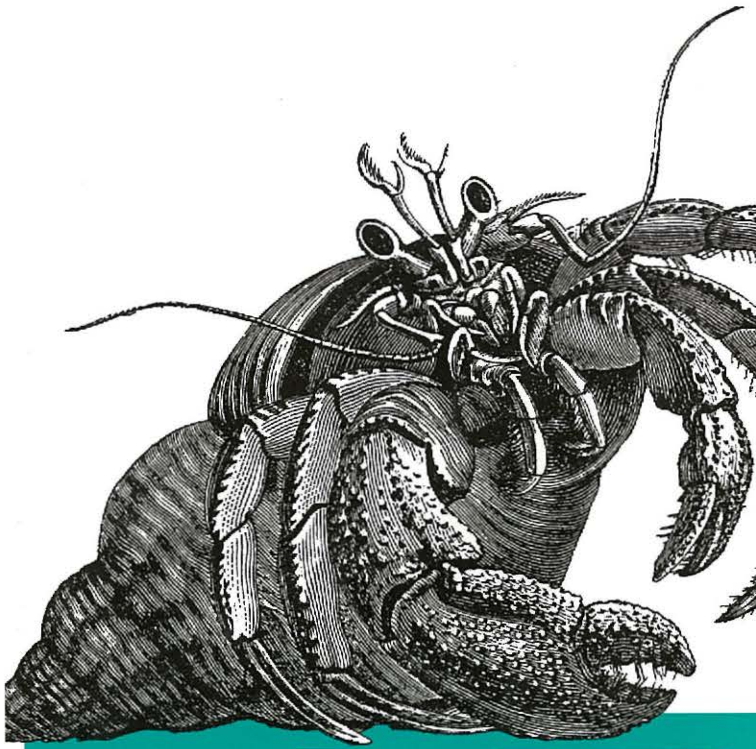
Mastering the MP3 Audio Experience



MP3

The Definitive Guide

O'REILLY®

Scot Hacker

MP3
The Definitive Guide

MP3

The Definitive Guide

Scot Hacker

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Paris · Sebastopol · Taipei · Tokyo

MP3: The Definitive Guide
by Scot Hacker

Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

Editor: Simon Hayes

Production Editor: Maureen Dempsey

Cover Designer: Hanna Dyer

Printing History:

March 2000: First Edition.

Library of Congress Cataloging-in-Publication Data

Hacker, Scot
MP3: the definitive guide/Scot Hacker.—1st ed. p. cm.
ISBN 1-56592-661-7 (alk. paper)
1. MP3. 2.MP3 players. 3. Music—Computer programs. 4. Internet (Computer network)—Computer programs. I. Title.

ML74.4.M6 H33 2000
780'.285'65—dc21 00-025403

ISBN: 1-56592-661-7 [4/00]
[M]

Table of Contents

Preface

This book has a simple premise: People want to build MP3 collections of the music they like and respect. To do justice to that music requires that the MP3 files constituting a personal music collection be of a high audio quality. But MP3 is generally considered to be a convenience format, not an audiophile format—its main advantages are its flexibility and its portability.

While the press generally refers to MP3 audio as being "near CD quality," audiophiles often point to anomalies in the fidelity of the typical MP3 download. But there's a big difference between the average MP3 file downloaded from the Internet and a file you encode yourself, at a decent bitrate, from your own source material, using the encoder you feel yields the highest quality. MP3 is very much capable of achieving CD quality—you just have to pay a little attention to the variables. As I began to research the MP3 scene in earnest, I found that only a small fraction of available resources were paying close attention to MP3 quality issues. As a hobbyist audiophile, I found this dissatisfying, and felt that it was important to provide readers with a "no-compromise" approach to MP3—you *can* have your convenience factors and a quality audio experience at the same time.

While this book provides plenty of introductory material that will coach any reader through the basic mechanics of MPEG audio, it puts quite a bit of emphasis on fidelity issues, in addition to some of the peripheral topics not covered in depth in other books and online resources. Beyond the basics, for example, we'll be taking a close look at the many legal issues surrounding the MP3 scene, the challenges of building your own MP3 playback hardware, the technical details involved in setting up your own MP3 streaming server, and more.

It was also important to me that this book not be overly Windows-centric. Microsoft Windows may be king in terms of both the number of users and the number of MP3 applications available, but I'm not convinced it's the best possible

MP3 playback and creation platform, for reasons we'll go into elsewhere in the book. The number of MacOS users is increasing once again, Linux use is rising at an incredible clip, and BeOS is highly optimized for media content creation and consumption, with lots of built-in MP3-specific goodies. Accordingly, I've tried to balance coverage of non-Windows operating systems evenly throughout this book. Even if you use only one operating system, I hope you'll find reading about some of the alternative approaches illuminating.

It practically goes without saying that the amount and variety of available MP3 playback and creation software is growing at an incredible rate, as are the number of MP3 hardware options available. I don't pretend to have covered everything available in this book, and plenty of new applications and gear not covered here will undoubtedly be available by the time you read this. I've tried to structure the coverage of available products with an eye toward concepts, rather than specifics, so that the provided coverage will (hopefully) be applicable even to products that have yet to be invented. Please regard the coverage in this book, even where application-specific, as a guide to MP3 creation and playback principles in general.

MP3 is a truly amazing codec and a great feat of engineering. In conjunction with the huge array of "peripheral" technologies and tools available, MP3 has single-handedly ushered in a new era of file-based digital music distribution. It is my hope that this book will help you get the most out of the codec and its surrounding technology, so you can get back down to what this is all supposed to be about: enjoying the music you love.

Conventions in This Book

The following typographical conventions are used in this book:

`Constant width`
: Indicates command-line elements, computer output, and code examples.

Italic
: Introduces new terms and URLs, commands, file extensions, filenames, directory or folder names, and UNC pathnames.

Indicates a tip, suggestion, or general note. For example, we'll tell you how to increase performance or save space, or we'll list links to useful web sites.

Indicates a warning or caution. For example, we'll warn you about easy-to-overwrite traps, crucial plug-ins you should not delete, or where it is important to re-encode your material.

How to Contact Us

We have tested and verified the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing to:

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
(800) 998-9938 (in the U.S. or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (FAX)

You can also send us messages electronically. To be put on the mailing list or request a catalog, send email to:

info@oreilly.com

To ask technical questions or comment on the book, send email to:

bookquestions@oreilly.com

We have a web site for the book, where we'll list examples, errata, and any plans for future editions. You can access this page at:

http://www.oreilly.com/catalog/mp3/

For more information about this book and others, see the O'Reilly web site:

http://www.oreilly.com

Acknowledgments

As with any book of this scope, I did not work alone. I am much indebted to my editor, Simon Hayes, for helping to get this project off the ground and for his guidance in structuring and shaping this book in the "big picture." I am also most grateful for the many contributions made by our pool of technical editors:

- mp3tech.org's Gabriel Bouvigne, who possesses a nearly encyclopedic knowledge of MPEG's technical arcana and who helped to flesh out the details of this book in numerous ways.
- Lifelong audiophile Mike Knapp, who can build high-end amplifiers in his sleep and who contributed immeasurably to Hi-Fi issues throughout the book.
- Bruno Prior, who literally built a house around an extensive room-to-room home MP3 network, and who seems to have used every MP3 tool on the planet extensively. Prior also contributed much on the topic of encoding from analog sources.
- MP3.com's "High Geek" Sander van Zoest, who offered much behind the scenes information, especially on the broadcasting and streaming side of things, and who turned me on to MP3 products and technologies before they happened.

