary to install everything listed here.

That being said, here are all the packages which need to be installed:

```
$ sudo apt-get install vorbis-tools flac cuetools shntool
```

Below is a brief overview of each package installed:

- **vorbis-tools** - this package contains Ogg Vorbis tools: **oggenc** (encoder), **ogg123** (Ogg Vorbis and FLAC command-line player), **ogginfo** (displays Ogg information), **oggdec** (decoder), **vcut** (Ogg file splitter), **vorbiscomment** (Ogg comment editor)
- **flac** - this package contains FLAC tools: **flac** (encoder/decoder) and **metaflac** (manipulates FLAC metadata)
- **cuetools** - this package contains tools used to manipulate CUE files; we will only need the **cuebreakpoints** tool from this package
- **shntool** - this is a utility for processing WAV files

Converting FLAC to WAV and Vice-Versa

Prerequisites

For this we will need to install only the **flac** package:

```
$ sudo apt-get install flac
```

To convert a FLAC file to WAV we would issue the following command:

```
$ flac -d input_file.flac
```

The **-d** switch tells flac to decode the input file specified as argument. Here's an example:

```
$ file input_file.flac
input_file.flac: FLAC audio bitstream data, 16 bit, stereo, 44.1 kHz,
13563984 samples
embryo ~/tut_audio $ flac -d input_file.flac

flac 1.2.1, Copyright (C) 2000,2001,2002,2003,2004,2005,2006,2007  Josh
Coalson
flac comes with ABSOLUTELY NO WARRANTY.  This is free software, and you are
welcome to redistribute it under certain conditions.  Type `flac' for details.

input_file.flac: done
embryo ~/tut_audio $ file input_file.wav
```

The simplest command to convert from WAV to FLAC is this:

```
$ flac input_file.wav
```

This will create the file **input_file.flac**. You can also specify the output filename, e.g.:

```
$ flac input_file.wav -o output_file.flac
```

You can also specify a compression level (from 1 to 8, default is 5). The higher the compression level, the smaller the resulting file will be:

```
$ flac -8 input_file.wav -o output_file.flac
```

This will produce **output_file.flac** with the highest compression level (8).

Converting FLAC or WAV to Ogg Vorbis

Ogg Vorbis is a patent-free, lossy audio format which is very well supported on all the major Linux distributions. Ogg Vorbis is the format of choice for many open-source games and applications.

In the following example I'll show how to convert FLAC to Ogg Vorbis (the exact same command applies for WAV too) using several common options. For this we will use the **oggenc** tool, which is the official Ogg encoder:

```
$ oggenc input_file.flac
```

This will produce an Ogg file called **input_file.ogg** using the default quality (3). The quality can be anywhere between **-1** (~ 45 kbps) and **10** (~ 500 kbps). The default quality level is 3 (~ 112 kbps). Fractional quality levels are permitted too. We can also specify the output file name. For example, the following command will encode a FLAC file to Ogg Vorbis using a quality level of 6 (~ 192 kbps). The resulting file will be called **output_file.ogg**:

```
$ oggenc -q 6 input_file.flac -o output_file.ogg
```

Instead of specifying a quality factor, we can specify the bitrate:

```
$ oggenc -b 192 input_file.flac -o output_file.ogg
```

Editing Ogg Vorbis Tags

Prerequisites

For this we will only need to install the **vorbis-tools** package, which includes the **vorbiscomment** tool used for editing tags:

```
sudo apt-get install vorbis-tools
```

Using 'vorbiscomment'

vorbiscomment is an utility which comes with the vorbis-tools package and which allows us to edit or remove Ogg Vorbis tags like song title, artist, album, year, track and so on. Custom fields are also possible. Let's see how it works by adding editing a new tag, called "TITLE" and adding a value for this tag, say "Wasted Years":

```
$ vorbiscomment -t "TITLE=Wasted Years" -w input_file.ogg
```

The above command will use the **-t** (or **--tag**) switch to add the tag **TITLE** with the value **Wasted Years**, and write it (**-w**) to input_file.ogg. To see the result, use:

```
$ vorbiscomment -l input_file.ogg
```

The **-l** switch is used to list tags in a Vorbis file. Notice that the **-w** switch will rewrite the file and any existing tags will be lost. In order to append a tag to existing ones, use **-a** instead of **-w**.
An alternate way of doing it is to read tags from a text file. For example:

```
$ vorbiscomment -c file.txt -w input_file.ogg
```

