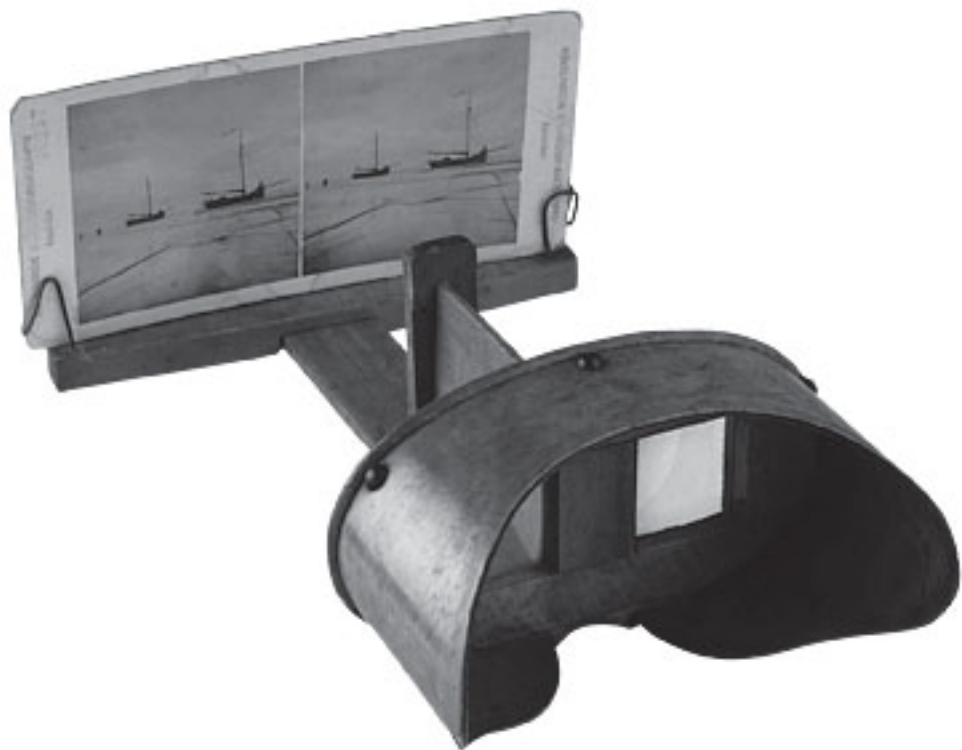


LINUX MULTIMEDIA HACKS™

*Tips & Tools for Taming Images,
Audio, and Video*



O'REILLY®

Kyle Rankin

Linux Multimedia Hacks™

by Kyle Rankin

Copyright © 2006 O'Reilly Media, Inc. All rights reserved.
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor:	David Brickner	Production Editor:	Jamie Peppard
Series Editor:	Rael Dornfest	Cover Designer:	Marcia Friedman
Executive Editor:	Dale Dougherty	Interior Designer:	David Futato

Printing History:

November 2005: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. The *Hacks* series designations, *Linux Multimedia Hacks*, the image of a stereoscope, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Small print: The technologies discussed in this publication, the limitations on these technologies that technology and content owners seek to impose, and the laws actually limiting the use of these technologies are constantly changing. Thus, some of the hacks described in this publication may not work, may cause unintended harm to systems on which they are used, or may not be consistent with applicable user agreements. Your use of these hacks is at your own risk, and O'Reilly Media, Inc. disclaims responsibility for any damage or expense resulting from their use. In any event, you should take care that your use of these hacks does not violate any applicable laws, including copyright laws.



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 0-596-10076-0

[M]

Audio

Hacks 13–46

Many people in the Linux community believe that choice is good. Perhaps second only to text editors, audio applications under Linux truly embody this philosophy. Even if you limit yourself to strictly audio players, you don't have enough fingers and toes to count all the programs out there. XMMS, Rhythmbox, amaroK, juK, and others are on the GUI end; hybrid programs can play both audio and video files; and command-line tools such as *mpg123* and *mpg321*, may also have GUI frontends.

Everyone listens to music differently. Some people listen to a single CD or album, every track in sequence. Others put their entire music collection on shuffle. Some people like minimalist audio players while others want every option and effect under the sun. With all the audio choices under Linux, no matter how you like to listen to your music, Linux has an application that fits. Beyond simply listening to music, Linux also has a number of tools to help you rip, mix, manage, and burn your audio—even “scratching” your digital audio is possible.

In this chapter, I highlight some of the best Linux audio applications. There are also hacks in this chapter that take you through the full audio extraction process from ripping CDs to encoding them to tagging the resulting files. Linux has a number of amazing audio tools to automate common tasks, and I've tried to include some of the most useful. At the end of the chapter, I also cover how to access a number of different portable media players under Linux so your music can follow you wherever you go. So put on some music, crank the volume up to 11, and let Linux handle the rest.

HACK
#13**Mix Your Audio for Perfect Sound**

Use the mixer to understand and tweak the different volume levels your sound card provides.

Almost every audio device you encounter has some way to control the volume, whether through a volume knob on a radio, the remote control on a television, or the series of controls on a sophisticated sound board. To control the sound on a Linux system, you use a mixer program. Even though the purpose of the mixer—to change the volume—is simple, often the sheer number of options the mixer provides can prove daunting to new users: it’s not as simple as adjusting a single master volume control. Also, Linux has two different systems for sound, OSS and ALSA, so there is additional complexity that is not immediately obvious. In this hack, I cover a few mixer programs and describe the differences among Master, PCM, and the other major volume controls.

You can launch a mixer program a few different ways. If you use the GNOME or KDE desktop environments, the quickest way to access the mixer is to click the speaker icon on the desktop panel. Windows users will find this applet similar to the speaker on the Windows taskbar. A single click on the icon lets you change the master volume (see Figure 2-1); a right-click shows you various options for the mixer applet. For common volume control, you may only need to access the master volume. To access all of the advanced volume controls, double-click on the applet to launch a complete mixer.

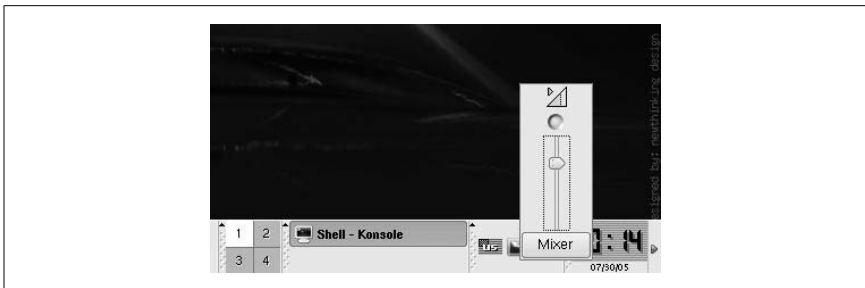


Figure 2-1. Panel Mixer applet

Whether you use OSS or ALSA sound drivers, you will find a base set of volume controls in a mixer application that are based on the default volume controls on the OSS mixer (`/dev/mixer`). OSS is the older of the two sound systems. As such, it is often more stable than ALSA, although a number of newer sound cards are only partially supported in their OSS drivers, if at all. OSS is still present in the latest Linux kernels, but ALSA is the preferred sound system, as it often provides more complete support for newer sound cards—something you will notice in the different options you have in an

ALSA-ready mixer. Whichever sound system you use, when you launch the mixer applet, you will see at least these basic volume controls:

Master

This option controls the master volume for the sound card. The master volume overrides the volume levels for all of the other controls, and muting it or moving it to the lowest level will effectively mute all other sound controls.

PCM

You will likely use this control as often if not more often than the master volume control. The PCM (Pulse Coded Modulation) control sets the volume level for the `/dev/dsp` device on the system, which most audio programs output to. You would use this control if you wanted to change the volume for your media player, for instance. Use PCM instead of the master volume to control the volume for these devices so you don't inadvertently change the volume for the other controls on your sound card. For example, you might want to keep your microphone at the same volume, but increase the volume of your speaker. Changing PCM leaves the mic levels where they were before, but changing the master volume affects both the speaker and the microphone.