In addition, I'd like to thank the members of the WinAmp, mp3stereo, SHOUTcast, and icecast mailing lists, as well as the community inhabiting various MP3 USENET groups. The following individuals have also offered assistance: John Hedtke, Malcolm Humes, Michael James, Henry Kingman, Bruce Lash, Marco Nelissen, Peter Urbanec, Rob Voisey, and Franc Zijderveld.

This book is dedicated to my wife, Amy Kubes, who cheerfully put up with the endless stream of music (both good and bad) flowing from my office over the past year, and for her unwavering support during the course of this project.

In this chapter:
- MP3 Basics
- Rights, Piracy, and Politics

1

The Nuts and Bolts of MP3

In April of 1999, the term "MP3" surpassed "sex" as the most-searched-on term at some of the Internet's top search engines—a phenomenal achievement for a complicated digital music encoding algorithm devised over the course of a decade by a few scientists and audiophiles in an obscure German laboratory.

What is it about MP3 that inspires such unprecedented levels of enthusiasm? For some, it's the prospect of being able to store vast quantities of music on a computer's hard drive, and to shuffle and rearrange tracks from that collection around at a moment's notice. For others, it's the promise of an entirely new model for the music universe—one that allows creative artists to publish their own work without the assistance of the established industry. But for millions of users, the thrill of MP3 is more simple than that: it's the possibility of getting their hands on piles of high-quality music, free of charge.

In this chapter, we'll get a bird's-eye view of the format and the MP3 phenomenon: what it is, how it works, how to download and create MP3 files, and how to listen to them. Then we'll take a look at some of the many issues surrounding MP3, including piracy, politics, digital rights, and the recording industry's stance on the matter. Finally, we'll examine the correlation between the MP3 and open source software movements, and find out why file-based digital music distribution is here to stay.

MP3 Basics

If you're new to the MP3 game, you'll want to know exactly what MP3 files are, where to get them, how they work, and how to make the most of a growing MP3 collection. As you read through this brief overview, keep in mind that these topics are covered in much greater detail elsewhere in this book.

What Is MP3?

Simply put, MP3 is an audio compression technique. Raw audio files—such as those extracted from an audio CD—are very large, consuming around 10 MB of storage space per minute. But MP3 files representing the same audio material may consume only 1 MB of space per minute while still retaining an acceptable level of quality. By drastically reducing the size of digital audio files, it has become feasible for music lovers to transfer songs over the Internet, for users to build enormous digital music collections on their hard drives, to play them back in any order at any time, and to move them around between different types of playback hardware. These possibilities have far-reaching ramifications not just for music lovers, but for artists and the recording industry as well. We'll explore the politics and philosophical issues raised by MP3 in the second part of this chapter.

Why the term "MP3?"

"MP3" is the quick way of referring to an encoding algorithm called "MPEG-1, Layer III," developed primarily by a German technology group called Fraunhofer and Thomson and now officially codified by the International Standards Organization, or ISO. The name, of course, corresponds to the extension found on MP3 files: *After_the_Goldrush.mp3*, for example. More on Fraunhofer and Co. can be found in the section "About the Codec," later in this chapter.

Small is beautiful: How MP3 works

Raw audio does not compress well via traditional techniques: if you try to zip up a WAV file, for instance, you'll find that the resulting archive is only marginally smaller than the uncompressed original.

MP3 takes a different tack on the compression problem. Rather than just seeking out redundancies like zip does, MP3 provides a means of analyzing patterns in an audio stream and comparing them to models of human hearing and perception. Also unlike zip compression, MP3 actually discards huge amounts of information, preserving only the data absolutely necessary to reproduce an intelligible signal. The amount of data preserved is configurable by the person doing the compressing, so an optimal balance between file size and quality can be achieved. The tool or software used to achieve the compression is called an "encoder," while the playback software is called a "decoder" or, more simply, an "MP3 player."

By running uncompressed audio files through an MP3 encoder, files can shrink to around one-tenth of their original size, while still retaining most of their quality. By compressing a little less (to around one-eighth of the original size), MP3 quality can be virtually indistinguishable from that of the original source material. As a result, a three-minute song can be transformed into a 3 MB file, which is something most people can find room for on their hard drives, and that most web surfers can download in a reasonable time frame. In other words, a 640 MB compact disc stuffed full of MP3 files rather than uncompressed audio can store around 10–11 hours worth of music. And since DVDs store around eight times as much as compact discs, a recordable DVD could hold nearly five days worth of continuous music on a single 5" platter.

The mechanics of the MP3 codec and perceptual encoding principles can be found in Chapter 2, *How MP3 Works: Inside the Codec.*

Working with MP3 Files

If you know how to download files from the Internet, have a grasp of basic file management concepts, and aren't afraid to experiment with new applications, you can probably get started on your own MP3 collection without much coaching. However, there are a lot of options and considerations to take into account, including the quality and efficiency of MP3 encoders and players, advanced features and functions, techniques used for organizing and customizing large MP3 collections, and so on. We've dedicated all of Chapter 3, *Getting and Playing MP3 Files*, and Chapter 4, *Playlists, Tags, and Skins: MP3 Options*, to these topics. For now, here's a brief tour of the basics.

Downloading MP3s

In order to start playing MP3 files, you'll need to get your hands on some, of course. There are two ways to do this: You can either download MP3s that other people have created, or you can create them from the music you already own.

Before you start downloading MP3s, you should know that the vast majority of files available out there are distributed illegally. Many people encode music they legally own, and then make it available on the Internet to people who do not own that music, which is illegal (see Chapter 7, *The Not-So-Fine-Print: Legal Bits and Pieces*, for more information). Whether you choose to download pirated music is a moral choice that only you can make. The wide availability of pirated music, however, should not stop you from seeking out legal MP3s. While there are far fewer of these available, you'll be surprised by the quality of the gems you'll find hiding out in the haystacks. A great place to find legal MP3s is *MP3.com*, though that site is certainly not the only source of legitimate files. If you use a commercial MP3 tool like RealJukebox (Chapter 3), you'll probably find a button or link in the interface that will take you directly to an MP3 download site.

Finding MP3 files

While most users start out by simply typing "MP3" into their favorite search engine, that probably isn't the most efficient way of going about things. You might want to start instead at a major site dedicated to indexing or distributing MP3 files, such as *mp3.lycos.com*, *www.listen.com*, *www.scour.net*, or *www.rioport.com*. Search engines can, however, be very useful for finding smaller sites run by individuals—but be prepared to encounter lots of broken links and unresponsive sites. Because many user-run sites are quickly shut down by Internet Service Providers (ISPs) under pressure from record labels, search engines often index links to sites that no longer exist.

The Web isn't the only way to find MP3 files—you'll also find plenty of files on FTP servers, in binary Usenet groups, and in IRC channels. Details on using these venues for MP3 downloading can be found in Chapters 3 and 4.

Users looking to swap MP3 files easily with music fans all over the world may want to check out Napster (*www.napster.com*), which is a sort of combined IRC, FTP, and search client with a twist. Rather than searching the Web, you'll be searching the hard drives of other Napster users for songs you like. Since you'll only see files on the systems of people currently using the service, you won't have to worry about broken links and downed servers. Log in to the Napster server, register your collection with a specific genre, and you'll be able to search for files on other people's systems by song name or artist. Find a song or songs you like and transfer them to your hard drive, while other people do the same with your music collection. Meanwhile, you can chat with other music lovers in the background as your transfer proceeds. Great idea, but the potential for copyright abuse inherent in this product is extreme, and none of the music we found during testing was legitimate. Nevertheless, Napster has single-handedly ushered in a whole new era of user-to-user file sharing, and has the music industry more worried than ever.

