

# iPod & iTunes HACKS™

*100 Industrial-Strength Tips & Tools*



O'REILLY®

*Hadley Stern*



HACK

#31

## Build a Cheap Linux Music Server

Build a Linux-based music server to store all your digital songs in a stable and affordable package.

“Build a Cheap MacMP3/AACServer” [Hack #30] covered how to create an awesome Mac-based digital music server. But what if you have a spare PC or a Mac that is too old to run OS X? For the old Mac, check out “Run iTunes on an Old Mac” [Hack #29]. Otherwise, your best option is to look at creating a Linux music server for your iTunes music collection.

That’s right, with an old machine you have lying around, and the best open source operating system out there, you too can easily build a jammin’ music server. To make your jukebox complete, you can route your music to your stereo with AirPort Express and AirTunes.

### Hardware

If you need to go out and buy a computer, I recommend a cheap PC. It doesn’t need to be the latest and greatest, because Linux can perform very well on modest hardware.

Here are the PC music server specifications:

- A Pentium II or AMD K6-III processor and 128 MB of RAM (so, a system manufactured in around 1999) should be fine. More speed and memory will hurt nothing but your pocket.
- A CD-ROM drive to run the Linux installer CDs.
- A monitor, at least temporarily, for server installation and configuration.
- A network card.
- A large hard disk.



My friend just bought a fast 120-GB hard drive at Best Buy for \$100. By the time you read this, there will no doubt be much larger drives available at that price.

For networking, assume a Linksys, D-Link, or similar router that does automatic DHCP IP assignments. For Linux, Debian is a great choice. Debian takes care of package management; you need only run the update command. How do you acquire this gem? At <http://www.debian.org>, you can choose to order the CDs or download and burn them yourself.

If you have an old Macintosh desktop that can’t handle Mac OS X (even with XPostFacto [Hack #23]), you can give Linux for PowerPC a shot. The distribution to get right now is Yellow Dog Linux (<http://www.yellowdoglinux.com>). You can download or order the CDs from the Yellow Dog folks. Make

sure you install the *apt* package with Yellow Dog so you can follow the configuration instructions for Samba or NFS, later in this hack.

## Getting Linux to Talk to iTunes

There are many ways to go about setting up an iTunes remote library configuration. I suggest a universal method using Samba, which is best for novices. You could also set up Rendezvous on your music server, but in my opinion, the way iTunes 4 does Rendezvous library sharing is not ideal. By using the Samba file-sharing method, you keep the power to rate your songs as well as change the genre and other (ID3) song information.

Before we get started, there's one more bit to twiddle. By default, the "Copy files to iTunes Music folder when adding to library" option in iTunes is selected. Turn off this option before adding songs from your new hard drives to iTunes, as shown in [Figure 2-13](#).



Figure 2-13. Adding songs from your new hard drives to iTunes

During the Linux installation, make sure you allocate a hefty partition on your hard drive for the music files. For help with Debian GNU/Linux installation, see <http://www.aboutdebian.com> and, of course, Google.

Here are a few tips:

- Once networking is working on your Linux installation, use Terminal in Mac OS X or download Putty for Windows so that you can SSH into your server from your other computer.
- Some knowledge of Unix commands is helpful.
- nano is a good basic text editor.
- Use the `df -h` command and note the “Mounted on” area where your music should go. I recommend using `/Music` for the mount point.

Once you’ve completed this hack, you’ll have an easily expandable place to keep your music. If you add another hard drive to your server, set it up using Google and the Debian manual. Then, repeat these steps to serve another share.

## Server-Side File Sharing

If your main machine is a Mac, then on your Linux box you can use either Samba or NFS to share your files (Windows users do not have the NFS option). Though the NFS option is faster, Samba is easier to implement.

**Option one: Samba.** Samba is your best bet to get off to a quick start. Its performance is not likely to be as fast as NFS’s for large file copies between your computer and server, but it is good enough. Security is handled with user-based control, where you set up user accounts and passwords. It uses the same user information as Linux.

To install Samba, simply run `sudo apt-get install samba`. Now, you must configure the shares. You probably already created a user during the Linux installation, but you can add more users with the command `/usr/sbin/adduser` (if it’s not installed, run `apt-get install adduser`) and set passwords with the `passwd` command.

To set up shares, Samba configuration is done in the `/etc/samba/smb.conf` file. To improve performance, I recommend adding this line after the `[global]` section:

```
[global]
read size = 32768
```

To set up your share, put the name of your share in brackets, followed by the share information:

```
[music]
comment = Josh's Music
writable = yes
path = /music
public = yes
valid users = josh, dan
```

## Build a Cheap Linux Music Server

```
admin users = josh
```

In my example, I have Unix accounts on my Linux system named *josh* and *dan*.

Now, it's time to create the connection to your Samba server. On your server, run `/sbin/ifconfig` to get its IP address.

If you're using a Mac, go to the Finder and select Go → Connect to Server. Then, enter `smb://` followed by the IP address you just found. You can then log in and choose your music share. Add it as a favorite for quick access in the future.

If you're using a PC, go to the Start menu and select Run. Enter `\\ip_address\music`, replacing *ip\_address* with your server's IP address.

**Option 2: NFS.** I have found NFS on Debian GNU/Linux to perform significantly faster than Samba, but it comes with some catches. The security isn't user-based, but rather IP-based. I actually use NFS myself, but I suggest Samba to the general population, because it is easier. To install NFS, run `apt-get install nfs-common nfs-user-server`. To add your share, edit the `/etc/exports` file. My exports file is lax and allows anyone connected to my network full access to my music. Here is an example full-access `/etc/exports` file:

```
/music 192.168.1.0/255.255.255.0(rw,insecure)
```

You might want to identify your iTunes-running machine specifically for more security. To do this, just enter its IP address. Here is an example restrictive `/etc/exports` file:

```
/music 192.168.1.2(rw,insecure)
```

Now, it's time to create the connection to your NFS server. On your server, run `/sbin/ifconfig` to get its IP address.

If you're on a Mac, go to the Finder and select Go → Connect to Server. Enter `nfs://ip_address/music` (replacing *ip\_address* with your server's IP address), and add it as a favorite.



Having problems reading, deleting, or copying files? I am assuming your server is to be used only by you or trusted people, so you can log into your server and run `chmod -R 777 /music`. This lets any user read, delete, and add to your music library and guarantees that Samba and NFS can do it as well.

## Managing Your Library

If you are using a Mac, you can use either Samba or NFS as your server. For library management, use Doug Adams's iTunes Library Manager AppleScript

(<http://www.malcolmadams.com/itunes/itinfo/ituneslibrarymanager.shtml>). It is shareware, free for up to two libraries—one for at home and one for on-the-go!

If you are using Windows, you must use Samba for your server. For library management, you might be in a tough spot, because I do not currently know of an available VBScript to manage iTunes libraries. One option is to use iTunes at home and use Winamp for your laptop's smaller collection of music when you're on the go.

Congratulations! Using off-the-shelf components and free software, you have set up a system to feed your music files to iTunes through file sharing over a network. Rock on!

—*Joshua Benjamin*