Speaker

This controls the volume for the speaker built into the PC. On a Linux system, you might want to use this control to change the volume of the system beep your terminal makes.

Line

The Line works as the master volume for the line-out port(s) on your sound card. Each line-out port also has its own volume control that is numbered starting from one, so a sound card with a single line out would also have a Line1 control.

Mic

The mic control manages the volume of your microphone and will often be muted by default so that any noise it picks up will not interfere with any other sound output on the system.

CD

On systems with CD-ROM drives that have been connected to the sound card with the audio-out cable, this control will manage the volume of the CD player.

PhoneIn/PhoneOut

These controls allow you to change the volume for the phone-in and phone-out ports (often RCA jacks on the back of computer) on your sound card. Fancier sound cards sometimes have these jacks available to connect to other AV equipment.

In addition to the above options, if you use ALSA you will find a number of other controls in an ALSA-capable mixer applet such as *alsamixer*. These controls let you access more advanced features of your sound card, but note that the presence of a control in the mixer doesn't necessarily indicate that the feature actually works! Here are some of the common options you might see:

Master Mono

This option is muted by default, but if it is enabled, it will make all audio monaural instead of stereo and control the volume of that mono-audio source.

Headphone

Present on some laptops, this option allows you to control the volume level on the headphone jack without affecting the volume of the speakers.

3D Control

This option is muted by default, but allows you to enable surround sound if your card supports it.



You do not necessarily have to use an ALSA-compatible mixer if you use ALSA, provided you enable the ALSA OSS compatibility layer, specifically OSS mixer compatibility. With OSS mixer compatibility, OSS mixers will still have access to the basic volume control options, but will not be able to change some of the more advanced options your sound card may provide.

Command-Line Mixer Applications

In addition to whatever mixer program your desktop environment provides, there are some basic command-line mixer programs you can use to access the same volume controls without using a mouse. Two common command-line mixers are *aumix* and *alsamixer*. *aumix* is an OSS mixer with some particularly useful features for scripting, and *alsamixer* is the default command-line mixer program for ALSA.

aumix provides you with access to the standard OSS mixer controls completely from the command line. If you type **aumix** at the command line with no arguments, you will enter its interactive mode, where you can use the up- and down-arrow keys to switch between mixer controls and the left- and right-arrow keys to change the volume of the selected control. Hit the *m* key to toggle mute on a control, and press Tab or Enter to switch between the volume and balance controls for a device.

aumix really becomes useful with command-line arguments. All the different mixer controls are assigned an argument as listed in Table 2-1.

Table 2-1. Common *aumix* command-line arguments

Argument	Function
-v	Main volume
-b	Bass
-c	CD
-i	Igain
-l	Line
-m	Microphone
-o	Line out
-p	PC speaker
-s	Synthesizer
-t	Trebel
-w	PCM
-x	Imix
-1	Line 1
-2	Line 2
-3	Line 3

Each of these arguments can accept a number of options such as `q` to query the current level for the device, and `+` and `-` to increment and decrement the volume by a single level. You can also specify a particular volume level by using a number instead of `+` or `-`, or you can even follow `+` or `-` with a number to control how much to increment or decrement. For example, to set the master volume to 50%, increase PCM by ten and increase the line out by one, type:

```
$ aumix -v 50 -w+10 -o+
```

This functionality is particularly useful to bind *aumix* commands to key sequences so that you can mute your sound with a quick keypress, instead of mousing to a mixer applet. You might also find *aumix* handy if you schedule any audio programs with *cron*. *aumix* solves the problem of making sure that no one changes the volume on the computer between the time you set it and time your script runs. You can run an *aumix* command immediately before your command executes.



HACK #14

Surround Yourself with Sound

Get theater-quality sound out of your multi-speaker setup.

When you've invested a lot of money in your audio and speaker setup, you want to make sure you get the most out of it. Don't settle for basic stereo sound—make your bass go boom boom, your center channel speak, and

your rear speakers provide ambience. Under Windows, configuring this is often quite easy, as the soundcard manufacturer usually provides a program to enable surround sound, but no such software exists for Linux, so you need to go a more manual route. This hack will show you how to enable surround sound, test it, and use it with MPlayer [Hack #48].

For this hack to work, you need a soundcard that supports more than just stereo output and, of course, multiple speakers. The ideal setup, and the one I am targeting in this hack, is what is known as “5.1 surround sound.” This means you have a pair of front speakers, a pair of rear speakers, a center channel speaker, and a subwoofer. As you can see, this is 5 speakers plus a .1 for the subwoofer. (Just to confuse things, some settings will call this a 6 channel configuration.) Here is what you can expect out of each speaker:

Front right and left

These two speakers provide basic stereo output. Most music CDs are designed to output stereo sound, which means these two speakers are all that is required to have an enjoyable music listening experience. Though adequate for listening to a DVD, you’ll miss out on many positional effects (the placement of sounds so that it seems as if they are coming from specific places, both on and off screen). When your system is configured for 5.1 surround sound, these speakers will provide the bulk of your nonspeaking sound.

Subwoofer

The job of the subwoofer is to output low frequency sound—just how low depends upon the quality of your subwoofer. With a subwoofer added to your setup, your other speakers are relieved of the duty of producing low bass sounds. This is good, because those speakers aren’t really designed for that type of output, and they’ll sound better because they will only be playing the audio they can handle well. Bass output is nondirectional, so it doesn’t matter where you place the subwoofer: you should still be able to hear and feel its effects. When the subwoofer is combined with just the front right and left speakers, it is known as 2.1 surround sound.

Center

This speaker mostly outputs the dialogue track from a DVD. Having speech come from a speaker underneath or on top of the monitor “centers” the voices to the images and prevents the odd effect in which dialogue doesn’t seem to be coming from the actors’ mouths, but rather someplace offstage. During loud action sequences with no speech, the center is sometimes used to enhance the overall fullness of the sound to further immerse you in the action.

Rear right and left

These speakers are used for off-stage and ambient sound. For example, if the movie scene has an actor being chased by a car, these speakers output the sounds of the car when it is behind the actor. Or if it is raining outside, these speakers output the rain audio to make you feel as if it is raining all around you instead of just in front of you. Scene music is often sent to these speakers as well, to more fully immerse you in the action. It may seem as if these speakers only have a supporting role in your audio setup, but think about it this way: is a movie itself any good without a supporting cast?

I tested this using nForce4 integrated sound and Logitech X-530 5.1 speaker set. The nForce4 sound is provided by the *Intel8x0* AlsA driver, which is commonly used in many integrated solutions, so the instructions I give here should work without modification on many setups. I'm going to assume you already have basic audio working. (This is typically not a problem with most modern Linux distributions.) You'll also need to follow your speaker installation instructions and make sure you have each of the cables plugged into the correct outputs.

Configuring Your Mixer

[Hack #13] provided a basic introduction to the concept of audio mixers that control the volume levels of your audio outputs. For this hack you are going to use the *alsamixer* command-line mixer, because it is very straightforward to use. You should find each of the options I'll describe here in the KDE (*kmix*) and GNOME (*gnome-alsamixer*) equivalents, but the settings may be in non-obvious places in the program. To get started, run this command inside a terminal:

```
$ alsamixer
```

This opens an ncurses-based mixer program that you can control with your keyboard. The left and right arrows will move you to the different options, the up and down arrows control the volume or settings of that option, pressing M toggles the muting of that channel, and Esc quits and saves your changes. If you already have basic sound working, then your Master and PCM channels are already unmuted and their volume set.

In this program you need to set volume levels for the other speaker channels as well as tell the mixer how you want audio sent to the various speakers.

Using the *alsamixer* options for my nForce4 Intel8x0 based soundcard as a basis, here are the settings you want to configure:

Center and LFE

Unmute and set volumes levels for these two channels. Center obviously controls your center channel speakers, whereas LFE controls your low frequencies, so it determines output levels to your subwoofer.

