LINUX VOICE MASTERCLASS

Without music, life would be a mistake. Join us as we celebrate the new UK legal status of ripping your own CDs.

RUBY RIPPER & PICARD MANAGE YOUR MUSIC

Efficient music management is one of those things that calls for a GUI application…

JOHN LANE

LV PRO TIP

Ruby Ripper also has a command-line mode: **rrip_cli**.

Today's portable media devices and smartphones have enough storage to allow most people to carry their entire music collection in their pocket. But if yours includes a CD collection, then you'll need to rip those discs into digital audio files that your devices can use.

This is a three-step process: you need to extract, or rip, the audio from the disk and encode it into suitably formatted data files that you can then tag with metadata that describes them. If your preference is for GUI tools, there are many options available to help you build and maintain your music collection, but there isn't any one-size-fits-all solution that does everything. You'll still need one tool for ripping and another for tagging and organising your rips, but this does allow you to select your preference in each area.

The default ripper on Gnome-based systems is called *Sound Juicer*, which can rip an audio CD into various formats, but only one at a time. KDE users have *K3b* and other possible alternatives include applications called *Grip* and *Asunder*. These will all rip your CDs quickly and easily, but we're going to use *Ruby Ripper*. This uses the **cdparanoia** command line tool, but goes further to ensure accuracy by ripping each track at least twice and comparing them.

If you're an Ubuntu user, you'll need to add a third-party repository to install *Ruby Ripper*, because it isn't officially supported:

```
$ wget -q -O - http://archive.getdeb.net/getdeb-archive.key |
sudo apt-key add -
$ sudo sh -c 'echo "deb http://archive.getdeb.net/ubuntu
trusty-getdeb apps" >> /etc/apt/sources.list.d/getdeb.list'
$ sudo apt-get update
$ sudo apt-get install rubyripper
```

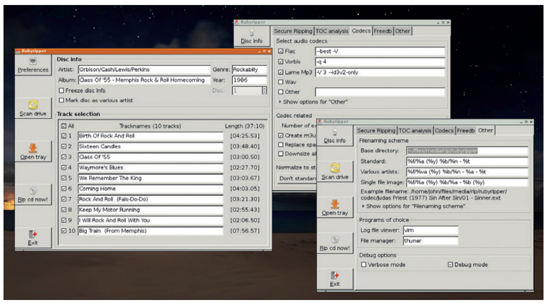
Arch Linux users have life a little easier:

```
$ pacman -S ruby-ripper
```

Once you've installed *Ruby Ripper*, insert a CD and launch the GUI application:

```
$ rrip_gui
```

The disc is scanned and looked up on the FreeDB



Disc info
Artist: Orbison/Cash/Lewis/Perkins Genre: Rockabilly
Album: Class Of '55 - Memphis Rock & Roll Homecoming Year: 1986
☐ Freeze disc info Disc: 1
☐ Mark disc as various artist

Track selection

☑ All	Tracknames (10 tracks)	Length (37:10)
☑ 1	Birth Of Rock And Roll	[04:25.53]
☑ 2	Sixteen Candles	[03:48.40]
☑ 3	Class Of '55	[03:00.50]
☑ 4	Waymore's Blues	[02:27.70]
☑ 5	We Remember The King	[03:03.67]
☑ 6	Coming Home	[04:03.05]
☑ 7	Rock And Roll (Fais-Do-Do)	[03:21.30]
☑ 8	Keep My Motor Running	[02:55.43]
☑ 9	I Will Rock And Roll With You	[02:06.50]
☑ 10	Big Train (From Memphis)	[07:56.57]

Select audio codecs
☑ Flac --best -V
☑ Vorbis -q 4
☑ Lame Mp3 -V 3 --id3v2-only
☐ Wav
☐ Other
Show options for "Other"

Codec related
Number of e
☑ Create m3u
☐ Replace spa
☐ Downsize al
Normalize to st
Don't standard