Creating your own MP3 files

Creating your own MP3s is only slightly more difficult than downloading them, but the payoff is worth it. You know for a fact that the music in your collection is the music you like, you can personally control the quality of the encodings, and you don't have to worry about whether any of your tracks are illegal.

Encoding tracks from your CD collection is a two-step process. First, bits from an audio CD must be transferred to your system as uncompressed audio, typically as a WAV file. This extraction process is known as *ripping*. The uncompressed audio is then run through an MP3 encoder to create an MP3 file. However, there are dozens of tools available that take care of all the hard work behind the scenes, ripping and encoding transparently in a single step. You'll meet a handful of ripper/encoder combination tools in Chapter 5, *Ripping and Encoding: Creating MP3 Files.*

Playback basics

Think of an MP3 file like any other document you might store on your computer and open in an application. You can open a document by using an application's File → Open menu, by double-clicking a document icon, or by dragging a document onto the application's icon. MP3 files are no different, and can typically be played in any of these ways. There are hundreds of MP3 players available for virtually all operating systems, and all of them are capable of playing all MP3 files. As a user, you have tons of options when it comes to picking your tools. In Chapter 3, you'll meet some of the most popular MP3 players available for Windows, MacOS, Linux, and BeOS, and be introduced to the fundamental principles of MP3 playback.

Playlists

One of the most liberating aspects of working with file-based music (as opposed to music stored on media such as CDs, tapes, or LPs) is the fact that you suddenly gain the ability to organize, randomize, and mix the tunes in your music collection in an infinitude of ways. If you've ever created custom mixed-music cassette tapes, you know how fun—and how time consuming—this can be. MP3 playlists let you enjoy the fun part while skipping the time-consuming part.

The vast majority of MP3 players include a "playlist" window or editor, into which you can drag any random collection of tracks. Any playlist can be saved for posterity, to be played again at a later date. A playlist can be as short as a single song or as long as your entire collection (some people have playlists referencing months of nonrepeating music). A playlist can reference all the music in a folder or an entire directory structure, or can be composed by querying your system for all songs matching a certain criteria. For example, you can create playlists of all country music written prior to 1965, or all of your acid jazz tracks, or all of your schmaltzy disco. Playlist creation and manipulation is covered in detail in Chapter 4.

Playlists are simple text files listing references to the actual locations of MP3 files on your system or on a network. As such, they consume almost no disk space. Because playlists reference songs on your system, it is usually not useful to trade them with other users. There are, however, playlists comprised only of URLs to MP3 files on the Internet, and these will, of course, work on anyone's system.

ID3 tags

MP3 files are capable of storing a certain amount of "meta-data"—extra information about each file—inside the file itself. Data on track title, artist, album, year, genre, and your personal comments on the track can all be stored in an MP3 file's *ID3 tags*. These tags will be inserted automatically by most tools as you rip and encode, or can be added or edited later on, often directly through your MP3 player's interface. ID3 tags become more important as your collection grows, especially when you start using database-oriented MP3 organizers, as described in Chapter 4.

Internet radio

Some people have neither the time nor the inclination to create and manage a huge MP3 collection. Fortunately, they don't have to. Thanks to the rise of outfits like SHOUTcast (*www.shoutcast.com*) and icecast (*www.icecast.org*), thousands of users are *streaming* MP3 audio from their computers to the Internet at large, running live broadcasts much like a radio station. There are several key differences between MP3 downloads and MP3 streaming:

- MP3 broadcasts aren't saved to the listener's hard disk, unlike MP3 downloads. When you tune in to a broadcast, the only thing that's saved to disk is a tiny text file containing some meta-data about the broadcast in question, including the server's address and a playlist. This file is passed to the MP3 player, which in turn receives and handles (buffers) the ongoing broadcast.
- Broadcasts are synchronous, while downloads are asynchronous. In other words, when you tune in to a broadcast, you hear exactly what's being played from a given server at that moment in time, just like the radio. When you download a file, you get to listen to it any time you want.
- Because of bandwidth constraints on most listeners, broadcasts are typically of a lower fidelity than MP3 downloads. MP3 broadcast servers usually send out MP3s that have either been down-sampled to a lower frequency, encoded at a lower-than-normal bitrate, or sent as a mono rather than stereo stream.

Full details on tuning in to MP3 broadcasts can be found in Chapter 4. The process of running your own Internet radio station is described in Chapter 8, *Webcasting and Servers: Internet Distribution.*

Beyond the computer

While you'll almost certainly create all of your MP3 files on your computer, and will most likely begin your MP3 explorations by playing them back through your computer as well, part of the magic of file-based digital audio is the flexibility. There's no reason an MP3 file can't be transferred to any device that includes a storage and playback mechanism. And sure enough, a whole new class of devices has arisen to meet this need: portable units similar to the classic Sony Walkman but geared for MP3 playback, rather than tape or CD, are becoming hugely popular. Meanwhile, we're beginning to see the emergence of a whole new range of home stereo MP3 components, capable of storing gigabytes of digital audio and being operated just like any other home stereo component. Of course, the technology is being applied to car stereos as well. Even hand-held computers such as the Handspring Visor are gaining MP3 playback capabilities.

Users with some technical know-how and a soldering iron are hacking out techniques for building MP3 playback hardware of their own, free from SDMI and other security mechanisms (see Chapter 7 for more about MP3 security and legal issues) that ultimately limit the functionality of commercial MP3 hardware. Chapter 6, *Hardware, Portables, Home Stereos, and Kits*, includes comparative analysis of MP3 portables, an early look at a few MP3 home stereo components, and introduces the concepts of building your own MP3 hardware from scratch.

About the Codec

So, what exactly is MPEG audio compression, and MP3 specifically? Technically, that's a bit of a long story, so we'll go into great detail on that in Chapter 2. You don't need to know how MP3 works in order to start playing with it, but to shed a little light on the subject now, MPEG audio compression is a "psychoacoustic" technique that exploits various limitations in both the human ear and the mind's ability to process certain kinds of sounds at very high resolutions. MPEG encoders store "maps" of human auditory perception in a table, and compare an incoming bitstream to those maps. The person doing the encoding gets to specify how many bits per second will be allocated to storing the final product. Taking note of that restriction, the encoder does its best to strip away as much data as possible (within the specified data storage limitation, or "bitrate") while still retaining the maximum possible audio quality. The more bits per second the user allows, the better described the final output will be, and the larger the resulting file. With fewer bits per second, the user will get a smaller file (better compression), and a corresponding decrease in audio quality. Again, we'll go into the process in greater detail in Chapter 2.

The MPEG family

MPEG is not a single standard, but rather a "family" of standards defined by the Moving Picture Experts Group, which was formed in 1998 to arrive at a single compression format for digital audio and avoid a standards war between various competing technologies. All of the MPEG standards are used for the coding of audio-visual data into compressed formats.

Coding in this sense of the word refers to the process of running a stream of bits through an algorithm, or set of rules. *Encoding* is the process of taking an uncompressed bitstream and running it through the algorithm to generate a compressed bitstream or file. *Decoding* is, naturally, the opposite—taking a compressed bitstream and turning it into an uncompressed file or an audible signal. The term *codec* is short for compressor/decompressor,* and refers to any algorithm capable of performing this bidirectional function.