Surround

There are two options for this. The first sets the volume levels for the rear speakers. The second controls how independently the center and rear speakers work. By default this set to Shared, but for movie watching, you should set it to Independent.

Channels

This option, which you have to scroll to the right to see, tells Alsa how many speakers you have. Set to 6 for 5.1. Most programs I use still need to be explicitly told how many speakers I have, so I've not found this option to be particularly meaningful. Perhaps in the future.

Duplicate front

When this option is unmuted, it duplicates the output sent to the front pair of speakers to the rear. This is useful when listening to stereo music if you want output from all of your speakers, but it is not desirable when movie watching. Keep this muted to ensure the rear speakers receive independent audio signals.

These settings provide the basic foundation for 5.1 audio. These settings should exist for any soundcard that supports surround sound, but I can't guarantee the option labels will remain the same.

Testing Surround Sound

When first playing with surround sound, it's useful to test speaker output using an audio file that sends audio to each speaker in turn. You can obtain such a file from <ftp://ling.lll.hawaii.edu/pub/greg/Surround-SDL-testfiles.tgz>. Download and unpack this file:

```
$ wget ftp://ling.lll.hawaii.edu/pub/greg/Surround-SDL-testfiles.tgz
$ tar -xvzf Surround-SDL-testfiles.tgz
```

In the expanded directory you'll find a few WAV files that have multichannel output. My personal favorite is *chan-id.wav*, which when played speaks the name of each channel in the appropriate speaker—provided, of course, that everything is configured correctly. To test, use this command:

```
$ aplay -Dsurround51 chan-id.wav
```

This command uses the Alsa *aplay* player to send output to each channel in turn. If you don't specify *-Dsurround51*, you will only get stereo output. You

know everything is working correctly if you hear a woman's voice announcing each speaker channel in the appropriate speaker. If this doesn't happen for you, then go back to *alsamixer* and see if you missed anything.

I've noticed a few oddities with this test. For example, when I run it on my Mandriva machine with the previous command, everything works correctly—except my subwoofer delivers a muffled and jumbled “rear rear left right,” and the rear channel speakers then give their identification correctly. The command's text output suggests that I run the command this way:

```
$ aplay -Dplug:surround51 chan-id.wav
```

When I do this, the front and center speakers work as they should, but I get no other output from my other speakers except for the rear right, which states the jumbled “rear rear left right.” None of this turns out to matter. As long as you do get the front and center speakers to announce themselves correctly, and you hear *something* out of the rear speakers, surround sound should work with MPlayer.

Movie Watching

The real reason for surround sound is to create a near theater experience when movie watching. This section covers configuring MPlayer for surround sound. As you may recall from “Use MPlayer” [Hack #48], the *mplayer* command for viewing a DVD is simply:

```
$ mplayer dvd://
```

Go to a scene with dialogue. Pay careful attention to which speaker the dialogue is coming out of. It will most likely be the left and right front speakers. Is any background sound coming from the rear speakers? Probably not. When you launch *mplayer* this way, the center and rear speakers remain silent. You have to tell *mplayer* that you have a multi-speaker setup like this:

```
$ mplayer -channels 6 dvd://
```

Again, listen to a scene with dialogue and listen for background sound from the rear speakers. In my setup, my center channel is still muted, but I hear sound and dialogue from my rear speakers. The problem here is that we haven't told *mplayer* what sound system to use. This is done with a simple addition to the command:

```
$ mplayer -ao alsa -channels 6 dvd://
```

Now, go to the same scenes as before. If everything is configured correctly in *alsamixer*, you should hear most of your sound coming from the front left and right speakers, dialogue coming from the center speaker, ambient sounds from the rear speakers, and, if you're watching explosions, then a nice boom boom from your subwoofer.



While performing your tests, you may find it useful to add the `-ss nn` option to your *mplayer* command. This skips forward the specified number of seconds so you can always start playing at a scene that lets you test both the rear- and center-channel speakers.

Specifying these settings each time you run MPlayer can be annoying. To get around this, add the options to your `~/.mplayer/config` file like so:

```
# Write your default config options here!  
ao=alsa  
channels=6
```

As you can see, each option is identical to what you type at the command line, minus the preceding dash and converting the space to an equal sign.

If you use a GUI frontend to *mplayer*, it will probably provide its own preference option to enable surround sound, so check those settings if the ones in your configuration file don't work.

—David Brickner

HACK
#15

Play Multiple Sounds at the Same Time

Play audio from multiple sources—even if your sound card doesn't support it.

Most Linux users don't think too much about exactly how sound is mixed. On many sound cards, you can send only a single sound at a time. So, for instance, if you are playing an MP3, you won't be able to hear any desktop sounds you have configured, and your instant messenger won't audibly alert you to incoming messages. Although some sound cards can support mixing from multiple sources, many less-expensive cards and those integrated on motherboards don't, so as a result you need to use a sound server. A sound server, in effect, sits between your applications and the sound card, mixes the different sound sources, and presents a single source for output. The default sound server under KDE is *aRts*, and the default under GNOME is *Esound*. This hack covers how to use and configure *aRts* and *Esound*.

aRts (Analog Real-Time Synthesizer) is a system of modules used to synthesize audio. These modules can perform various audio tasks, such as adding audio effects, mixing, filtering, and a number of other things. The *artsd* sound server uses this system to mix audio from multiple sources in real time. Because it sits between the sound card and applications, an application must have *aRts* support to directly take advantage of the *artsd* server (there's a workaround if it doesn't, which I will talk about in just a bit). *aRts* is a core component of KDE (at least for the time being), so it is fully supported there, and a number of other media applications can use *aRts* as well, through the use of plug-ins.

EsoundD (or Enlightened Sound Daemon) is a sound server like *artsd*. It sits between the sound card and programs accessing the sound card so that it can mix sounds and pass them along as output. Because of this, an application needs *EsoundD* support to work with *esd* properly. Since GNOME defaults to using *EsoundD* as its sound server, GNOME applications generally have *EsoundD* support. *EsoundD* has been around for some time, so there's a good chance that recent media applications (again, particularly those for GNOME) support *EsoundD* either directly or through a plug-in.

Since KDE uses aRts by default, if your distribution packages KDE it should package aRts and, specifically, *artsd*. In fact, if KDE is already installed, you already have *artsd* installed (and it is probably running). Likewise, *EsoundD* is packaged by most major distributions, particularly if they include GNOME, and the *EsoundD* package should already be installed if GNOME is installed. You don't have to run KDE to use *artsd* or GNOME to use *esd*. Simply install the *artsd* or *EsoundD* package included with your distribution to use it with your window manager or desktop application.

Generally, *artsd* and *esd* do not start when the computer starts. Instead they are launched by the desktop environment when it starts. If you use KDE or GNOME, this is taken care of for you; otherwise you will need to consult with the documentation for your window manager or desktop environment to learn how to start applications when the window manager is launched.

Configure aRts

aRts can be configured directly from the KDE Control Center, so launch the Control Center from the K Menu or, alternatively, type **kcontrol** in a terminal. Inside the Control Center click Sound & Multimedia → Sound System to access the aRts configuration (see Figure 2-2). The first clue that *artsd* is running is the first checkbox, labeled “Enable the sound system.” If it is checked, *artsd* is running; if not, it isn't. That means if you do need to completely bypass *artsd*, you can disable it here. If you find that sound skips during playback, you might need to check the next option, Run, with the highest possible priority (realtime priority) and, optionally, increase the size of the sound buffer until the skipping stops.

I previously mentioned that *artsd* sits between the sound card and programs, so if you have it running, programs that directly access */dev/dsp* won't work. The next option in the control center lets you automatically suspend *artsd* when it's idle. That way other programs can use the sound card directly once the timeout period has elapsed.