Filenaming scheme
Base directory: ~/files/media/rip/rubyripper
Standard: %f/%a (%y) %b/%n - %t
Various artists: %f/%va (%y) %b/%n - %a - %t
Single file image: %f/%a (%y) %b/%a - %b (%y)
Example filename: /home/john/files/media/rip/rubyripper/codec/Judas Priest (1977) Sin After Sin/01 - Sinner.ext
Show options for "Filenaming scheme"

Programs of choice
Log file viewer: vim
File manager: thunar

Debug options
☐ Verbose mode ☑ Debug mode

With *Ruby Ripper* you can edit your CD's track names before ripping and use the preferences pages to select encoders and define how files should be written.

music database (if there are multiple matches you can choose which one to use). You're presented with a simple dialog that lists CD and track details and you can edit these if necessary prior to hitting the Rip CD Now button to begin ripping.

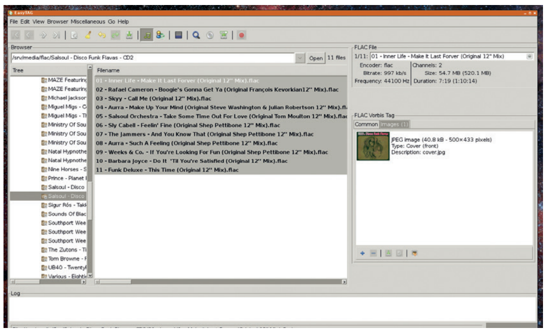
GUI CD ripping tools perform lookups to FreeDB and tag the ripped tracks accordingly. However, a dedicated tagger will ultimately give you more control over the process.

EasyTag is a feature-rich tagging tool that allows you you tag files, rename them and organise them into directories. It enables you to view, add or remove album cover art but doesn't have a facility to locate it (**http://albumart.org** can help with this). *EasyTag* is in most distros' default repositories.

EasyTag makes it easy to perform bulk operations. It can generate metadata from a file's path and tag files with it. It can use existing metadata or fetch it from FreeDB or MusicBrainz (another music database) and use it to re-tag, move and rename files.

The Brainz behind the music

MusicBrainz is, in its own words, "an open music encyclopedia that collects music metadata and makes it available to the public". It's similar to FreeDB and CDDB, which preceded it, but takes the concept



/srv/media/flac/Salsoul - Disco Funk Flavas - CD2 — 11 files

MAZE Featurin
MAZE Featurin
Michael Jackson
Miguel Migs - C
Miguel Migs - Tl
Ministry Of Sou
Ministry Of Sou
Ministry Of Sou
Natal Hypnothe
Natal Hypnothe
Nine Horses - S
Prince - Planet I
Salsoul - Disco
Salsoul - Disco
Sigur Rós - Tak
Sounds Of Blac
Southport Wee
Southport Wee
Southport Wee
The Zutons - Ti
Tom Browne - F
UB40 - Twenty
Various - Eighti

01 - Inner Life - Make It Last Forver (Original 12" Mix).flac
02 - Rafael Cameron - Boogie's Gonna Get Ya (Original François Kevorkian12" Mix).flac
03 - Skyy - Call Me (Original 12" Mix).flac
04 - Aurra - Make Up Your Mind (Original Steve Washington & Julian Robertson 12" Mix).fl
05 - Salsoul Orchestra - Take Some Time Out For Love (Original Tom Moulton 12" Mix).flac
06 - Sly Cabell - Feelin' Fine (Original Shep Pettibone 12" Mix).flac
07 - The Jammers - And You Know That (Original Shep Pettibone 12" Mix).flac
08 - Aurra - Such A Feeling (Original Shep Pettibone 12" Mix).flac
09 - Weeks & Co. - If You're Looking For Fun (Original Shep Pettibone 12" Mix).flac
10 - Barbara Joyce - Do It 'Til You're Satisfied (Original 12" Mix).flac
11 - Funk Deluxe - This Time (Original 12" Mix).flac