The MPEG family is broken down into major classes (MPEG-1, MPEG-2, MPEG-4), which are further broken down into sub-classifications called *layers*. Each major class and layer is optimized for specific real-world applications, such as compressed movie soundtracks, broadcast, or file-based musical coding. Each successive layer is more complex than the preceding layer. For example, a layer III decoder will be 2.5 times more complex than a layer I decoder. The MPEG "layers" are described in sub-documents of each class, with audio coding schemes described in a document labeled "ISO/IEC11172-3." The MPEG coding technique that interests us in this book is MPEG-1/MPEG-2 Layer III, referred to throughout this book simply as "MP3."

Technically, MPEG-1 Layer III and MPEG-2 Layer III are both referred to as MP3, as are the rather obscure MPEG 2.5 extensions. MPEG-1 Layer III is used for 32, 44.1, and 48kHz sampling rates, while MPEG-2 Layer III is for 16, 22.05, and 24kHz sampling rates. The MPEG 2.5 extensions allow for 8 and 11kHz. MP3 players can play any of these, and the specs are very similar.† The vast majority of files you'll encounter in the wild are simple MPEG-1 Layer III.

Do not confuse MPEG-1 Layer III (MP3) with MPEG-3—there is no such animal. There was once an MPEG-3 classification in development, which was intended to address high-quality video. However, MPEG-2 was shown to deliver sufficiently high quality, so MPEG-3 was conjoined with the existing MPEG-2 specification. The spec now skips from MPEG-2 to MPEG-4.

* In some circles, the term stands for enCOder/DECoder, though this interpretation has lost favor to compressor/decompressor.

† MPEG-2 also allows for multichannel extensions of up to five channels, though few people have ever actually seen this in action. Multichannel efforts are concentrated on MPEG-4, covered in Chapter 9, *Competing Codecs and Other File Formats*.

The MP3 patent

The fact that the MP3 spec is maintained by the MPEG Working Group doesn't mean they invented the technology. The working group merely codifies standards to guarantee interoperability between various applications, operating systems, and implementations. One of the very first tasks of the working group was to circumscribe the conditions of the ownership of intellectual property under the umbrella of international standards. Their conclusion was that patented technologies are allowed to be codified as standards, but that those patents must be fairly and equitably licensable to all comers, so that no single company could gain a monopoly on a specific audio/video compression technology.

The MP3 codec itself was devised by the Fraunhofer Institute of Germany and Thomson Multimedia SA of France (referred to throughout this book simply as "Fraunhofer"), who originally published the standard in 1993.* Fraunhofer and Co. own the intellectual copyright on any technology capable of creating "an MP3-compliant bitstream." While Fraunhofer publishes low-grade sample code that can be used as a basis for more sophisticated MP3 coding tools, Fraunhofer still requires developers of MP3 encoders to pay hefty licensing fees (full details on that can be found in Chapter 5).

To learn more about the MPEG working group and MPEG specifications in general, there is no better starting point than *www.mpeg.org*. To learn more about Fraunhofer and MP3 licensing issues, see *www.iis.fhg.de*. The official web site of the MPEG Consortium is *drogo.cselt.stet.it/mpeg/*.

Rights, Piracy, and Politics

The flexibility and portability of MP3 has left the recording industry wondering where to turn, unsigned musicians newly empowered, signed artists with mixed reactions, and fans making out like bandits. The debate centers on a quest for the right balance between exploiting the promotional power of this new medium and protecting the intellectual copyright of artists and labels.

MP3's Impact on the Recording Industry

In July of 1999, the International Federation of Phonographic Industries (IFPI) estimated that around three million tracks were downloaded from the Internet every day, most of them without the permission of their copyright holders. The Recording Industry Association of America (RIAA) claims to have lost as much as $10 billion through music piracy in 1998. It's not just record company executives and

* Fraunhofer did not work alone; other companies and organizations (notably AT&T) contributed to the development of the encoder as well.

artists who stand to lose; the digital music revolution has implications for everyone in the channel: record store owners, CD pressing plants, and even truck drivers. Of course, most signed artists resent having their intellectual property illegally distributed as well. Well-known artists ask the RIAA every day to clamp down on pirate sites hosting their music (although it's also the case that many signed artists are much more supportive of MP3 than are their labels). In the rest of this chapter, we'll take a look at some of the many difficult issues currently being faced by the industry and music lovers alike, and take a look at some of the techniques the industry is proposing to deal with the situation. The legal nitty-gritty of MP3 is discussed in more detail in Chapter 7.

File-based digital audio changes the game

For nearly a century, the record industry has held the distribution of musical content in a hammerlock. If you wanted to own music, you had to do it on their terms, purchasing music distributed on the media they had officially blessed, and only through their approved channels. While the industry's stranglehold on music distribution slipped for the first time in the 1950s with the advent of reel-to-reel tape decks, and even more in the '70s with the popularization of the cassette tape, tape technologies had a major Achilles' heel: analog copies always lose a little quality as successive copies are made—a third-generation copy of a well-recorded LP doesn't sound so well-recorded anymore. In addition, the person making the copy is burdened with having to create a new physical instance for each person to whom she wants to distribute her tunes.

I don't have to tell you that MP3 has changed all of that. Digital copies (of anything) are virtually bit-perfect, so no quality is lost in successive generations—a 74th-generation copy sounds every bit as good as the original. And then there's the Internet. Because digital music can be file-based rather than media-based, a single file representing any kind of content can be placed on a web or FTP server and made available to the entire world at once. The burden of making physical copies, which naturally limited the rampancy of tape-based copies to a large extent, has vanished.

But until recently, there's been another "gating factor" that has limited the spread of file-based audio distribution: size. While it's always been possible to rip an audio track from a compact disc and make it available on the Internet, doing so was impractical because uncompressed audio consumes around 10 MB per minute of storage space. Few people had the available bandwidth or storage space to be whipping 30 MB pop songs around.

If all of us had Internet connections with unlimited bandwidth and hard drives large enough to store terabytes or petabytes of data, MP3 would be unnecessary. Limited bandwidth and small hard drives are the only reason MP3 even exists (or, at least, the only reasons it's become popular). Ironically, the MP3 phenomenon hit at a time when both of these issues were being addressed at a rapid clip. More and more, people are having DSL or cable modem connections installed in their homes, and 36 GB hard disks are available for a few hundred bucks at this writing. If the trend continues, and there's no reason to think that it won't, one can almost imagine audio compression in general becoming obsolete due to a lack of demand. But for the foreseeable future, limited bandwidth and modest hard drives are a reality, so digital audio compression is a necessity.

If you can't beat 'em, join 'em

Will the recording industry be able to put an end to MP3? If not, how will it cope with the phenomenon? There are several factors at work here. While people commonly claim that the cassette revolution had little impact on the industry, the truth is that, for whatever reason, sales today aren't as great as they were in the '70s. Of course, there are several reasons for this, such as the fact that CDs cost twice what LPs used to cost, and the fact that we no longer seem to have anything like the giant mega-stars of the '70s. Pink Floyd's "Dark Side of the Moon" stayed in Billboard's Top 100 for the better part of a *decade*. While we still have stars, the days of the Beatles and the Stones are, most likely, gone for good. Of course, the industry feels that illegal MP3 downloads are making these problems worse.

But a 1999 report from industry analyst Jupiter Communications concluded that only three percent of consumers would be purchasing downloaded music by the year 2003. While the industry has good reason to be concerned, some see that last factoid as a wake-up call to the industry—either adapt to a world in which downloadable copyrighted music is a reality, or be left out altogether. If this projection turns out to be true, the industry will either have to find a way to crush unprotected MP3 distribution (unlikely), or accept the fact that MP3s and other digital audio files will continue to be distributed without hope of a significant revenue return.