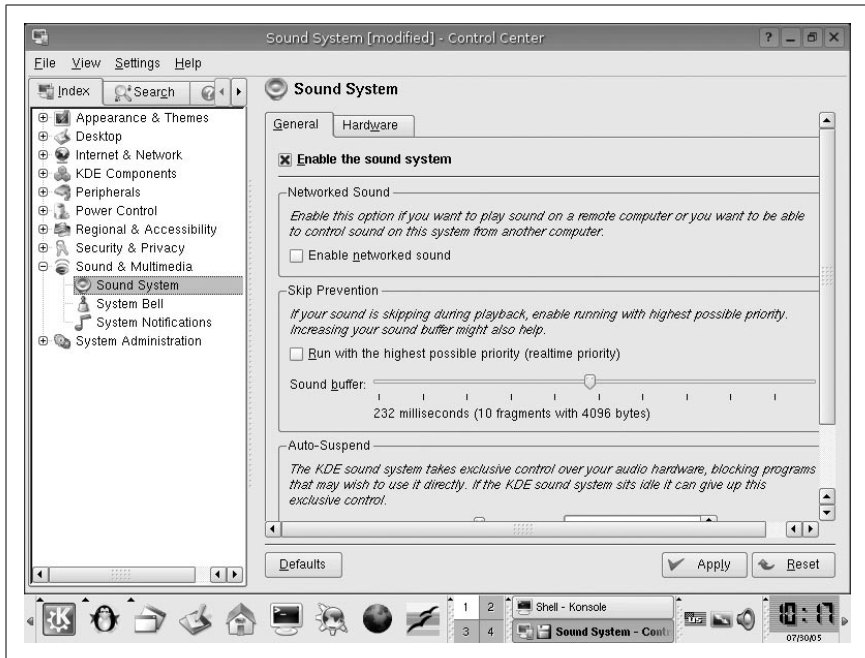


Figure 2-2. KDE Control Center aRts configuration

Configure Esound

There isn't much configuration to be done with Esound: essentially, you just need to be concerned with whether or not GNOME uses it. To check, launch the GNOME Control Center from a menu, or type **gnome-control-center** in a terminal. Then go to the sound section and check whether the sound server checkbox is checked.

Legacy Compatibility

Both aRts and Esound provide another method of compatibility with legacy programs through *artsdsp* and *esddsp*, respectively. Certain programs, such as *Quake*, were programmed to access OSS and know nothing of Linux sound servers. What's more, games like this are no longer being developed, so it's unlikely support is going to be added in. The *artsdsp* and *esddsp* programs basically act as a wrapper around a program and intercept and process any sound card calls. The standard way to use *artsdsp* is:

```
$ artsdsp programname option1 option2 ...
```

Some programs, such as *Quake*, want to do direct memory mapping for sound, which *artsdsp* can emulate with the `--mmap` option. To play *Quake* run:

```
$ artsdsp --mmap quake
```

The *esddsp* program acts in basically the same way, just type *esddsp*, then whatever program you wish to run, followed by its arguments:

```
$ esddsp programname option1 option2 ...
```

artsdsp and *esddsp* don't work for every program. For *artsdsp*, experiment with and without the `--mmap` options to see what provides the best results.



HACK
#16

Get MP3 Libraries for Red Hat–Based Distributions

Enable MP3 support on Red Hat–based distributions with a simple addition to yum.

Software licenses, particularly licenses for multimedia libraries, have always been a touchy point for Linux. There are a number of proprietary media formats out there competing for our eye and ear. Many of these media formats have restrictions in their licensing that require either the user or, more often, the developer of multimedia software to pay a licensing fee to use their library. One such media format is MP3.

Many people consider MP3 to be a “free” format simply because it doesn't have DRM functionality built-in but, in fact, the company Fraunhofer Gesellschaft owns a number of patents on parts of the technology that creates MP3s and requires that MP3 encoders and possibly MP3 decoders pay licensing fees. Most Linux distributions manage the MP3 encoder licensing simply by not including MP3 encoding libraries directly into their distribution. Instead you must download the encoder from a third-party source to limit the distribution's liability ([Hack #24] will provide more information about MP3 encoders under Linux). For the most part, Linux distributions have historically ignored the liability for shipping MP3 decoding libraries—that is, until Red Hat 8. Starting with this version, Red Hat stopped shipping MP3 libraries with the OS (http://www.redhat.com/advice/speaks_80mm.html). As a result, if you use Red Hat or a derivative such as Fedora or CentOS, you will need to go through an additional step to install MP3 support.

To install MP3 support on Red Hat–based systems, you will first need to add a third-party package repository to your package manager. Although there are several repositories that provide the package you need, for this example, I use the Dag software repository (<http://dag.wieers.com/home-made/apt>) for everything.

The first step is to add the Dag GPG key to RPM's list of keys. To do this, open a console, become root, and type the following:

```
# rpm --import http://dag.wieers.com/packages/RPM-GPG-KEY.dag.txt
```

Use Yum

The next step is to add the Dag repository. How you add the Dag repository depends on whether you use *yum* or *up2date* to install packages. Fedora and CentOS both have Yum installed by default for upgrades, while RHEL uses *up2date* by default. If you use Yum for upgrades, you edit */etc/yum.conf*. Below are lines to add to */etc/yum.conf* for Fedora, CentOS, and RHEL (if *yum* is installed):

Add this line if you use Fedora:

```
[dag]
name=Dag RPM Repository for Fedora Core
baseurl=http://apt.sw.be/fedora/$releasever/en/$basearch/dag
gpgcheck=1
enabled=1
```

Add this line if you use RHEL or CentOS:

```
[dag]
name=Dag RPM Repository for Red Hat Enterprise Linux and CentOS
baseurl=http://apt.sw.be/redhat/el$releasever/en/$basearch/dag
gpgcheck=1
enabled=1
```

Add this line for an older version of Red Hat Linux such as RH8.x and RH9.x:

```
[dag]
name=Dag RPM Repository for older Red Hat Linux
baseurl=http://apt.sw.be/redhat/$releasever/en/$basearch/dag
gpgcheck=1
enabled=1
```

After you have edited *yum.conf*, save your changes, update the list of *yum* packages, and then install the *xmms-mp3* package to get MP3 support:

```
# yum update
# yum install xmms-mp3
```

Use up2date

If you use *up2date* for upgrades, then you must edit */etc/sysconfig/rhn/sources* instead. For RHEL4 or CentOS4, add this line:

```
yum dag http://apt.sw.be/redhat/el4/en/$ARCH/dag
```

If you use RHEL3 or CentOS3, replace *e14* in the above line with *e13*. If you use Fedora, add this line to */etc/sysconfig/rhn/sources*:

```
yum dag http://apt.sw.be/fedora/4/en/$ARCH/dag
```

Replace *4* with *3*, *2*, or *1*, if you use Fedora Core 3, Fedora Core 2, or Fedora Core 1, respectively.

Now you can install the *xmms-mp3* package from *up2date*:

```
# up2date xmms-mp3
```


**HACK**
#17**Configure Network Sound**

Set up local sound to play across the network.

In “Mix Your Audio for Perfect Sound” [Hack #13] I discuss how to use EsoundD as a sound server to mix sounds from multiple programs for a sound card that supports only one input. One extra feature of EsoundD is the ability to send and receive sound over the network. This can be particularly useful if you use programs like VNC over a local network, because you can use audio applications remotely and hear them on your local machine. Plus, in general, it’s just pretty cool to be able to send sound around over the network.

To do this *esd* must be installed on the server (the machine on which you hear the sound). You don’t have to have *esd* installed on the client—your client doesn’t even need a sound card—but make sure your programs are EsoundD-compatible. Most distributions should have EsoundD packaged, so install it from your package management tool. If you use GNOME, EsoundD is probably already installed. Once *esd* is installed on the server, you need to start it with special options so that it knows to listen to the network. Kill any open *esd* processes, and then start *esd* from the command line:

```
greenfly@napoleon:~$ esd -promiscuous -tcp -public -port 16001
```

You should hear a series of beeps on your computer to tell you that *esd* has started. Now on your client machine open a terminal and type:

```
greenfly@moses:~$ esdplay --server="ip_of_server:16001" sample.wav
```

Replace *ip_of_server* with the IP address of the *esd* server. The audio file will play back on the remote server, so be sure to adjust the mixer on that machine so that you can hear it. This works fine on an application-by-application basis, but you can also set an environment variable so that any EsoundD-compatible applications you start will output to the remote server. Type the following in a terminal:

```
greenfly@moses:~$ export ESPEAKER="ip_of_server:16001"
```

Now you can start *XMMS*, *mplayer*, *xine*, or any of the EsoundD-capable audio players from this terminal and the sound will be sent over the network. Be sure to add this setting to your *.bashrc* so it’s available each time you start a program.