1/11: 01 - Inner Life - Make It Last Forver (Original 12" Mix)
Encoder: flac | Channels: 2
Bitrate: 997 kb/s | Size: 54.7 MB (520.1 MB)
Frequency: 44100 Hz | Duration: 7:19 (1:10:14)

JPEG image (40.8 kB - 500×433 pixels)
Type: Cover (front)
Description: cover.jpg

EasyTag auto-corrects tags to add missing spaces as well as updating the ID3 tag version on MP3 files. Changed files that need to be saved are highlighted.

further by providing a relational database of music (not just compact discs). All its data is either in the public domain or released under a Creative Commons Non-Commercial Share-Alike licence.

It provides a very capable application called *Picard*, which enables you to tag your music files with data from the MusicBrainz database. It has lookup tools that use a disc ID to locate albums in the database. Where matches cannot be found, it integrates with your browser to provide a seamless interface between the MusicBrainz website and the *Picard* application.

It's easy to extend Picard's functionality with plugins available at **https://musicbrainz.org/doc/MusicBrainz_Picard/Plugins**. We recommend 'Add Cluster as Release' because it simplifies adding albums to MusicBrainz, but there are several to choose from or you can write your own in Python.

Make it so

Begin a ripping session by opening *Picard*. It has a status line that lets you know what is happening, and you can get more feedback by running it from a terminal in debug mode. Should you want to do that, begin with:

```
$ picard -d
```

Place a CD in your drive and use picard's "Lookup CD" to see if it's already in the MusicBrainz database. This displays a pop-up containing any matches so that you can select one. Alternatively, you can press a 'Lookup Manually' button to open a browser where you can perform a manual search. *Picard* listens for updates (usually on port 8000) and sends its port number in the browser request. MusicBrainz displays an additional 'Tagger' link, a green icon, that you can use to send your selection back to *Picard*.

Fire up *Ruby Ripper* to rip the CD according to your needs (we ripped into Flac, Ogg and MP3). If you found a MusicBrainz match you needn't worry too much about the quality of the FreeDB data returned by your ripper, because *Picard* will re-tag your files.

Navigate *Picard*'s file browser (the left-hand panel - do Ctrl+B if it isn't visible) to the directory where your ripped files were written, drag it into the centre panel and then hit the Cluster button. This groups files into clusters based on common metadata (your ripping process should have tagged the files using metadata from FreeDB). You should see a single cluster containing all of the tracks from the ripped album; there'll be multiple files for each track if you ripped into multiple formats.

Now is the time to add a release to MusicBrainz if you were unable to find a MusicBrainz match before ripping. If you right-click the cluster and choose Plugins-> Add Cluster As Release, Picard will open the MusicBrainz Add Release page in your browser and pre-populate it using the ripped files' metadata.

The page has several tabs that you must navigate to supply all of the required information. You should try to supply an external link as a cross-reference. You can reference pages from Discogs and/or Amazon; the latter will also be used as a source for album artwork.

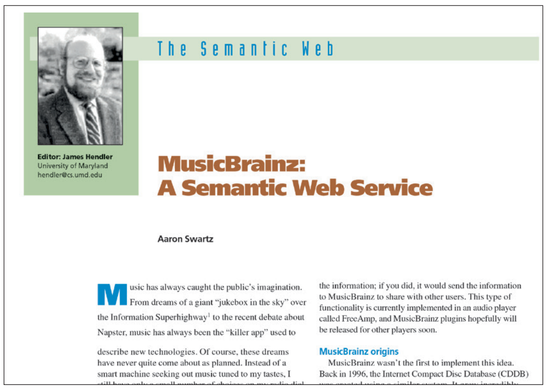
The final Edit Note tab is where you'll find the the Enter Edit button, which submits the new release. Before adding a release for the first time, you should familiarise yourself with what is expected by reading the MusicBrainz Beginners' Guide at **https://musicbrainz.org/doc/Beginners_Guide**.