In the short term, the RIAA launched a campaign to start shutting down pirate sites. For the most part, this has consisted of cease-and-desist letters being sent to Internet service providers, warning them to remove illegal files from users' sites or face prosecution (see Chapters 7 and 8). ISPs are generally quick to oblige (and are bound by law to do so). But even with the best lawyers in the land at their disposal, the industry has found that trying to crush pirate sites in the midst of a phenomenon this large is like playing a vast and endless game of Whack-a-Mole—eliminate one site and six others spring up in its place. Even with the cooperation of ISPs, the task is futile and the industry knows it, though they are still obliged to continue trying.

Beat 'em: The Secure Digital Music Initiative (SDMI). In the face of this apparent futility, the industry has decided to approach the problem from another angle: Chop it off at the knees. Since the sprawling and largely ungovernable Internet cannot be easily controlled, the industry has decided to work with its partners and make it harder to create copy-able digital music files to begin with. By colluding with the makers of compact discs, sound cards, and software vendors, and by embedding special codes into newly created CDs, the industry hopes to get as many people using copy protection-enabled equipment and source material as possible.* According to a quote made by SDMI Executive Chairman Dr. Leonardo Chiariglione in *Billboard* magazine:

> You will be able to play your MP3 files on the portable devices of today, but at a certain point in time, which may happen quite soon, the record companies will start embedding some signals into their future content so that it can become playable only on SDMI devices.

As you'll see throughout this book, it is logically and practically impossible to create a 100% secure system, since savvy users can always trap music as it's heading out of the computer and toward the sound card. The industry's goal, then, is to create a fence high enough that the vast majority of users will lack the technical know-how or wherewithal to try and jump over it. This plan, known as the Strategic Digital Music Initiative (SDMI), is already much-delayed in its implementation at this writing, and is, many feel, doomed for failure. Only time will tell. More on SDMI can be found in Chapters 6 and 7.

Join 'em: MP3 and electronic commerce. If illegal MP3 distribution cannot be stopped or even significantly curtailed, then the industry may learn to embrace the new paradigm and accept the fact that its role is changing in the digital world. Major labels will most likely come to appreciate the "buzz" effect that can be created by posting tracks on their own web sites. Releasing a few good tracks from an upcoming album by a big-name star is likely to result in more sales and more word-of-mouth. Already, we're beginning to see the first glimmerings of this phenomenon, as major stars such as Tom Petty, Billy Idol, The Grateful Dead, and Alanis Morissette embrace the format.

* Embedded codes in music, known as *watermarks*, are created such that their presence persists even when transformed from one format to another, even when "jacked" out of an audio port.

The industry faces a major obstacle in persuading hardware vendors to collude with them in making computers secure against music piracy: Consumers *want* the ability to copy music freely. If one hardware vendor adopts SDMI and another does not, many consumers are simply going to buy equipment from the one who does not. That's quite a disincentive to vendors considering implementing SDMI in hardware, and without global and enforced legislation (unlikely), the industry faces an uphill battle. Nevertheless, Sony has already developed a pair of technologies, called MagicGate (for use in devices) and OpenMG (for use in PCs), that guarantee that a signal can be moved from one place to another, but not copied. Of course, Sony is also a record label, so their desire to move on this front is understandable. But unless they can get other vendors to adopt the same or similar technology, savvy consumers will simply avoid these products.

But the industry doesn't have to *give* away MP3 files. What's wrong with selling them? One can imagine a future in which fans can log into a label's site and download tracks for a buck a pop, selecting only the songs worth purchasing and (happily) ignoring the duds. Whether this is managed via micropayments or ongoing accounts with labels, there's a big problem here: Once a fan has purchased a file, what's to stop him from hacking it into an unprotected version, placing that file on his own site, and making it available for download—enabling infinite copying of the unprotected version to the rest of the world? SDMI and audio formats with built-in security (see Chapter 9) will help here, but again, it only takes one user to break or somehow get around the security mechanism, and the file is once again in the clear and released into the wild. Right or wrong, most people are going to download the free, unprotected version rather than the 99¢ version when given a choice. One begins to appreciate the industry's dilemma.

So, if the industry decides to go for it and start selling music online in a big way, how will they do it? First of all, due to its lack of security, MP3 is very badly suited for the job. Online sales of digital music are likely to come in some other file format, and more likely in several of them. No big deal—the flexibility of software makes it easy for users to store lots of formats on their hard drives, and probably even to play them all back through the same player. Regardless, you can practically rest assured that whatever formats are used will be SDMI-compliant. Taking that as a given, here are the models:

Micropayments

Because credit card charges always incur a transaction fee, low-cost items (such as individual songs) are more awkward to sell online than are sweaters or beer. The notion of micropayments is that users maintain an account with an independent organization. Users may load up the account with money in advance, as with a phone card, and charge small purchases against it. Record labels may also run their own micropayment schemes.

CD distribution

Because the production of compact discs is relatively inexpensive, users may be able to order up a bunch of songs at once and have them pressed to an audio CD, which will then be mailed to the user. Because the cost will be higher, standard credit card transactions will be feasible. This model, in fact, has already been used by companies such as Liquid Audio (*www.liquidaudio.com*), though they have not managed to reach a significant cross-section of the population. Such a model run by a major label would likely have more success.

Subscription

In this model, users would pay a flat monthly or yearly fee and be entitled to download the latest hits from a variety of artists selected by the user or the label, in whichever format the label chooses to work.

Temptation

Perhaps the simplest model, and the one most akin to the shareware model computer users are familiar with, is to simply give away a track or two from upcoming albums, with the hope that users will like the tracks enough to purchase the entire CD, either online or from a local record store.

Advertising

Controversial for good reason, the MP3 advertising model is an extension of the banner ads on web pages with which we're all familiar. Each downloaded MP3 file comes with a brief advertisement embedded into the first few seconds, which is the first thing the user hears. While this is essentially similar to what we get on the radio every day, most users dislike the notion of having their personal MP3 collections riddled with ads. Nevertheless, this model is already in use by Amp3.com, who has managed to sign a number of big-name artists on to the program.

"Guilt sites"

This rather odd idea, which has been tossed around by many people in many forums, has yet to see a working model. The concept is that many users want to enjoy the free trade and exchange of MP3 files, but still want to pay a royalty back to the labels and artists who made the music possible. "Guilt sites" would allow people to check in anonymously and say, "I've decided to actually keep and listen to 24 of the 113 files I downloaded this month. Here's a list of the songs. Please accept my $24 in payment." A fascinating idea, but one which the recording industry is unlikely to bless.

Is MP3 Legal?

One of the questions most frequently asked by people concerned about MP3 piracy is, "Why not just make MP3 illegal?" By making MP3 illegal, they reason, it would be easy to arrest music pirates and deter others from taking part in the ongoing plundering of the traditional music business. There are several problems with this line of thinking:

- MP3 has lots of legitimate uses, and is intended for legitimate use. The mere fact that it can also be used for illegitimate purposes does not in and of itself provide sufficient reason to outlaw it.
- MP3 is a codec. Nothing more, nothing less. Nothing about MP3 is inherently dangerous. You can use a crowbar to break into a drug store, chicken manure to make bombs, or a laptop computer to crack security systems, but that doesn't mean any of those things should be illegal. You can kill yourself or others by drinking too much alcohol, swinging a machete in the wrong vector, or by eating too much bacon, but that doesn't mean those things should be illegal.
- If we were to make MP3 illegal, we would also have to outlaw all other computer file formats, from *.DOC* to *.JPEG*, since all of them can be used to store and distribute intellectual property illegally as well.