You can also configure *esd* globally so that it automatically starts with these network options by default. This is particularly useful if you use GNOME on your *esd* server since GNOME is configured to start *esd* automatically. Edit your */etc/esound/esd.conf* file and edit the *default_options* line to say:

```
default_options=-promiscuous -tcp -public -port 16001
```

Save your changes, and the next time *esd* starts, it will automatically start with those options enabled.

HACK
#18

Manage Your Audio with XMMS

XMMS is a classic Linux audio player you can use to easily play a number of different audio file types.

A number of audio players are available under Linux, but XMMS is a classic that you are likely to find available no matter which Linux distribution you use. XMMS has an interface similar to Winamp under Windows and can play most popular audio formats, including MP3, WAV, Ogg Vorbis, and audio CDs, along with many other formats, if you install the proper input plug-in.

The default XMMS interface has controls to play, pause, stop, and skip forward and backward in your playlist—believe me, they’re all there, just very tiny. To open the playlist editor, click the button labeled PL on the interface, right-click on XMMS, and select Playlist Editor, or type Ctrl-E. Within the playlist editor, you can add, delete, arrange, and sort tracks. To add tracks, select the add button and browse to the directory, or simply drag and drop files from your file manager onto the playlist. You can also save playlists you have created so that you can refer to them later. All playlist options that are available via buttons on the bottom of the interface can also be accessed if you right-click on the playlist. The audio settings are adjusted with the equalizer. To display the equalizer, click the EQ button next to the PL button on the interface, right-click, and select Graphical EQ, or type the keyboard shortcut Ctrl-G.



XMMS supports streaming audio. To add a streaming audio source to the playlist, click Add → URL and type in the URL.

Once the playlist has been set up, click the play button or type **x**. The buttons to control playback including previous track, play, pause, stop, and next track can be accessed with **z**, **x**, **c**, **v**, and **b**, respectively. Notice how all of the keys on the keyboard line up in the same order as on the interface. As a track is playing, you can control the volume and balance directly from the main window. To skip ahead in a track, just click and drag the long bar that slowly progresses along the window. The main window also has checkboxes so that you can toggle shuffle and repeat modes for the playlist.

Plug-ins

XMMS is a highly configurable application that supports a number of plug-ins for input, output, visualization, and other options. The most popular plug-ins can be downloaded directly from <http://www.xmms.org> and extracted into your `~/.xmms/Plugins` directory. Right-click XMMS and select Options → Preferences or type Ctrl-P to access additional XMMS options. The preferences window (see Figure 2-3) has tabs organizing different XMMS plug-ins and options:

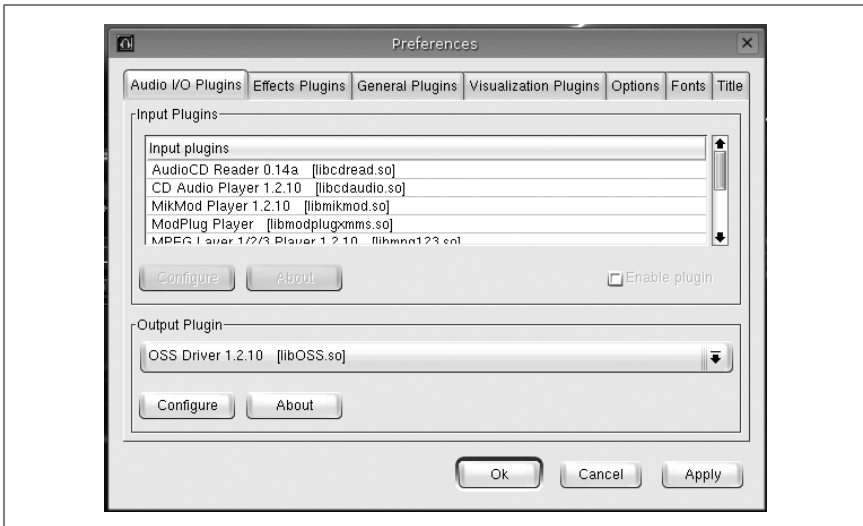


Figure 2-3. XMMS Preferences

Audio I/O Plugins

This tab lets you enable a number of input plug-ins that XMMS supports. You can enable support for different audio file formats including Ogg Vorbis and FLAC, plug-ins that let you play MPEG videos, and a number of other types of input. The Output Plugin section lets you configure how XMMS outputs sound. If your sound card supports OSS (whether natively or through the ALSA OSS compatibility layer), the default OSS driver output plug-in should work fine. Alternatively, if you want to directly use ALSA, choose that output plug-in. If you use Esound to manage multiple audio streams, enable that output plug-in here.

Effects Plugins

This tab lets you enable and configure several effects plug-ins written for XMMS. For instance, you can install a Volume Normalizer plug-in that makes all the sound output stay around the same volume so you don't find yourself raising the volume for a particularly quiet song, just to have your eardrums blasted if the next song is extra loud. There are a number of other useful effects plug-ins, including ones that cross-fade between tracks, provide extra stereo, and create echoes in the music.

General Plugins

This tab is a catch-all location for plug-ins that don't fall into any of the other plug-in categories. Here you will find plug-ins that let you control XMMS with a joystick or remote control (if you have an IR receiver configured; see "Take (Remote) Control" [Hack #79]), along with a number of other miscellaneous features. To configure one of the plug-ins, select it and then click the Configure button.

Visualization Plugins

This tab shows all of the different visualization options XMMS has beyond the standard display in the main window. These plug-ins can be pretty fun to enable, as they provide a lot of colorful visuals to match the music you are listening to. Plug-ins in this category range from simple and three-dimensional spectrum analyzers, to plug-ins that have Tux the Linux mascot disco-dancing to your music (which is a fun thing to run in the background at a party).

Options

This tab lets you configure most of the common XMMS options you might want to change. One particularly interesting option is the Use real-time priority when available. To use this option, you must run XMMS with root privileges, and when enabled, XMMS will run with real-time priority, which gives it priority over other apps seeking CPU time, resulting in fewer skips during heavy CPU load.



Some people find the default look of XMMS too “techno.” You can download new skins directly from XMMS’s official site at <http://www.xmms.org> and save them in `~/.xmms/Skins`. Right-click on XMMS and select Options → Skin Browser to choose one of the skins you downloaded.

XMMS Command-Line Control

One of the particularly useful features of XMMS is that you can control much of it directly from the command line. Table 2-2 lists some of the more useful XMMS command-line arguments.

Table 2-2. Common XMMS command-line arguments

Argument	Function
-r, --rew	Skip backward in the playlist.
-p, --play	Start playing the current playlist.
-u, --pause	Pause the current song.
-s, --stop	Stop the current song.
-t, --play-pause	Toggle between paused and play states.
-f, --fwd	Skip forward in the playlist.
-e, --enqueue	Add the following file(s) to the playlist without clearing the playlist first.

The playlist control options are especially handy if you have a multimedia keyboard, as you can use your desktop environment’s keybinding editor to run the particular XMMS command for that function. Even if you don’t have a multimedia keyboard, you might want to bind a key sequence to

some of these keys. For example, to toggle between pause and play, tell your keybinding program to run `xmms --play-pause`.

The `-e` option is also useful to quickly add tracks to the playlist. To add, for instance, your entire Ramones collection to the playlist, you could type:

```
$ xmms -e ~/mp3/ramones
```

Of course, change the path to the path of the files you wish to add. If you ran that same command without the `-e` option, XMMS would still load those files into the playlist, but it would also clear out the playlist beforehand. When XMMS exits, it remembers the contents of the playlist, and next time you start it, it will reload that playlist unless you tell it otherwise.



