

How File Sharing Works

by [Marshall Brain](#)

At its peak, [Napster](#) was perhaps the most popular Web site ever created. In less than a year, it went from zero to 60 million visitors per month. Then it was shut down by a court order because of [copyright](#) violations.

Napster became so popular so quickly because it offered a unique product -- free music that you could obtain nearly effortlessly from a gigantic database. You no longer had to go to the music store to get music. You no longer had to pay for it. You no longer had to worry about cuing up a [CD](#) and finding a cassette to record it onto. And nearly every song in the universe was available.

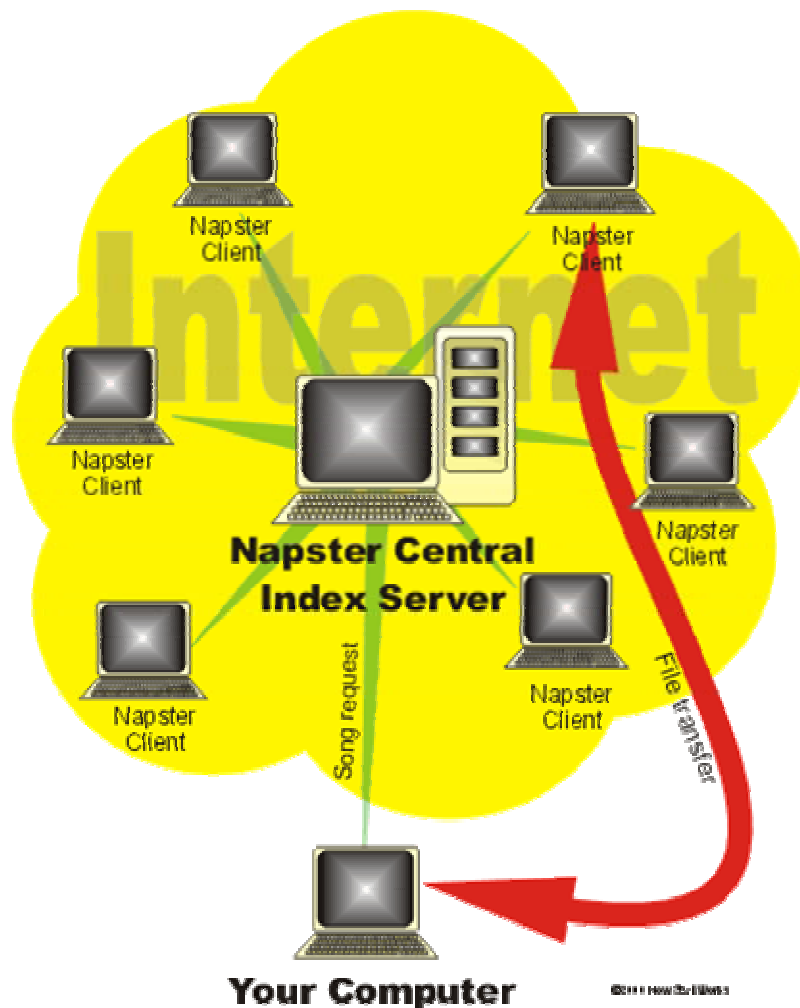
Given that it was distributing an illegal product, Napster's key weakness lay in its architecture -- the way that the creators designed the system. When the courts decided that Napster was promoting **copyright infringement**, it was very easy for a court order to shut the site down.

The fact that Napster promoted copyright violations did not matter to its users. Most of them have turned to a new file sharing architecture known as **Gnutella**. In this article, you will learn about the differences between Gnutella and Napster that allow Gnutella to survive today despite a hostile legal environment.

Napster's Architecture

On the Web as it is normally implemented, there are [Web servers](#) that hold information and process requests for that information (see [How Web Servers Work](#) for details). Web browsers allow individual users to connect to the servers and view the information. Big sites with lots of traffic may have to buy and support hundreds of machines to support all of the requests from users.

Napster pioneered the concept of **peer-to-peer file sharing**. With Napster, individual people stored files that they wanted to share (typically [MP3 music files](#)) on their [hard disks](#) and shared them directly with other people. Users ran a piece of Napster software that made this sharing possible. Each user machine became a mini server.