It displays the new release in your browser so that you can check it or augment it further, perhaps supplying album art if none was found automatically. You should click the green 'Tagger' icon to load the new release into *Picard*, where it should appear in the panel on the right-hand side.

The final step is to tag the clustered files with the release. Open the release by double-clicking on it; this will display the tracks beneath. Now, drag the cluster and drop it on top of the tracks (it doesn't matter which track the cluster lands on) and then right-click the release and refresh, which moves the files onto the correct tracks. You can review the metadata for each file before saving (another right click on the release).

Depending on your configuration (Options > Options > File Naming), *Picard* can also rename the ripped files and copy or move them into a new location.

Keep another tagging tool, such as EasyTag or Ex Falso, close by – it can be easier to perform other edits outside *Picard* should you need to use tags that *Picard* doesn't.



The Semantic Web

Editor: James Hendler
University of Maryland
hendler@cs.umd.edu

MusicBrainz: A Semantic Web Service

Aaron Swartz

Music has always caught the public's imagination. From dreams of a giant "jukebox in the sky" over the Information Superhighway[1] to the recent debate about Napster, music has always been the "killer app" used to describe new technologies. Of course, these dreams have never quite come about as planned. Instead of a smart machine seeking out music tuned to my tastes, I

the information; if you did, it would send the information to MusicBrainz to share with other users. This type of functionality is currently implemented in an audio player called FreeAmp, and MusicBrainz plugins hopefully will be released for other players soon.

MusicBrainz origins

MusicBrainz wasn't the first to implement this idea. Back in 1996, the Internet Compact Disc Database (CDDB)

MusicBrainz, through its website and its *Picard* tagging application, gives you rich metadata to tag your music with and makes it easy to contribute data yourself.

LV PRO TIP

To get the best from MusicBrainz and *Picard*, you should register as a user on the MusicBrainz website.

LV PRO TIP

New MusicBrainz users are followed by moderators for a short period to help maintain data quality. Consider this as having a free mentor to help get you started!

RIP CDS FROM THE COMMAND LINE WITH CDPARANOIA

Use the command line to rip, tag and organise your CD collection.

JOHN LANE

Back in the day, CD ripping was performed with a tool called **cdda2wav**, but this has been improved upon by **cdparanoia**, which can recover from disk read errors including some disk scratches. It's our ripping tool of choice, and you should find it in your distro's repository. Pop a CD in your drive and try it out

```
$ cdparanoia -B
```

This rips the audio CD **/dev/cdrom** and produces WAV format files in the current directory. You'll see something like this for each file:

```
outputting to track02.cdda.wav
 (== PROGRESS == [                          | 031371 00 ] == :^D * ==)
```

The progress bar keeps you informed of what's happening; your rip should be fine if you don't see **!** or **V** characters.

"cdparanoia is our CD ripping tool of choice, and you should find it in your distro's repository."

The WAV files contain the uncompressed digital data, but there are many other, more suitable, formats like MP3, Ogg Vorbis and FLAC. There are separate encoders for each format, and a little *Bash* lets us process each track into the formats we want:

```
$ for f in track*.cdda.wav; do
> oggenc $f
> lame $f
> flac $f
> done
```

The default encoder settings are acceptable, but if you want more control over quality vs compression you can check the commands' man pages for their respective arguments, which enable you to customise the encoding process.