Short answer: Yes, MP3 is legal—it's just an innocent codec. Issues surrounding the protection of intellectual and creative copyright are completely separate from the mechanism of distribution. However, MP3 differs from other audio compression techniques; it provides no built-in means to prevent unrestricted copying. This is an issue the industry is struggling to solve, but MP3 itself is but one player in a larger problem, not the problem itself.

The Artist's Turn

Great talent does not automatically end up at the top of the music business. All over the world, millions of talented, creative artists are playing music for their friends, or in small clubs, or even on tour, trying to make a name for themselves. You think that a talented artist will automatically end up with a recording contract? Think again. The record business isn't necessarily looking for talent... but that's another story.

Unless you're scouring the local newspaper and heading off to local clubs night after night looking for something fresh, you may never hear thousands of great musicians and songwriters. Unless an artist is being spotlighted by the recording industry, mass exposure is almost impossible for an artist to get.

The next wave of self-publishing

The MP3 revolution addresses this concern head-on. Because any musician with a computer can encode their own songs to MP3, they can potentially expose themselves to the world at large without ever having to sign a recording contract. They can go directly to the people, bypassing the industry as we know it. This possibility is analogous to the great desktop publishing revolution of the '80s and the web publishing revolution of the 1990s—suddenly, everyone has the ability to publish their own music (no matter how bad or how good) without help from "the man."

Unsurprisingly, this—perhaps the single most liberating and legitimate aspect of the MP3 revolution—is seldom if ever mentioned by record industry executives when talking about piracy problems. An artist signed to a label generally does not have the right to post his own songs to the Internet, because the label owns the copyright to those songs. But an independent, unsigned artist can do whatever she likes with her own tunes. And while that represents a lesser threat to the industry, it represents a threat nonetheless; the prospect of a burgeoning "industry" that runs itself, outside of the RIAA's purvey, is at hand. And unlike piracy issues, there isn't a thing the RIAA can do about the rise of self-publishing musicians. Sites like MP3.com represent this concept in its full glory. Rather than requiring users to surf the Web for scattered, legitimate downloads by unsigned artists, users can search through archives of thousands of unsigned artists all in one place.

While MP3.com is focused on and dedicated to the promotion of unsigned artists, the site does indeed enjoy an arrangement with The American Society of Composers, Authors, and Publishers (ASCAP), which allows MP3.com to stream (not offer for download) the works of musicians signed to labels. MP3.com's relationship with ASCAP is outlined at *www.mp3.com/ascap/*.

Do the math: A good deal for artists

There's more than just exposure in it for the artist. Let's do the math. According to one artist who's been struggling to make it in the business for years, a band who sells 150,000 CDs through a medium-size label will still not be generating profit for themselves—yes, there's that much overhead in being associated with a label.* Furthermore, only around 5% of signed artists end up turning a profit—signing with a label is not necessarily a "ticket to ride." But if an artist presses his or her own CDs, and sells them for $10 each over the Internet, they can make a profit by

* It is for reasons such as this that the relatively successful group TLC was forced to file for bankruptcy in 1999.

selling only 15,000 copies. This artist would make $5 per CD in profit, netting $75,000 in sales. And, in fact, this is exactly what MP3.com allows artists to do, using their D.A.M. CD service (see Chapter 8 for details).

Presumption of Guilt

There's another, possibly more frightening aspect to the rise of self-publishing: Anti-piracy measures instituted by the industry—such as SDMI—may have a negative impact on the rise of this new distribution mechanism. As you'll see in more detail in Chapter 7, SDMI and similar mechanisms operate on a presumption of guilt. Because so much music in MP3 format is pirated, security mechanisms tend to make the assumption that anything in MP3 format is probably pirated. This is nothing but a slap in the face to the unsigned, unrepresented artist using the MP3 codec for legitimate purposes, to willfully and legally publish his own music. If security formats, technologies, and hardware are pushed on users by the industry, unsigned artists will be left right where they were before MP3 promised them an alternative.

Legitimate uses of MP3 are not constrained to music. One doesn't have to look too far to find other creative, legal, and legitimate uses of the technology. *The New York Times* makes spoken-word versions of its news available for download from its site. MP3LiT.com features hundreds of published and unpublished authors reading excerpts from their books, going on political rants, reading or improvising poetry, and offering self-help material, all in MP3 format. These and other legitimate uses of MP3 will be adversely affected by any attempt by the record industry to squash unprotected MP3 files in general.

Why MP3 Is Here to Stay

As you've probably heard, MP3 isn't the only digital audio compression format out there. As you'll see in Chapter 9, there are numerous alternatives to the format available to the public. In fact, many of them achieve even better compression ratios and/or better sound quality than MP3.

Not the best, but good enough

So why hasn't the public moved toward competing formats in a wild rush? Even MP3's most staunch supporters readily admit that MP3 is a less-than-ideal solution from an audiophile perspective. But like Betamax or the Macintosh, history proves that the best technology doesn't always win. The VQF and AAC formats are great examples of this in action: both offer superior quality, and often, smaller file sizes than MP3 as well.* Nevertheless, MP3 is entrenched—you might say that MP3 is to

* Many MP3 players also handle VQF and other audio formats; you can switch between them seamlessly.

the Internet audio industry as Windows is to the operating system market—functional and workable, but not the ideal solution, even though the depth of its penetration practically guarantees that it isn't going anywhere anytime soon.

More often than not, standards are set by the technology that establishes a strong user base first. And establishing that user base has as much to do with marketing, wide availability, and ease-of-use as it has to do with quality. And history also shows that a superior technology doesn't just have to be a little better in order to turn heads—it has to be way, way better. While some of MP3's competing formats are better than MP3, the fact of the matter is that MP3 is good enough. At decent bitrates, it sounds great, and the compression ratios are perfectly acceptable to most people. While some people make the point that the public will migrate to another technology as soon as something better comes along, this is not proving to be the case. Excellent alternatives do exist, but they're not taking hold (at least, they're not as of early 2000). For another format to surpass the momentum of MP3, it would have to have all of the following attributes:

- Smaller file sizes
- Superior audio quality
- Be free and unprotected

Ironically, it's that third point that makes all the difference. The first and second points are already satisfied by other file formats and codecs. But the public wants what the public wants, and they have spoken very clearly: they want lots of choice, available source code, and freedom from copying restrictions. The fact that this also means a huge headache when it comes to protecting intellectual property is another matter.

The source, Luke, use the source

Another important key to MP3's popularity is the fact that it's highly available. Rather than hanging from the awnings of a single company, literally hundreds of MP3 encoders and players are available for virtually every operating system under the sun. And, of course, there are millions of MP3 files out there. Most people don't have the time or the inclination to evaluate all the competing formats—they go with what works.

But perhaps most importantly, Fraunhofer and Thomson have made sample code available to the world. As you'll see in Chapter 7, that doesn't mean the MPEG codec is open source exactly (technically, developers still owe licensing fees to the codecs copyright holders), but it does mean that it's been possible for competition to take root and for a multiplicity of approaches to flourish. From the most complex and exacting command-line applications to dirt-simple, all-in-one GUI applications, a quick search of any MP3-related site will uncover as many approaches to MP3 encoding and playback as there are types of computer users. That kind of diversity has a way of generating a deep, grassroots kind of spirit around a technology—one need look no further than the amazing success of Linux to witness the kind of results that can arise in an open, "biologically diverse" environment.