If you logged into Napster to download a song, here's what happened:

1. You started the Napster software on your machine. Your machine became a small server able to make files available to other Napster users.
2. Your machine connected to Napster's central servers. It told the central servers which files were available on your machine. So the Napster central servers had a complete list of every shared song available on every hard disk connected to Napster at that time.
3. You typed in a query for a song. Let's say you were looking for the song "Roxanne" by The Police. Napster's central servers listed all of the machines storing that song.
4. You picked a version of the song from the list.
5. Your machine connected to the user's machine that had that song, and downloaded the song directly from that machine.

The creator of Napster had a couple of reasons for this approach:

- Napster eventually grew to have billions of songs available. There is no way a central server could have had enough **disk space** to hold all the songs, or enough **bandwidth** to handle all the requests.
- Napster was trying to take advantage of a **loophole in copyright law** that allows friends to share music with friends. The legal concept behind Napster was, "All of these people are sharing the songs on their hard disks with their friends." The courts did not agree with that logic, but it gave Napster enough time to prove the concept and grow to massive size.

This approach worked great and made fantastic use of the Internet's architecture. By spreading the load for file downloading across millions of machines, Napster accomplished what would have been impossible any other way.

The **central database for song titles** was Napster's Achilles' heel. When the court ordered Napster to stop the music, the absence of a central database killed the entire Napster network.

With Napster gone, what you had at that point was something like 100 million people around the world hungry to share more and more files. It was only a matter of time before another system came along to fill the gap.

Gnutella's Architecture

Currently, the most popular system for sharing files is another peer-to-peer network called [Gnutella](#), or the Gnutella network. There are two main similarities between Gnutella and Napster:

- Users place the files they want to share on their hard disks and make them available to everyone else for downloading in **peer-to-peer** fashion.
- Users run a piece of **Gnutella software** to connect to the Gnutella network.

There are also two big differences between Gnutella and Napster:

- There is **no central database** that knows all of the files available on the Gnutella network. Instead, all of the machines on the network tell each other about available files using a distributed query approach.
- There are **many different client applications** available to access the Gnutella network.

Because of both of these features, it would be difficult for a simple court order to shut Gnutella down. The court would have to find a way to block all Gnutella network traffic at the ISP and the backbone levels of the Internet to stop people from sharing.

Gnutella Clients

Napster had one piece of "client software" -- the software that users run on their machines to access the Napster servers. Gnutella has dozens of clients available. Some of the popular Gnutella clients include:

- [BearShare](#)
- [Gnucleus](#)
- [LimeWire](#)
- [Morpheus](#)
- [WinMX](#)
- [XoloX](#)

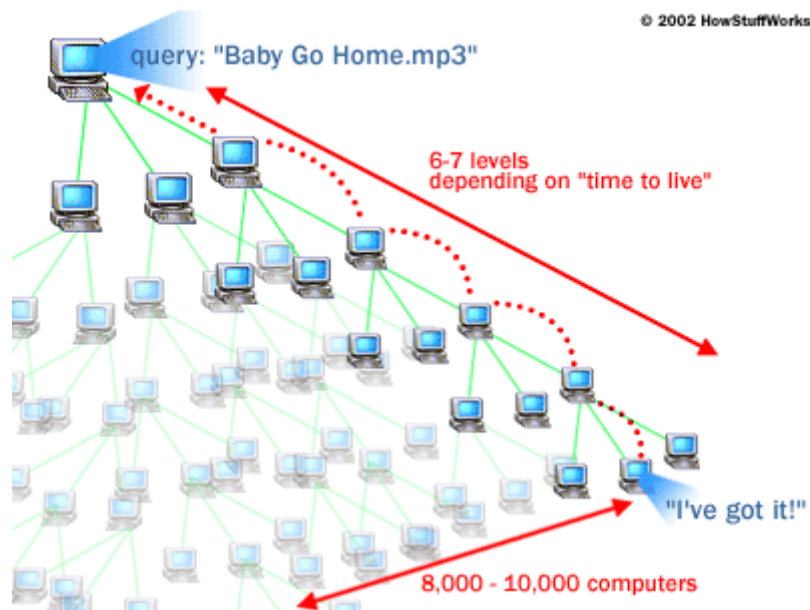
How a Gnutella client finds a song

Given that there is no central server to store the names and locations of all the available files, how does the Gnutella software on your machine find a song on someone else's machine? The process goes something like this:

- You type in the name of the song or file you want to find.
- Your machine knows of at least one other Gnutella machine somewhere on the network. It knows this because you've told it the location of the machine by typing in the [IP address](#), or because the software has an IP address for a Gnutella host pre-programmed in. Your machine sends the song name you typed in to the Gnutella machine(s) it knows about.
- These machines search to see if the requested file is on the local hard disk. If so, they

- send back the file name (and machine IP address) to the requester.
- At the same time, all of these machines send out the same request to the machines they are connected to, and the process repeats.
- A request has a **TTL** (time to live) limit placed on it. A request might go out six or seven levels deep before it stops propagating. If each machine on the Gnutella network knows of just four others, that means that your request might reach 8,000 or so other machines on the Gnutella network if it propagates seven levels deep.