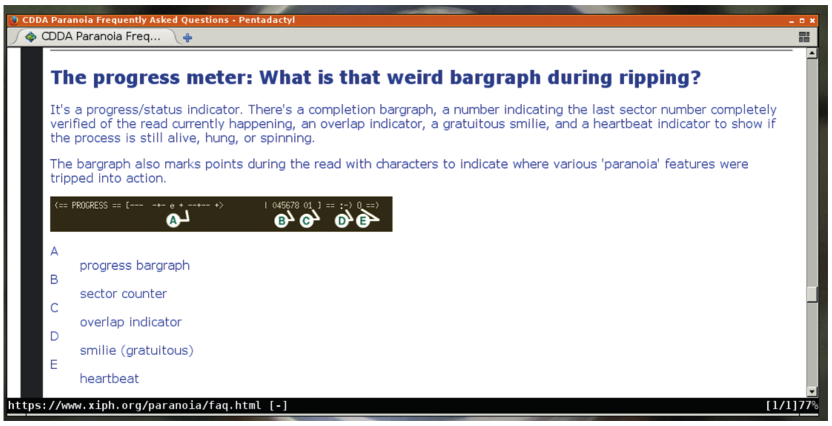
We now have the tracks off the CD in our desired formats but we yet don't know what they are. They all have names like **track01.cdda.flac**. We could look at the CD sleeve and rename the files by hand or we can identify the disc and look it up in a database. A disc ID is a 32-bit number, presented as eight hexadecimal digits, which can be used to identify an audio CD. It is derived from the starting times of each track and the total playing time of the disc. The **cd-discid** tool will show the disc ID and track offset data for a CD:

```
$ cd-discid /dev/cdrom
900c330c 12 150 17142 31490 57867 84507 105025 120762
135657 152745 179847 198320 217650 3125
```

You can use the disc ID to perform a search in various CD databases. These are mostly community-maintained databases that are free to use and have open-source licences. They were forked from an open CD Database called CDDB that has since become a proprietary service called Gracenote. Due to their common origin, the forked services use a similar protocol now known as the FreeDB CDDB Protocol.

You can use Telnet to access a CDDB protocol server on port 8880:

```
$ telnet freedb.freedb.org 8880
201 mirror1.freedb.org CDDBP server v1.5.2PL0 ready
> cddb hello foo bar baz 1
200 Hello and welcome foo@bar running baz 1.
> cddb query 900c330c 12 150 17142 31...
200 misc 900c330c Pulp / Different Class
> cddb read misc 900c330c
210 misc 900c330c CD database entry follows (until terminating
`.')
# xmcd CD database file
#
# Track frame offsets:
```



CDDA Paranoia Frequently Asked Questions - Pentadactyl

The progress meter: What is that weird bargraph during ripping?

It's a progress/status indicator. There's a completion bargraph, a number indicating the last sector number completely verified of the read currently happening, an overlap indicator, a gratuitous smilie, and a heartbeat indicator to show if the process is still alive, hung, or spinning.

The bargraph also marks points during the read with characters to indicate where various 'paranoia' features were tripped into action.

```
(== PROGRESS == [---  -+- e + --+-- +>          | 045678 01 ] == :-) 0 ==)
```

A
progress bargraph
B
sector counter
C
overlap indicator
D
smilie (gratuitous)
E
heartbeat

https://www.xiph.org/paranoia/faq.html [-] [1/1]77%

It's official: **cdparanoia** has a weird bargraph. Read the FAQ to understand it.

Audio CD duplication

If you want to make a physical copy of an audio CD the process is different; you don't rip the tracks, but instead make a disc at once copy. The tool for this is **cdrdao**:

```
$ cdrdao read-cd --datafile mycd.bin mycd.toc
```

You cannot use data tools like **dd** to copy audio CDs because they don't have a filesystem. You don't create an ISO file, but instead a pair of files: the **.bin** is the disc image and its **.toc** is a text file that contains its table of contents and describes the track layout. You can augment the table of contents with track titles and artist names from FreeDB:

```
$ cdrdao read-cddb mycd.toc
```

You can use these files to write to a recordable CD:

```
$ cdrdao write mycd.toc
```

CD-text will also be written to the disc if the table of contents contains FreeDB data.

```
#          150
#          17142
...
> quit
```

When you first connect, you must send a 'hello'. It expects a user and hostname and the name of the connecting client software and version but, as our example shows, it doesn't matter what you send. Next, you issue a 'query', which takes the output of **cd-discid**. You then use the returned genre and disc ID values to perform a 'read' (if you already know these then you can omit the query). The resulting display contains various information, including the track names that you can use to rename your files.