Since XMMS can be controlled from the command line, you can also control it remotely or from a script. Simply write a script containing the commands you want to run and schedule it via *cron* or *at*. I've used this feature along with *aumix* to control the volume and create a quick alarm clock. This also means you can control XMMS over *ssh*—a fun way to play a trick on a co-worker or roommate.

HACK
#19

Shuffle Your Music the Smart Way

Use IMMS to weight your music collection based on your listening habits.

Think of all the ways you interact with a computer each day. Any action you take, or even don't take, conveys some meaning. For example, when listening to your music collection, you might sometimes skip songs. What does that mean? There are a number of possibilities. Maybe you do not like the song that was playing, or it does not suit your current mood, or, possibly, you've listened to this song too much and would rather it be played less often.

Is it possible to build a system that uses this information to learn which music you prefer and play it more often? Yes! Intelligent Multimedia Management System (IMMS) is an attempt to create such a system—an adaptive playlist framework that tracks your listening patterns and dynamically adapts to suit your personal music tastes. Plug-ins are currently available for two popular Linux media players: XMMS (<http://www.xmms.org/>) and Beep Media Player (<http://beemp.sourceforge.net/>).

The main feature of IMMS is its complete transparency to the user. It is incredibly unobtrusive—you never have to interact with IMMS directly. Just continue using your player as usual and, over time, IMMS will influence the song selection to cater to your preferences. When your music player chooses the next “random” track in shuffle mode, IMMS weights its choice, based on which songs you've played and which songs you've skipped previously.

IMMS also offers a number of features beyond basic rating. For example, it keeps track of when a song was played last and makes sure the same song does not repeat too often—a common complaint with traditional shuffle-based systems. Moreover, it is able to recognize different versions of the same songs (for example remixes) and treat them as the same song.

IMMS also learns which songs should be played together and which should not. It does so by both watching and learning from the user, and performing acoustic analysis on the actual songs to determine their tempo and spectral “color,” and then attempts to group more similar songs together.

Get IMMS

Depending on the distribution you use, there may already be an IMMS package available to install. There is an up-to-date Debian package for IMMS, as well as a Gentoo ebuild. If you are unable to find a pre-packaged version of IMMS, you can easily build one from source. The tarball can be downloaded from the IMMS homepage at <http://www.luminal.org/wiki/index.php/IMMS>. A list of required dependencies and instructions on building IMMS are included with the source in an *INSTALL* file.

Once you have IMMS installed, you will need to enable it in XMMS. Open the Preferences dialog, go to the General plug-in category, select IMMS, and click the “Enable plugin” checkbox in the lower-right corner.

That’s it. From now on your music player will be imbued with IMMS intelligence. You will notice that music selection will improve over time without your having to do anything at all. Of course, you should not expect this to happen overnight. It will take time for IMMS to learn your preferences. You may find IMMS to be a little slow at first, since it needs to read the tags and calculate checksums of all your files. Depending on the size of your collection, this process should not take very long. In the meantime, type:

```
$ tail -f ~/.imms/imms.log
```

and watch IMMS rate your music.

More IMMS Magic

Although IMMS is built from the ground up to require no direct user interaction, it is possible to do some useful things with the data it accumulates. For example, if you own a portable MP3 player, you may want to fill it with the songs IMMS has identified as your favorites. James C. Jones wrote an excellent program called IMMS Magical Favorites Collector, or IMFavorites for short (<http://imfavorites.sourceforge.net>), for just that purpose. Simply specify how much space your player has, and IMFavorites will create a directory of symbolic links to the gems of your collection (see Figure 2-4).

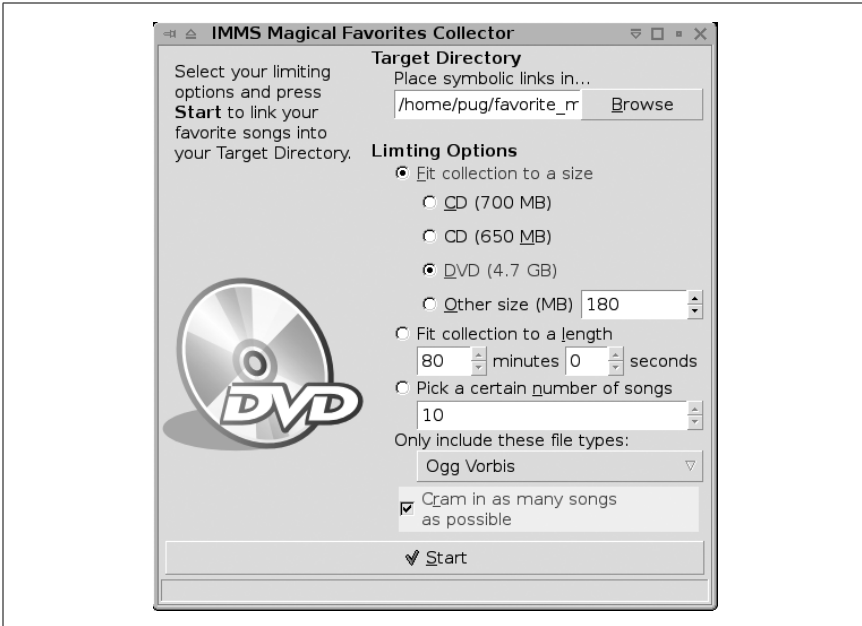


Figure 2-4. IMFavorites in action

A lot of the flexibility of IMMS comes from the fact that it is based around an embedded SQL database — SQLite (<http://www.sqlite.org>). SQLite allows users with even a very basic knowledge of Structured Query Language to generate neat statistics and reports about their music collection and preferences. To harvest its power, you will need to use the command-line client that comes with SQLite 3.0 and above—*sqlite3*. IMMS stores all of its data in *.imms* directory in your home directory, and the database is located in *~/imms/imms2.db*. So to run an SQL query on it, you need to execute:

```
$ sqlite3 ~/.imms/imms2.db '<your query>'
```

Here are some queries you can use:

```
SELECT path FROM Identify NATURAL INNER JOIN Rating WHERE rating > 145 ;
```

This query will effectively generate a list of favorites, because ratings in IMMS range from 75 (really bad) to 150 (really good).

A more involved example that allows you to list the top ten artists for last month would look like this:

```
SELECT SUM( delta ) as sumdelta, readable FROM Journal
NATURAL INNER JOIN Library
NATURAL INNER JOIN Info
NATURAL INNER JOIN Artists
GROUP BY Artist ORDER BY sumdelta DESC LIMIT 10;
```



To familiarize yourself with IMMS schema run:

```
SELECT FROM sqlite master;
```

This lists the tables present in the database and their attributes. Also refer to the description of query language understood by SQLite at <http://www.hwaci.com/sw/sqlite/lang.html>.

—Michael “mag” Grigoriev

HACK
#20

Try Rhythmbox

Use Rhythmbox to manage your audio collection much like the iTunes audio player.

There are many different approaches to audio players under Linux. Some players focus on minimalism, others focus on features, and some aim to provide similar functionality to a media player on another platform. Rhythmbox falls into the latter category, as it was created to provide an iTunes-like interface for audio under Linux.

Rhythmbox is a popular package, and most major distributions package it. If your distribution doesn't, you can download the source from the official project page at <http://www.rhythmbox.org> and compile and install it yourself.

To start Rhythmbox, launch it from your application menu or type **rhythmbox** in a console. The first time you run it, Rhythmbox will take you through an initial configuration wizard so it can find your music files. This step is optional, but I recommend you go through the full wizard so that Rhythmbox can find your files.

After you are finished with the wizard, Rhythmbox will begin scanning your entire MP3 collection (or other media files) for its internal database. Instead of organizing your collection by directory path, Rhythmbox organizes your collection based on the ID3 tags it finds in your MP3s or Oggs, so you can quickly build collections or find songs based on the artist or album. The initial scanning process can be a bit time consuming (ten to thirty minutes, depending on your machine and the size of your collection), and you can watch the collection grow in the main window as Rhythmbox finds new music.