It is an extremely simple and clever way of distributing a query to thousands of machines very quickly.



This approach has one big advantage -- **Gnutella works all the time**. As long as you can get to at least one other machine running Gnutella software, you are able to query the network. No court order is going to shut this system down, because there is no one machine that controls everything. However, Gnutella has at least three disadvantages:

- There is no guarantee that the file you want is on any of the 8,000 machines you can reach.
- Queries for files can take some time to get a complete response. It might be a minute or more before all of the responses, seven levels deep, come in.
- Your machine is part of this network. It is answering requests and passing them along, and in the process routing back responses as well. You give up some amount of your bandwidth to handle requests from all the other users.

Apparently, these disadvantages are minor, because people have downloaded hundreds of millions of copies of Gnutella clients.

XoloX Example

[XoloX](#) is a typical, fairly simple program for connecting into the Gnutella network. It does not have some of the bells and whistles of the more sophisticated clients, but it does work, it is a small file to download (only 600 [kilobytes](#) or so), it has no ["spyware"](#) or bundled [pop-up advertising](#) mixed in with it, and it is very easy to install and use. Its simplicity makes it useful to demonstrate how a typical Gnutella client works.



There are three big things you can do with XoloX: search for files, transfer files to your machine and look at your downloaded files. There are three buttons at the top of the XoloX window that let you toggle between these three activities.

The figure above shows a typical screenshot during a search. All you do is type in the name (or keywords) of the file you are looking for. You can also select the file type: audio, video, etc., or "All Types." Your XoloX client sends out a message containing your search string, and over the course of 30 to 60 seconds a search window fills with results from the thousands of other machines that are processing your query.

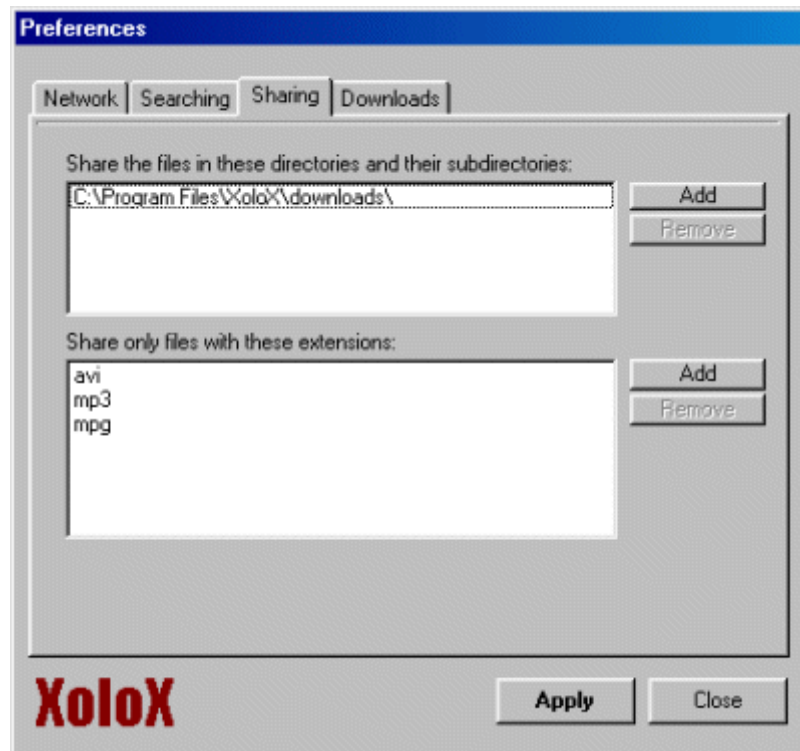
One thing you will notice in the search window is a **score**. The score represents the number of machines currently online that have the same file available. By choosing a file with a high score, you increase your odds of actually getting the file you want.

To download a file, you simply double-click it in the search window. This sends the file name to the Transfer window. Once a filename is in the transfer window, your copy of XoloX will connect to the peer machine to download the file. One nice thing about XoloX/Gnutella is that if multiple machines have the file available, your client can connect to several of them simultaneously to download the file very quickly. In the figure below, you can see that Filename1.avi in particular is taking advantage of this capability to download the file at a rate of 69.2 kilobytes per second. XoloX is estimating 43 minutes to complete the download of over 100 megabytes.