Another way to do this is to send an HTTP query from your web browser or use **curl**, which is probably the best approach if you're writing a script. The commands used above are encapsulated in a query string; the following example performs the same query as above:

```
$ curl "http://freedb.freedb.org/~cddb/cddb.cgi?cmd=cddb+query+$(cd-discid | sed 's/ /+/g')&hello=foo+bar+baz+1&proto=6"
```

Spaces in the query need to be represented using "+" so the output of "cd-discid" is passed through "sed" to perform this conversion. Performing the read operation is similarly straightforward:

```
$ curl 'http://freedb.freedb.org/~cddb/cddb.cgi?cmd=cddb+read+misc+900c330c&hello=foo+bar+baz+1&proto=6'
```

Tag art

Correctly naming files isn't enough, because most media players rely on metadata contained within the files to identify them. You can add these tags with command-line utilities. Again, each format has its own tagger:

```
$ eyeD3 -A "Mis-Shapes" -G "BritPop" -Y 1995 01:Mis-Shapes.mp3
$ vorbiscomment -w -t 'TITLE=Mis-Shapes' -t 'GENRE=BritPop' 01:Mis-Shapes.ogg
$ metaflac --set-tag 'TITLE=Mis-Shapes' --set-tag 'GENRE=BritPop' 01:Mis-Shapes.flac'
```

You can also attach images as tags. If you have album cover art images, you can embed these within the files' metadata. It's easy for MP3 and FLAC files:

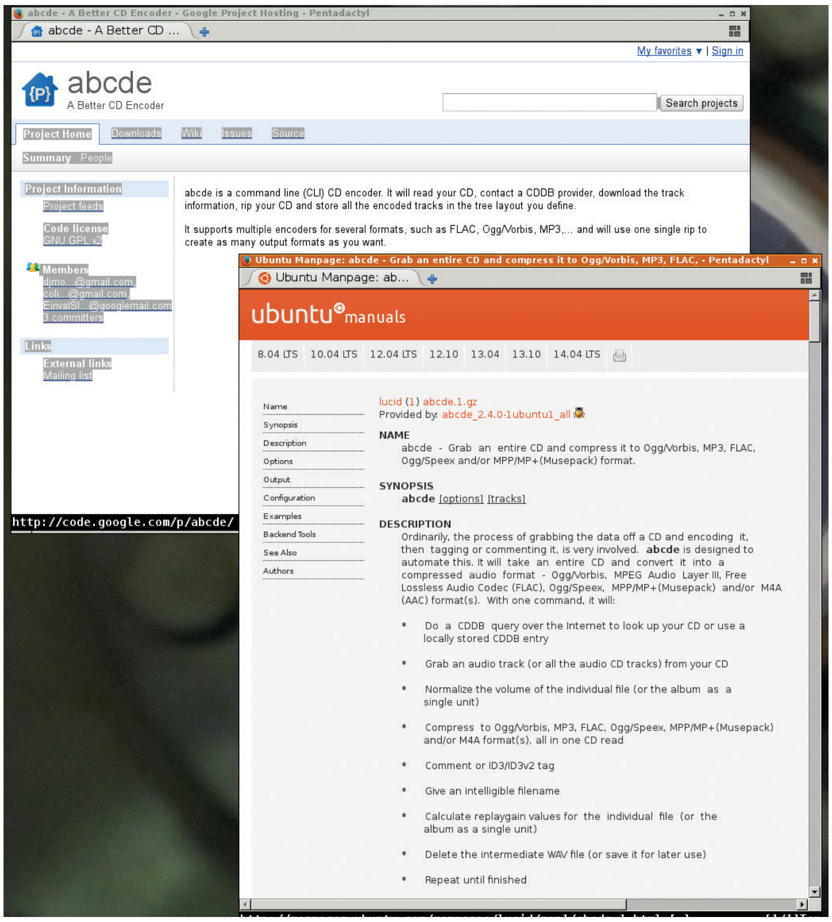
```
$ eyeD3 --add-image cover.jpg:FRONT_COVER 01:Mis-Shapes.mp3
$ metaflac --import-picture-from cover.jpg 01:Mis-Shapes.flac'
```