Among the contenders to MP3 is Microsoft's own digital audio compression format, Windows Media. To be sure, Windows Media does deliver good sound at much smaller file sizes than MP3 (at the lower bitrates).* Windows Media offers security features for artists and labels who want to take advantage of it. It's built-in streaming features make it competitive with MP3 and RealAudio simultaneously. But the real kicker, of course, is the fact that Windows Media is built into later versions of Windows, giving it a level of direct access with which no one can hope to compete. But at this writing, Windows Media was not gaining any kind of serious buy-in from users (though that doesn't mean it never will). Think about that: if not even Microsoft can break into a market it desperately wants to occupy, then it's a pretty safe bet MP3 will be sticking around for a good long time. MP3 is in a position relative to digital audio compression that Microsoft itself is in relative to operating systems—it got in on the game early and established a strong user base, making it the standard. And standards have a way of remaining standards (imagine what would happen if you purchased a new toaster and it came with an electrical plug that didn't fit your wall sockets). Windows Media and other competitive codecs are covered in Chapter 9.

"Open source" music

To be sure, there are a handful of bands who have adopted the "open source" approach to music. The Grateful Dead is probably the most famous example of this force in motion. For decades, attendees at Dead shows were not patted down for hidden microphones; rather, taping of live shows was actively encouraged by the band and its management. The Grateful Dead realized that when you encourage a meme to spread and make your fans feel like they're an integral part of the process, the fans become loyal to the family; the Dead always sold out their concert venues. While industry execs still fear that giving things away means cannibalizing their own business, the important lesson is that if what you give away is good, then people want more of it, not less.

* Though the consensus is that the format creates noticeable artifacts, especially with certain types of sounds—see Chapter 9 for more information.

Today, with the Dead a thing of the past, this spirit lives on at sites like *www.deadabase.com*, where users can trade MP3 versions of all those live shows. The only restriction placed on people downloading tracks is that they aren't allowed to profit from them, even by selling ad space on their sites. In subsequent years, bands such as Phish, Metallica, the Black Crowes, and the Dave Matthews Band have all jumped on the "open source music" bandwagon, although this always applies to tapes of live shows rather than of albums.

Comparison between the MP3 movement and the open source software movement doesn't stop there. Some argue that the entire history of music is open-source-like, noting that Homer's *Iliad* and *Odyssey* were composed collectively over the years by "skilled Greek emcees," and that the modern practice of resampling, remixing, and audio collage forms a perfect audio analog to the open source movement.*

Give the people what they want

Finally, there's the fact that MP3 is unprotected. Ironically, it is this fact that labels and signed artists fear the most—any MP3 file can be duplicated a bazillion times, traveling from one user to the next like an unrestrained virus. Right or wrong, millions of MP3 users are interested in the technology simply because it's so easy to get "free" music. While the record industry struggles to introduce protected formats that will allow us only to copy our own music, for our own use, the public apparently wants just the opposite—a simple format that works the way they expect the rest of our computer documents to work, not one that imposes restrictions and limitations on who can get at the contents of a particular file, and on what machine.

Protected audio compression formats are going to make some headway, because the industry is going to make sure that they do. But there are too many people who appreciate MP3 precisely for its openness for it ever to go away. While some songs may eventually become available only in protected formats, people will continue to use available MP3 tools.

Biting the hand that feeds

The question is, does the freedom afforded by MP3 ultimately threaten the very industry it so enjoys? The industry certainly made this argument at the dawn of cassette tapes, and again at the dawn of video tapes, only to be proven wrong for the most part. But as mentioned earlier, infinite digital copying really is a horse of a different color, and this time around, the industry has a right to be afraid. But is that really such a bad thing? MP3 may force the industry to change the way it does

* Julian Dibbell, "You Say You Want a Revolution?," *www.mp3.com/news/258.html.*

business. Labels may ultimately *have* to give a few tunes away in order to sell an album. They may at some point *have* to start selling songs one tune at a time rather than as complete albums—the whole concept of what constitutes an "album," which is currently an artifact of the amount of recording time that can be fit onto a vinyl LP or CD, may be permanently affected. They may even be forced to lower the unjustifiably high prices they currently charge for compact discs.

In short, MP3 and related formats may force the record industry to downsize its approach to the whole game, and to come to grips with the fact that they can't fight this phenomenon forever. For its part in the debate, the watchdog organization Electronic Freedom Foundation (*www.eff.org*), has stated that:

> ...architecture is policy. Given the choice, consumers would choose to purchase music in open formats. We believe that artists who allowed their works to be distributed in open formats would gain competitive advantages over artists who locked up their work.

The EFF's fear is that if the Big 5 record companies (Sony Music Entertainment, EMI Recorded Music, Universal Music Group, BMG Entertainment, and Warner Music Group) are allowed to monopolize the trade in online music, independent artists will be stymied just as their big opportunity appears on the horizon.

Can't we all just get along?

The record industry is not going to simply go away. They'll still be around to promote artists, to manage tours, to print and sell t-shirts and bumper stickers, and to run the web sites of their artists. As one representative put it, "Without the help of the industry, the independent artist's experience may be closer to that of a street performer, rather than a stage performer." It is, after all, possible that a marketing machine more powerful than MTV itself may rise like a Phoenix from the chaos of the Internet.

The industry has yet to feel the full impact of MP3, but they will—it's inevitable. The genie is already out of the bottle.

4

Playlists, Tags, and Skins: MP3 Options

In this chapter:

The stock $35 SoundBlaster-compatible card, driving a pair of $25 plastic desktop speakers, has become a de facto standard throughout the '90s. But as computing power increases in accordance with Moore's law,* media production and playback functionality takes a more central role in the consumer computer. With the advent of audio cards like the SoundBlaster Live! and the increasing popularity of satellite/subwoofer systems and computer-to-stereo connections, quality file-based audio becomes a realistic possibility for millions of users. MP3's natural home may be the wildlands of the Internet, but its reach is quickly being extended into other domains as well. As homes embrace the "convergence" model, computers, stereos, and televisions are blending together into all-in-one, networked infotainment centers, and stereo component manufacturers are introducing hardware-based MP3 players. Sales of portable units such as the Diamond Rio and Creative NOMAD are skyrocketing, and more and more cars are being fitted with MP3 playback units as well. As a result of all this, it becomes more important than ever to think beyond the simple task of encoding and playing MP3 files. Increasingly, people need to move large amounts of data between their computers and their playback devices, optimize the quality of their MP3 collections, maintain a well-organized filing system or MP3 database, make sure all of their files are tagged with useful meta-data for future reference, parse meaningful playlists out of collections extending into the gigabytes, and extend the reach of their MP3 players to handle unusual situations.

* See the glossary for definition and more information.

MP3 Options and Considerations

Once you've mastered the basics, you'll want to expand your horizons a bit and start checking out the many advanced capabilities, cosmetics, playlist generation techniques, ID3 tagging tools, plug-ins, and other toys available. There's a lot out there, and we'll only have space to touch on a handful of examples of useful "peripheral" software here. Do some searching through your favorite MP3 site or software library and you'll find hundreds of tools not discussed in this book.

Before we get to the goodies, however, we'll discuss some of the issues affecting the quality of MPEG audio. Because the compression format discards some data, MP3 already stands on shaky ground from a fidelity standpoint. That's not to say the quality of MP3 stinks, as some critics claim—but there are some things you can do to optimize the quality of your MP3s during encoding and during playback.

On the lighter side of things, MP3 players can often be dressed up in "skins"—small collections of bitmap images that sit on top of your player to give it a customized appearance. You can download skins from the Internet or create your own, although the process is admittedly a bit tricky. Don't worry—we'll show you how to create your own skins from start to finish.