And the file **file.txt** could contain tags, one per line, like this:

```
ARTIST=Iron Maiden
TITLE=Wasted Years
ALBUM=Somewhere in Time
YEAR=1986
TRACKNUMBER=2
```

The easiest way to clear all the tags is to create an empty file, and then use vorbiscomment to read and write the contents of it to the Ogg Vorbis file, e.g.:

```
$ touch empty_file
$ vorbiscomment -c empty_file -w input_file.ogg
```

What follows is a short script to remove all the tags from all Vorbis files in a directory:

```
#!/bin/bash

echo "Ogg Tag Remover"
echo "Creating empty file..."
touch empty_file
```

```
echo "Removing all tags in Ogg files..."
for i in *.ogg; do
    vorbiscomment -c empty_file -w "$i"
done

echo "Removing empty file..."
rm empty_file
echo "Done!"
```

And here is another script which automatically fills the **TRACKNUMBER** tag for each Ogg Vorbis file in a directory:

```
#!/bin/bash

echo "Ogg TRACKNUMBER Editor"

n=1
for i in *.ogg; do
    vorbiscomment -t "TRACKNUMBER=$n" -a "$i"
    n=$((n + 1))
done

echo "Done!"
```

This time we used the **-a** switch to **append** this tag to existing tags. If we would've used **-w**, the file would've been rewritten and the existing tags lost.

Converting FLAC or WAV to MP3

Prerequisites

For encoding FLAC or WAV to MP3 we will use **lame** (http://lame.sourceforge.net/), a free MP3 encoder. To install it in Ubuntu type in a terminal:

```
sudo apt-get install lame
```

To install it in Debian, add the Debian Multimedia repositories to your **/etc/apt /sources.list** file (follow the instructions on the site) and issue the following commands as root:

```
apt-get update
apt-get install lame
```

Converting FLAC or WAV to MP3 Using LAME

If our files are in FLAC format we will first need to convert them to WAV (e.g. **flac -d input_file.flac -o output_file.wav**). Next, the simplest way to convert WAV to MP3 with LAME is like this:

```
lame input_file.wav
```

This will result in an MP3 file called **input_file.wav.mp3** encoded at 128 kbps. You can specify the output file name like this:

```
lame input_file.wav output_file.mp3
```

To specify a certain bitrate, use the **-b** switch:

```
lame -b 192 input_file.wav output_file.mp3
```

Unfortunately lame doesn't support wildcards, so converting using a command such as **lame [0-9]*.wav** will not work, you will have to use something like the following one-liner:

```
for i in *.wav; do lame $i; done
```

Again, do **not** use wildcards with lame. If for example you have two WAV files in a directory called, say, file1.wav and file2.wav and issue the command **lame *.wav**, this will expand into **lame file1.wav file2.wav**, and lame will interpret file2.wav as the output filename after converting file1.wav. This will result in file2.wav being rewritten by an MP3 file.

Ripping Audio CDs

Prerequisites

The tool for ripping audio CDs is **cdparanoia** (http://www.xiph.org/paranoia/). To install it use:

```
sudo apt-get install cdparanoia
```

We will also need **id3v2**, an ID3v2 tag viewer and editor (this is only used by the **abcde** tool):

```
sudo apt-get install id3v2
```

Ripping Audio CDs With cdparanoia

The simplest way to use cdparanoia is just like this:

```
cdparanoia -B
```

The **-B** switch will tell cdparanoia to split the output into multiple WAV files. The files will have the name starting with **trackN.cdda.wav** and followed by a number (e.g. **track01.cdda.wav**).

Additional parameters can be passed to cdparanoia, for example to rip a single track:

```
cdparanoia -B 3
```

Or to rip, say, tracks 4-9 we would use:

```
cdpranoia -B 4-9
```

You can also rip a certain track from a specified time up to another specified time:

```
cdparanoia "5[:20.00]-5[5:30.00]"
```

This will rip track 5 starting at second 20 and stopping at minute 5 and 30 seconds.

The files will be in WAV format by default, so you can convert them to FLAC, to Ogg Vorbis or to MP3 if you like.

Using abcde

abcde (A Better CD Encoder) is a frontend to cdpranoia and several other tools which allows you to rip audio CDs, encode and tag the resulting files. Here's an example of using abcde:

```
$ abcde -d /dev/cdrom -o mp3
```

Which will rip audio CD from device /dev/cdrom and encode the output to MP3 format.