Ogg Vorbis requires the image to be base-64 encoded, which requires a little more effort and the **base64** command-line utility:

```
$ vorbiscomment -a -t "COVERART=$(base64 --wrap=0 < cover.jpg)" 01:Mis-Shapes.ogg
```

As easy as abcde...

If all this sounds too hard and you want something to take the pain away, you can try 'A Better CD Encoder', or **abcde**. This is a *Bash* script that wraps the tools that we've been using and makes using them painless and it's in most distributions' repositories.



abcde is a command line (CLI) CD encoder. It will read your CD, contact a CDDB provider, download the track information, rip your CD and store all the encoded tracks in the tree layout you define.

It supports multiple encoders for several formats, such as FLAC, Ogg/Vorbis, MP3,... and will use one single rip to create as many output formats as you want.

lucid (1) abcde.1.gz
Provided by: abcde_2.4.0-1ubuntu1_all

NAME
abcde - Grab an entire CD and compress it to Ogg/Vorbis, MP3, FLAC, Ogg/Speex and/or MPP/MP+(Musepack) format.

SYNOPSIS
abcde [options] [tracks]

DESCRIPTION
Ordinarily, the process of grabbing the data off a CD and encoding it, then tagging or commenting it, is very involved. **abcde** is designed to automate this. It will take an entire CD and convert it into a compressed audio format - Ogg/Vorbis, MPEG Audio Layer III, Free Lossless Audio Codec (FLAC), Ogg/Speex, MPP/MP+(Musepack) and/or M4A (AAC) format(s). With one command, it will:

* Do a CDDB query over the Internet to look up your CD or use a locally stored CDDB entry
* Grab an audio track (or all the audio CD tracks) from your CD
* Normalize the volume of the individual file (or the album as a single unit)
* Compress to Ogg/Vorbis, MP3, FLAC, Ogg/Speex, MPP/MP+(Musepack) and/or M4A format(s), all in one CD read
* Comment or ID3/ID3v2 tag
* Give an intelligible filename
* Calculate replaygain values for the individual file (or the album as a single unit)
* Delete the intermediate WAV file (or save it for later use)
* Repeat until finished

It's easy to rip from the command line. Look no further than your distro's repository.

The easiest way to use it is to customising its configuration file, **/etc/abcde.conf**, and save it as **~/.abcde.conf**. Once you've done this, you can rip, encode and tag to multiple formats with a single command. Put a CD in your drive and do

```
$ abcde
```

abcde rips your CD, performs a CDDB lookup, allows you to choose the match you want in the even of the search returning multiple matches, and gives you the chance to edit the data before use. It encodes to multiple formats and stores the files using a file- and directory-naming convention of your choice. You control all of this from A Better CD Encoder's configuration file.

The default configuration is mostly acceptable but you may wish to customise how it names files and directories. Look at the first 50 or so lines of the file and edit to suit your needs. The ones you'll most-likely want to alter include the **OUTPUTTYPE**, which specifies the desired output formats, and the **OUTPUTDIR** and **OUTPUTFORMAT** entries, which define where files are written and how they are organised into directories and named. The **CDDBURL** enables you to specify the FreeDB service that you want to use.

abcde greatly simplifies command-line CD ripping and is probably all you'll really need, unless you want to add a GUI tagging tool into the mix.

LV PRO TIP

Some CDs contain track and artist data as 'CD-Text' that you can see with a tool called **cd-info**.