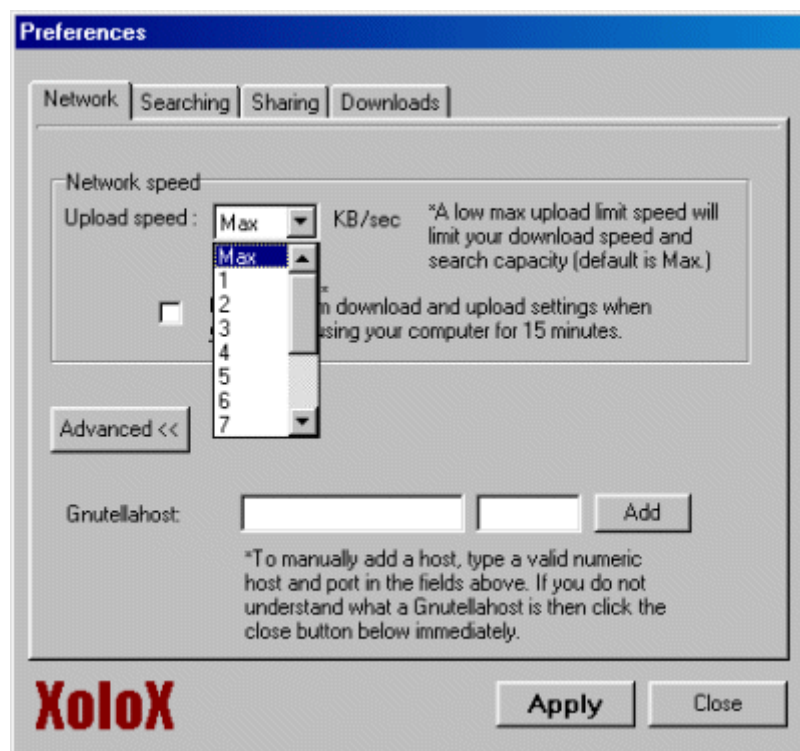
Filename	Filesize	Progress	Est. time	Speed	Download status
Filename1.avi	192,960,512	04%	00:43:29	69.2	Active 4, busy 15, buffer 4
Filename2.avi	175,296,512	16%	04:51:36	8.2	Active 3, busy 15, buffer 1

When you pick a file for downloading, it is fairly common for nothing to happen. That is, XoloX cannot connect to the machine that has the file, or the machine holding the file is already busy helping other people. You can solve this problem either by waiting (eventually a busy machine can get unbusy), by choosing files with high scores (increasing the likelihood of finding an unbusy machine), or by deleting a file that is going nowhere from the transfer window and replacing it with an identical file from the search window.

Once you have the files on your machine, you can find them in a XoloX directory and in the Files window of XoloX. You can share all the files you've downloaded with other people if you like. You do this by first specifying the directories and file types you want to share in the Preferences dialog:



You can also control how much outgoing bandwidth you allow XoloX to consume when people download files from you:



This can keep people from chewing up all your upstream bandwidth.

Is Gnutella Legal?

Gnutella itself is legal. There is no law against sharing **public domain files**. It's when people use

Gnutella to distribute copyrighted music and films that its use becomes illegal. This is the problem that got Napster in trouble. The music industry is officially upset about Gnutella, but there is currently no easy way to control it.

Attacking the Gnutella architecture is one way to disrupt file-sharing activities. There are currently two approaches being used:

1. Overloading the Gnutella network with a flood of bogus search packets.
2. Filling Gnutella servers with corrupted files.

Gnutella's many developers have adapted to problems in the past, so it is probable that new software can work around these threats and keep the files flowing.

The debate at the moment is how much financial damage file-sharing actually causes. Is a shared file a theft, or is it a form of free advertising and exposure just like airtime on the radio is? That's an open question. See the links at the end of the article for some different perspectives.

The Pressplay Alternative

What if you find the idea of copyright infringement uncomfortable and you would like to obtain copyrighted music through a legal mechanism? Sony and Universal teamed up to create a Web site called **pressplay** that distributes music.