Unlike music players such as XMMS, Rhythmbox has a large main window but displays only a basic set of controls. Most of the window is taken up with playlist and collection information (see Figure 2-5). The left panel displays music options, such as your complete library of music, playlists you created, streaming radio, and so forth. The right side of the main window is

split into a browser and a track list. The browser provides filters that apply to the track list to make it easy to filter and find tracks based on attributes such as artist and album (or genre, in the case of streaming audio).

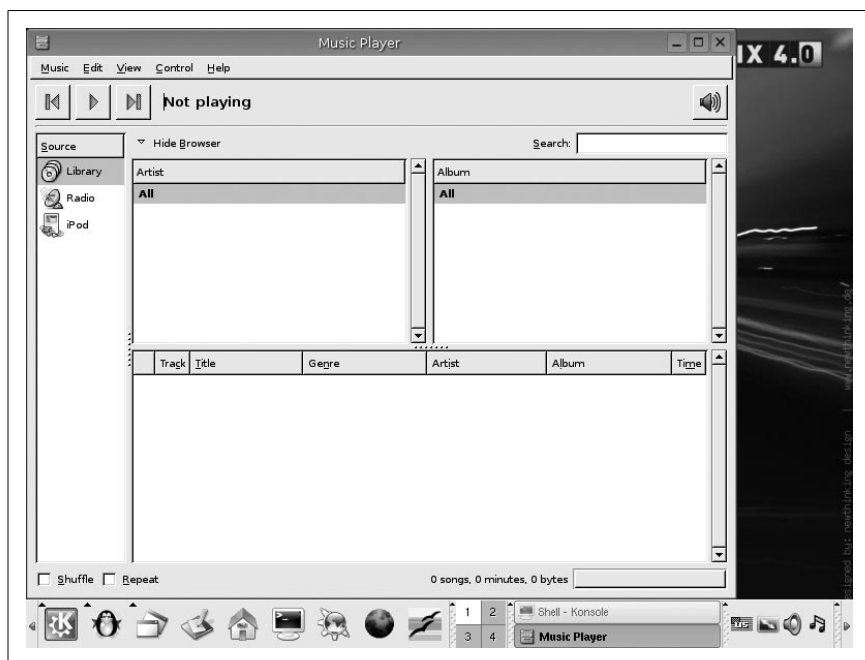


Figure 2-5. Main Rhythmbox window

A Hide Browser/Show Browser toggle above the track list lets you toggle whether the browser is visible. The track list shows the list of tracks for the current level of filtering. By default it displays all tracks for the current source, and if you execute a search or filter using the browser, the track list changes based on those criteria. When a source is changed, the track list changes based on that source, so music from your library displays various music tracks you have added while streaming audio sources display a list of streaming audio servers.

Click the Play button at the top of the interface or Control → Play to start the playback of a track. The navigation interface for Rhythmbox is very simple, with just the controls you need to play or change tracks displayed along with information about the current song. There are also toggles on the interface to change the shuffle and repeat modes for playback.

Rhythmbox can also display playlists in the source area. To create a new playlist, click Music → Playlist and select either New Playlist to create a new

Let amarok Rock Your Music Collection

playlist location, or “Load from file” to load a playlist that you have saved to a file. You can also click Music → Playlist → New Automatic Playlist to add “Automatic Playlists.” The dialog box that appears will let you create a new playlist based on criteria you specify, such as a particular title, artist, album, genre, or rating. You can choose songs that match or don’t match a particular criteria and click Add to add them to the automatic playlist, then click New to create the playlist in the source area.

Rhythmbox can manage a number of audio streams, as well as local files. All the configured audio streams show up under the Radio source. To add a new stream, click Music → New Internet Radio Station and enter the information for your audio stream in the dialog that appears.



Rhythmbox does take up quite a bit of desktop space by default, but you can click View → Small Display to display only the navigation controls. You can also minimize Rhythmbox completely and control it from the music note icon in your notification panel.

HACK
#21

Let amarok Rock Your Music Collection

Use amarok to integrate great music features such as album art, Audioscrobbler, Musicbrainz, song lyrics, and great playlist management into one application.

As more people have started listening to their music collections on their computers, a number of Internet services have appeared to help manage collections and to utilize all this great metadata that exists on MP3s. The services have ranged from CDDB and FreeDB, which provide album and track information based on CD signatures, to services that provide album art and lyrics. A number of the music players under Linux have utilized one or two of these types of services either natively or through the use of plugins, but the amarok music player seems to be one of the best at integrating them all by default into a useful and powerful interface. In this hack, I discuss many of the different features of the amarok player and how to get the most out of them.

Get amarok

amarok is becoming a pretty popular music player, and many distributions already provide packages for it, so you can probably use your standard packaging tool to install it—but if your distribution doesn’t provide it, go to the official page at <http://amarok.kde.org> and download a package for your dis-

tribution. If your distribution doesn't have a package, just grab the source and build it according to the installation instructions.

Once amaroK is installed, launch it from your desktop menu or type **amaroK** into a console. The first time amaroK launches, it will run you through a configuration wizard so that it can find your music files. Point it to your collection and amaroK will scan through each of the files and collect all the metadata for its internal database. Depending on the size of your collection, this may take some time; however, I've found that amaroK is able to scan even a large collection pretty quickly.



amaroK will update information about your collection automatically, but if you have made a large number of changes and want to force a complete rescan of your collection, click Tools → Rescan Collection. You can also click Tools → First Run Wizard to run through the initial configuration wizard again.

As part of the initial configuration wizard, amaroK will ask you to choose the default display style for the application. The *xmms* style splits amaroK into a small navigation window with controls to play, pause, and so on, and a large playlist window with the rest of the controls. The *juk* style combines everything into a single large window (it's named after another KDE music management program with a similar interface). Which style you choose is a matter of taste; however, I found that when I tried the *xmms* style, I was always opening the other window to glance at context-sensitive information such as album art, so I changed it to a single window interface. For simplicity, I will describe sections of the amaroK window as though it were in the single-window style for the rest of this hack.

amaroK Window Sections

Once you have finished the first run wizard and your music is loaded, you will see the main amaroK window (see Figure 2-6). This window is split into two main sections—a playlist on the right and a series of tabs along the left that open into a sidebar and display a number of different context-sensitive information panels. This sidebar is where the real power of amaroK lies, and I will talk more about it in a minute.

The playlist section is like most other playlist sections for media players and displays title, artist, album, and other metadata that amaroK has scanned from your music. To play a track, select it and click the play button along the bottom or click Actions → Play/Pause. This toolbar along the bottom has controls to play, stop, and skip forward or backward in the playlist, a visual-



Figure 2-6. Main amarok window

ization section, and controls for the volume and to clear the playlist. You can also right-click on a track to access playback and playlist functions.

The right-click window also gives you the ability to edit metadata for the track. Select View/Edit Meta Information from the right-click menu, and you can change artist, title, and other information about the track manually or click Fill-In Tags Using MusicBrainz to take advantage of amarok's MusicBrainz integration. MusicBrainz compares an audio signature from your track with its own database of signatures and provides you with a list of songs it thinks the track might be. In my experience, usually only one (or occasionally two) of the options it returns will be applicable, so it's easy to pick out the track. Select the track and MusicBrainz will fill in all the metadata for you. When you are finished editing the information, click Save & Close to save your changes. [Hack #30] provides more information about tagging your files using MusicBrainz.

Configure amarok

amarok has many different options, from basic features to advanced sound engine controls. Click Settings → Configure amarok to display the configuration window. This window is split into a number of subcategories—General, Appearance, Playback, OSD, Engine, Collection, and Scrobbler. The General category lets you configure miscellaneous settings, including whether to remember the current playlist on exit and whether to show the splash screen at startup. The Appearance category allows you to tweak the font settings and color scheme for amarok to give it a personal look. The

playback category lets you control specific audio effects, including cross-fading and whether to resume playback upon start. The OSD category pops up a sample on-screen display that you can drag around your desktop to set its position. You can also configure the font and color settings for this display as well as what information to show.