Split FLAC, WAV or APE Using With a CUE File

Prerequisites

For this we will need three packages: **cuetools**, which contains the **cuebreakpoints** utility for reading CUE files, **shntool**, a WAV processing tool, and **monkeys-audio**, a package which contains the **mac** utility, for creating Monkey's Audio lossless files. To grab them type in a terminal:

```
$ sudo apt-get install cuetools shntool monkeys-audio
```

Split FLAC/WAV Using a CUE File

A CUE file will hold the information about starting and ending time of a track, information which can be used to split larger files (like an album which comes as a single FLAC or WAV file instead of individual tracks). The command to split these is the following (I used .flac in this example, but the same goes for WAV):

```
$ cuebreakpoints cue_file.cue | shnsplit audio_file.flac
```

You can also edit the CUE file using a text editor and alter track lengths, for example for concatenating two melodies or more in a single one.

Split APE Using a CUE File

First we will convert our APE file to WAV:

```
$ mac input_file.ape output_file.wav -d
```

If you have more than one APE files inside a directory, use the following one liner:

```
$ for in *.ape; do mac "$i" "$i.wav" -d; done
```

Now we can proceed and split the WAV just like we did in the section above:

```
cuebreakpoints cue_file.cue | shnsplit output_file.wav
```

Converting WMA to Ogg Vorbis

Prerequisites

For this one we will need a tool called **dir2ogg**, which is a Python script that allows us to convert formats like MP3, M4A and WAV to Ogg Vorbis:

```
sudo apt-get install dir2ogg
```

Converting WMA to Ogg Vorbis

Using dir2ogg is very simple:

```
dir2ogg input_file.wma
```

Or, for all the WMA files in a directory:

```
dir2ogg *.wma
```

Converting APE to Ogg Vorbis or MP3

Prerequisites

We will need the **vorbis-tools** (for Ogg Vorbis encoding), **lame** (for MP3 encoding) and **monkeys-audio** (for APE) packages:

```
$ sudo apt-get install vorbis-tools lame monkeys-audio
```

In Debian, add the Debian Multimedia repositories to install **lame**.

Converting APE to Ogg Vorbis or MP3

First we will use the **mac** tool (the command-line APE decoder frontend) to convert APE to WAV:

```
$ mac input_file.ape output_file.wav -d
```

If you have several APE files in a directory and you want to convert them all, use this one-liner:

```
for i in *.ape; do mac "$i" "$i.wav" -d; done
```

Now, we'll convert WAV to Vorbis the normal way:

```
oggenc -b 256 input_file.wav
```

Or, for MP3:

```
lame -b 256 input_file.wav
```

Converting AC3 to WAV Using MPlayer

Prerequisites

We will need to install **mplayer** (http://www.mplayerhq.hu/design7/news.html):

```
sudo apt-get install mplayer
```

Converting AC3 to WAV

The command we will use goes like this:

```
mplayer -ao pcm:file=output_file.wav input_file.ac3
```

This will convert input_file.ac3 to output_file.wav using the PCM audio output.

Music Players

Music players

There are many good music players for Linux, for both GUI and terminal. I'll mention a few graphical ones like:

- **Amarok** - Very powerful and popular music player for KDE
- **Rhythmbox** - The default music player in GNOME
- **Banshee** - Powerful GTK-based music player
- **Clementine** - A port of the classic Amarok 1.4 to KDE4
- **Jajuk** - Java-based, full-featured audio player
- **Qmmp** - XMMS-like audio player for KDE
- **Audacious** - XMMS-like audio player for GNOME

And here are several which run in a terminal, and are either command-line or text user interface (TUI):

- **CMus** - feature-rich player using ncurses text user interface
- **moc** - yet another music player using TUI
- **mp3blaster** - powerful console music player
- **ogg123** - command-line player for Ogg Vorbis and FLAC
- **mpg123** - command-line player for MP3 and various other formats
- **mplayer** - very powerful audio and video player

These are only a few though. Recently I put up a comprehensive list with reviews of no less than 16 music players. The article is located on the TuxArena Blog, here.

In addition to these, there are also client-server players like MPD or XMMS2, with clients like Sonata or GMPC.

If you have suggestions or corrections to these tutorials, please contact me at craciun.dan@tuxarena.com or leave a comment on the TuxArena website.