Pressplay is not file sharing -- it is a **subscription music service**. You pay a monthly fee between \$15 and \$25 to access the pressplay music library, and you download files from pressplay's central server. The payment options (in July 2002) are:

- Silver: 500 streams - 50 downloads - 10 burns
\$9.95/month for 3 months; \$14.95/month thereafter
- Gold: 750 streams - 75 downloads - 15 burns
\$19.95/month
- Platinum: 1000 streams - 100 downloads - 20 burns
\$24.95/month

There are three different ways to listen:

1. **Stream** - The song comes over the Internet and plays on your computer in real-time. No copy is left on your [hard disk](#). You have to be connected with a decent Internet line to listen.
2. **Download** - The file downloads to your computer and you can play it as many times as you like.
3. **Burn** - You [burn](#) the file onto a [CD](#). It is a normal CD track.

Obviously the first hurdle of pressplay, when you compare it to original Napster or Gnutella, is the **money**. You're either okay with the subscription model or you aren't. Given that you are paying, however, the number of burnable songs seems extremely restricted.

The second hurdle is getting the pressplay **software** installed.

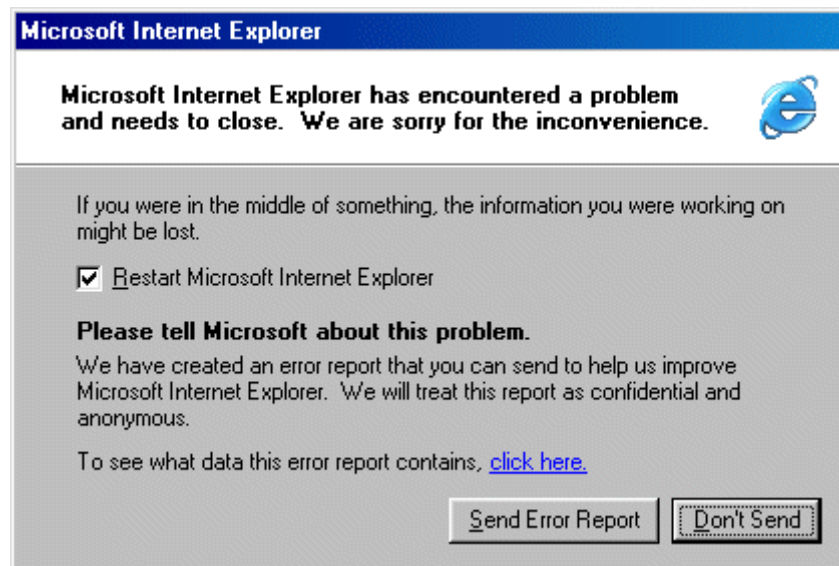
Pressplay Software

The whole pressplay service is wrapped around a custom pressplay application that you download and install on your machine. The application connects with Windows Media Player version 7 or 8. What you'd like to think is that you could go to the pressplay Web site, type in your information and [credit card](#) number, and start listening to music in a few minutes (as you can with XoloX). It's not quite that simple.

You do have to type in your login ID, password, credit card information and so on, but this process is very easy and *pressplay* asks for no more information than is necessary to complete the transaction. That is nice. The **EULA** (end user license agreement) that you have to agree to is quite extensive, and we'll come back to this a moment.

I filled out the two forms, submitted my request, and then clicked a button to start the download process.

Within seconds of pressing the button, this dialog popped up on the screen:



Internet Explorer had crashed. This creates a problem, because the whole download/installation process dies when the browser window disappears. To recover, you open a new browser window and you naturally return to where you were, www.pressplay.com... But the next step is unclear. You don't want to go through the whole process of getting an account *again* -- then you would have two accounts and two bills every month. But there is no obvious place to restart the software download. It took about 10 minutes of poking around to discover this part of a very long [FAQ file](#):

Can I access my *pressplay* service from another location, such as work, home, or even on the road?

Yes, you can access your *pressplay* account from your home or office, or anywhere that you have Internet access. You can play streaming files from anywhere that you have Internet access, and store your downloads on up to two computers.

To access your *pressplay* account from a computer other than the one you signed up on, click on the appropriate affiliate link below to download *pressplay*.

- ***pressplay* on MSN Music** members [click here](#)
- ***pressplay* on Yahoo! Music** members [click here](#)
- **Roxio *pressplay*** members [click here](#)
- ***pressplay* on MP3.com** members [click here](#)
- ***pressplay* on Sony's Musicclub** members [click here](#)
- ***pressplay* connect** members [click here](#)

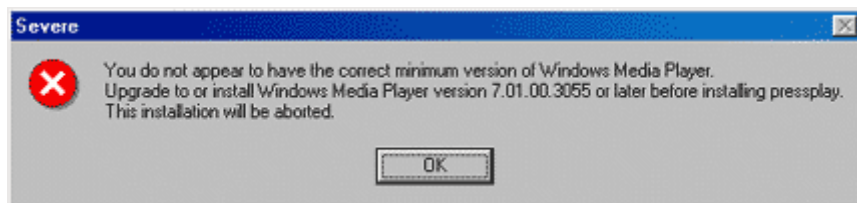
Click on the provided link to download *pressplay*.