The engine display is where configuration can get a bit complicated. amaroK itself doesn't know how to play MP3s, OggS, and so on, and instead uses various audio engines that do. This makes amaroK very flexible in that it can adapt to a number of audio engines to suit its environment. Audio engines include the xine engine, GStreamer, aRts, and others. Which engine you choose depends on your specific environment, but for instance, if you use a KDE desktop you probably use aRts to manage your audio, so choose that engine. If you use GNOME you might want to try the GStreamer plug-in. If either of those engines don't work or seem buggy, try the xine engine, which I've personally found to be more stable than the others. Depending on which engine you choose, this display allows you to configure a number of advanced options for that engine, so tread carefully here.

The collection display lets you tell amaroK where to find your music on the system through a tree view of the filesystem. Click on a directory and check it to add it to amaroK's list. In this display you can also configure whether amaroK uses a local SQLite database or a MySQL database to store its collection database, including listening stats. By default, amaroK uses an embedded SQLite database, but [\[Hack #22\]](#) has information on how to configure and use amaroK with a centralized MySQL database.

Finally, the Scrobbler display will allow you to configure an Audioscrobbler account if you wish to submit your listening data to Audioscrobbler. Audioscrobbler is an Internet service through which subscribers upload their listening habits. Audioscrobbler uses this data to figure out listening preferences and displays suggestions in the sidebar or adds them to your current playlist. Set this option by clicking the small toggle box in the main window next to the volume slider. You'll know you have the right one when you see the tool tip "Append Suggestions on."

The Sidebar

The real usefulness of amaroK comes into play in the sidebar. This sidebar splits different types of context-sensitive information into a series of vertical tabs called Context, Collection, Playlists, Media Device, and Files. Each tab opens into a different sidebar full of information and controls for your music collection. Click on a tab to show that set of information and click a second time to hide the sidebar completely.

The Context Pane. The Context pane is the first tab in the sidebar and is the tab you will likely display most when you are just listening to your music. This pane has three tabs along the top: Home, Current, and Lyrics. Like the name indicates, the Context pane displays context-sensitive information based on the current playback state and the track that is playing. If no track is playing only the Home tab is enabled. This tab displays lots of different information about your listening habits and music collection, including a list of your favorite tracks from data that amaroK automatically collects based on your listening habits. In addition, the Home tab lists the newest tracks you have added to the collection and a list of least-played tracks.

When a track is playing, the Current and Lyrics tabs are enabled. The Current tab provides a lot of useful information about the current track split into a few sections. The first section displays information about the current track, such as the title and album, along with album art if it has been configured. If you haven't yet found album art for this track, click the question mark icon or right-click and select "Fetch from amazon.com" and amaroK will search Amazon.com for appropriate cover art. Cycle through the list of images until you find the correct cover and click Save to set that image as the cover art for this album. Once you set the cover art for one song, all the other songs for that album will display the cover art as well.

Click the Lyrics tab to attempt to retrieve lyrics for the current track and display them in that pane. If amaroK can't find a perfect match, it will list a number of possible matches or even allow you to enter your own search terms to find lyrics.

amaroK also keeps track of your listening habits and will display the first time and the last time that you listened to this particular track in addition to its current rating. amaroK calculates ratings based on your listening habits. The rating algorithm is a bit more complex than this description, but basically songs that you listen to all the way through are rated higher and songs that you skip or only partially play are rated lower, so to give a song a higher rating, listen to it more often and listen to it completely.



If you want to perform general maintenance on album art for your collection, click Tools → Cover Manager. Amazon requests that all album art be deleted within 30 days of download and amaroK performs this chore for you automatically.

If you are connected to the Internet, the next section of this pane will display a list of suggested songs based on the current track using Audioscrobbler. With Audioscrobbler support enabled, amaroK will grab a list of suggested tracks from Audioscrobbler based on the track you are currently

listening to, and display the tracks you happen to have in your collection, sorted by your ranking. Right-click a track and select Append to Playlist to add it to your current playlist. Click Settings → Append Suggestions to append these suggested songs to your current playlist—to quickly create a playlist that matches your current mood.

The next section in this pane displays a list of your favorite tracks by the same artist and underneath that a complete list of your albums by that artist, sorted by date, so that you can easily add them to the current playlist. You can also use this view to easily edit metadata for entire albums. Right-click on one of the album titles in the list and select View/Edit Meta Information, and you will be able to edit information about the album and apply it to all tracks for that album.

The Collections Pane

The Collections pane provides you with a complete list of tracks that amaroK has found in your collection sorted by artist and album. You can enter a keyword at the top of the pane to filter the list. Right-click on an artist, album, or specific track to add it to the playlist, queue it in the current playlist, create a new playlist, or edit meta information.

The Playlist Pane

The Playlist pane provides an interface for all of the playlists that you have created or that amaroK has found in your collection. You can add, delete, or edit existing playlists from the top part of the pane. The bottom section of the pane is titled “Smart Playlists” and provides a number of virtual playlists that amaroK has created based on your collection. These virtual playlists include all songs in the collection, your favorite tracks (based on amaroK’s rating system), your most played tracks, and even a set of fifty random tracks.

Click Create Smart-Playlist to open a window where you can create your own rules-based playlist. For example, using this feature, you can tell amaroK to create a playlist of all Rock genre songs released between the years 1982 and 1988 that you have listened to less than five times in the past two months. The process is simple; just create one or more rules that specify your criteria and save. A rule is composed of three pieces: a database field to act upon which is chosen from the drop-down list; an operand that states how to match your criteria against the data; and finally a criteria string. The + button at the end of the default single rule line lets you add additional rules.

The Media Device Pane

If you use an iPod, the Media Device pane will let you drop files into a queue to transfer to your iPod. Click the Transfer button at the bottom and amaroK will initiate the transfer process.

The Files Pane

The Files pane is a general file system browser that lets you find tracks or playlists on your system and add them to amaroK. You can also use a keyword filter to display only files that match your keyword.



HACK #22

Store amaroK Data in MySQL

amaroK has the option to store its collection data in a remote MySQL database. Use this feature to create a centralized database of your listening habits so they follow you from computer to computer.

By default, amaroK stores information about your collection including listening habits and rankings into a SQLite database file on your filesystem. If you only use amaroK on a single computer this works fine, but if you run it on multiple computers you might want your information to be persistent across machines. amaroK supports the use of a MySQL database for this type of storage, but it does require a bit of setup beforehand to use. This hack will describe the steps necessary to migrate to a MySQL collection database.

First, you need root access to a MySQL database to create a new database for amaroK. If you plan on accessing this database from anywhere, you will want it to be on an Internet-facing machine that's always on. Log in to the database as root and create the initial database:

```
$ mysql -p -u root
mysql> CREATE DATABASE amarokdb;
mysql> USE mysql;
mysql> GRANT select, insert, update, delete, create, drop, create \
temporary tables, index ON amarokdb.* TO amarok@localhost IDENTIFIED \
BY 'password_here';
mysql> FLUSH PRIVILEGES;
```

Replace *password_here* with the password you want to use for the amaroK user. Now that the database is created, click Settings → Configure amaroK and then click Collection from the amaroK configuration window. Change the Collection Database Engine to MySQL and then enter the hostname for your MySQL database along with the name of the database (amarokdb in this example) and the username and password (amarok and the password you chose in this example). Click OK. amaroK will now start using the MySQL database to store its settings.

Transfer SQLite Data to MySQL

To import SQLite data into MySQL, the amaroK team created this method, which is not officially supported. First, copy your `~/kde/share/apps/amarok/collection.db` file to your database server, and then type:

```
$ sqlite3 collection.db .dump | \
grep -v "BEGIN TRANSACTION;" | \
grep -v "COMMIT;" | \
perl -ne "s/INSERT INTO \"(.*)\"/
VALUES/INSERT INTO \1 VALUES/; print" | \
mysql -u root -p amaroK
```