Note: you can access your downloads on one additional computer by using the **Sync/Restore** feature available from the **My Account** drop-

down menu.

What's happening here is odd. For some reason, when you sign up, you have to pick a "partner company" from the five or six listed (Yahoo!, MSN, etc.). To download the *pressplay* software, you have to return to the partner company's Web site, not to *pressplay.com*.

Eventually, after a crash or two and some FAQ exploration, you retry downloading the *pressplay* software from the partner's site. Somewhere in the middle of the process, the installation script looks at your system to detect whether you have Windows Media Player version 7. Just the day before I had installed a piece of software that also needed Windows Media Player 7, and I had installed Windows Media Player 7, so I was very surprised to see this dialog:



And that's it. When you click OK, the software installation aborts and there is not a hint of information as to what you should do next.

So... If you know what you are doing, you head to Microsoft.com to download the right version of Windows Media Player version 7. If you don't know what you are doing, I imagine you bail out at this point. Or you call customer service. (I tried that, just to see what would happen. After being on hold for 10 minutes, I hung up.) Or you go paging through the FAQ, which does eventually tell you to go to Microsoft.com.

After downloading Windows Media Player version 7 (a 10-megabyte file), installing it (and going through its EULA), and rebooting, it's back to the *pressplay* partner site to try downloading the software AGAIN, and after installing it and rebooting AGAIN, it actually ran. I was able to stream and download songs, and it seemed to work. The whole process took nearly an hour.

Now, about the EULA for *pressplay*...

Pressplay EULA

Pressplay has embraced the entire "Digital Rights Management" (DRM) process and done a full implementation, enforcing it through Windows Media Player. The goal of DRM is for a media company to have total, absolute control over who can listen to what, when they can listen to it and how the music is accessed. Some examples:

- If you stop paying *pressplay* and unsubscribe to the service, all the songs you have downloaded to your computer will stop working at the end of the month.
- If you try to copy a downloaded song onto a [CD](#), it will not work. The only way to do it is to use one of your precious burns.
- If you try to copy a song onto a [PDA](#) or a player that handles Windows Media files, it will not work
- If you try to [e-mail](#) a song to someone else, it will not work.
- If you download a music file to your [PC](#), they track how many times you listen to it.
- And so on...

It is an amazing document. Compared to Napster and Gnutella, *pressplay* feels incredibly restrictive -- almost Orwellian. Putting a *pressplay* song onto an [MP3 player](#) is nearly out of the question, but one way you could do it would be like this:

- You download it.
- You burn it onto a CD (using up one of your precious burns in the process).
- You rip it back off the CD, converting it to an MP3.
- You then download it to your MP3 player.

And in doing that, you would violate the EULA. And you can only do that if *pressplay* has the song you want, and if *pressplay* has designated the song as burnable. Another way would be to stream the file, connect the speaker port of the computer back into the microphone port, record the song with [SoundForge](#) or something similar while it streams, and save it to an [MP3 file](#) so you can download it to the MP3 player. The quality would not be as good, but it would be acceptable. And that violates the EULA, too.

An interesting question is this: Does the restrictiveness of the EULA bother people so much because we were born and raised in a society where copying songs onto [cassettes](#) was the norm? Or is it because the *pressplay* EULA is really way out there? Should the record labels have the right to say, essentially, "You are paying for the right to listen to this one song on a single device for a limited time, and that's it." Or should it be, "When someone pays for a song, the person should be able to listen to it on a computer, on a stereo, in a car and on a MP3 player, forever." The latter interpretation is what many people have grown up with, so it seems right. There are lots of people, raised on Napster and Gnutella, who believe that it should be, "I should be able to listen to any song any time anywhere for free." And it's pretty clear that this can't work, because there would be no working musicians under that model.

It's interesting to compare *pressplay* and Gnutella because they work on completely different structures and have completely different rules. In the end, it comes down to what you're comfortable with: *pressplay* is totally legal and sanctioned by the recording industry, while Gnutella definitely is not.