Two of the most important "peripheral" technologies you'll meet in this chapter are ID3 tags and playlists. We discussed ID3 tags in a technical vein in Chapter 2, *How MP3 Works: Inside the Codec*; they're the extra space in MP3 files that let you store "meta data" about a file, including the artist, album, and track names, as well as genre, year, and personal comments. An up-and-coming modification of the ID3 specification, called ID3v2, is much more powerful, and lets you store a nearly unlimited amount of additional information (up to 256 MB). We'll check out the many ways in which ID3 tags can be created or edited, either directly through your MP3 player or via separate software.

One of the most enjoyable aspects of building a large MP3 collection, and one of the things that makes it so different from building a tape or CD collection, is the fact that you can mix and match songs into a customized sequence at a moment's notice. These personalized "albums" are called playlists, and consist of simple text files referencing the locations of tracks scattered across your system. Playlists can be created by dragging tracks one at a time into an MP3 player's playlist editor, by dragging entire directory structures onto your MP3 player, by trawling your disk with scripts, by running command-line queries, or by scanning through your collection with a database-like solution like Helium.

Ensuring that your files have solid ID3 tag data (accomplished either while encoding or after downloading) can be very important later on when you want to start creating playlists based on query criteria, such as "Create a playlist of all songs on all of my disk volumes by either Neil Sedaka or The Carpenters written between 1971 and 1975 that don't have the word 'schmaltzy' in the comment field."

The functionality of many MP3 players can be extended by installing plug-ins, much as you would for Netscape Navigator or Adobe Photoshop. A bewildering variety of plug-ins is available, which will let you do everything from controlling your MP3 player via infrared remote control to applying sound effects to displaying wild visualizations of the music as it plays. We'll take a look at a few of the more popular plug-ins later in this chapter as well.

An increasingly popular way to listen to MP3s is not to save them to disk or to encode them, but to listen to them in real time as they're broadcast out over the Internet. While the process of running your own Internet broadcast is covered in Chapter 8, *Webcasting and Servers: Internet Distribution*, we'll show you how to tune into the two main types of streamed MP3 (broadcast and on-demand) in this chapter.

Finally, you'll want to make sure your system is well-tuned for MP3 encoding and playback, so we'll talk about the hardware requirements—what kind of processor you need, sound and video (yes, video) card interactions, disk speed issues, and the like. We'll also show you how to benchmark MP3 players and encoders in case you have an older machine and want to make sure you're not dragging down the rest of your system's performance.

Equalization and Sound Quality

While true audiophiles may never accept lossy compression formats as a home standard just for the convenience factor, most of us *do* place a very high priority on the many conveniences offered by the MP3 format. So we're left with a question: How do we make sure we're getting the most from our MP3 collections?

Can Quality Be Measured?

Before even *thinking* about evaluating standards of quality, it's critical to understand that "this way lies madness." Audiophiles have wrestled for decades to achieve "objective" measurements that would fairly represent the quality of a recording. And while there are many objective measurements that can be made for any given signal, it's important to understand that the mathematics of audio measurement and the subjective experience of quality are two different animals. Furthermore, the mechanics of lossy compression (discussed in Chapter 2) more or less nullify the validity of just about any objective criteria. In other words, the only tests that matter for MP3 audio are subjective tests, i.e., real-world listening tests. Your ears don't lie. Even so, it's possible that the file you encode to your own standards today could end up being played on much better equipment tomorrow, so it pays to shoot for quality a little above your own thresholds.

The most important thing to keep in mind when testing for MP3 quality is that your computer probably isn't the best place to do it. How's that? Unless you've invested a fair sum in high-end computer audio equipment, chances are that your home stereo sounds a lot better than your computer. And if you're planning to keep your MP3 collection around for a long time, your home stereo may very well be the ultimate destination for your MP3 files. Because your sound card and computer speakers may mask a whole lot of subtlety, it's important to eliminate those components from the testing chain so that you can tell which limitations are introduced by your hardware and which are introduced by the MP3 encoding itself. In other words, you want to isolate the encoding as much as possible and give it an optimum environment in which to be tested.

Conducting listening tests

Since "CD quality" is our benchmark here, you want to get your MP3s onto an audio CD. To run this test, you'll need access either to a CD burner or to a dedicated MP3 playback device. Find a couple of songs with which you're well-familiar, preferably ones that include a wide range of instruments, including those from the brass and strings families. It's often easier to detect limitations in the MP3 codec with non-electronic music, and the brass and string families do an especially good job of putting the codec to the test. However, it's also important that you choose music you enjoy personally. To be well rounded, you might want to choose one each of your favorite jazz, classical, and rock tracks, for example.

If you have a portable or home stereo MP3 player, just patch it into your stereo as described in Chapter 3, *Getting and Playing MP3 Files*, and load it up with encodings created as described in the next section. The same principles described in this section apply whether you're working from a test CD or from a dedicated playback device.

Next, you'll need to save a reference track for each song, meaning you should rip uncompressed versions of each track to your hard drive (typically in WAV format, though Mac OS users will want to use AIFF format). See Chapter 5, *Ripping and Encoding: Creating MP3 Files*, for details on ripping tracks directly from CD without encoding them. For the actual compression, you'll want to use the highest quality encoder you can find (again, see Chapter 5),* and encode each track a

* The test described here can also be used to test various encoders against one another. The only difference is that you'll want to encode the reference track at the *same* bitrate, but with different encoders. In fact, you may want to encode at various bitrates with each encoder in your test, so you can determine whether a given encoder excels at low bitrate encodings but falls down at higher bitrates, for example.

number of times, at different bitrates. As you encode, give each file a descriptive name, including its bitrate. For example, you might use the following scheme:

```
Improv.Spacious.64.mp3
Improv.Spacious.128.mp3
Improv.Spacious.160.mp3
Improv.Spacious.192.mp3
Improv.Spacious.256.mp3
```

Automating Test Encodings

If the process of encoding the same track over and over again sounds tedious, it is. Here's a script you can use under Unix/Linux or BeOS to automate the process, assuming you have the LAME encoder somewhere in your path. This should be very easy to modify to work with other encoders:

```
#!/bin/sh
FileName="$1"

# Start a loop and encode once at each bitrate specified
for i in 64 96 128 160 192 256; do
        echo Creating "$FileName".$i.mp3
        lame -ms -b$i "$FileName".wav "$FileName".$i.mp3
        echo
done
```

If you're on a DOS/Windows machine, here's a batch file that will achieve the same effect:

```
Echo off
rem example
set filename=%1

if not exist %filename%.wav goto error
rem The following must go on a single line -
rem We had to wrap it to two lines for this book
for %%B in (64 96 128 160 192 256) do lame -ms -b%%B %filename%.wav
%filename%-%%B.mp3
goto exit
:error
  echo File %filename%.wav does not exist
  goto exit
:exit
```

Whether you're running the the bash or the DOS version, save the script as a file in your system's path with a filename like "TestEnc" and then run:

```
TestEnc filename
```

assuming your input file is called `filename.wav`. Of course, these scripts assume that LAME can be found in your system path as well.

Facebook's Exhibit No. 1062
Page 36

Once you've encoded each sample at a variety of bitrates, you'll need to translate them *back* to WAV format and burn the resulting tracks to an audio CD (although your burning software may take care of this conversion for you; see Chapter 5 for details). Because the filenames won't be present on the audio CD, take care to print out or write down the track numbers and the corresponding bitrates so you'll have a reference sheet later on. Once your audio CD is complete, find the highest quality stereo you can (yours or a friend's), grab your reference sheet, dim the lights, relax, and listen through the CD. While you may want to skip back and forth between the reference tracks (which will sound identical to the original CD audio tracks) and the encoded tracks later on, you should first listen to the disc all the way through. Don't be too quick to flip back and forth between tracks, as that will only lead to schizophrenia and ear fatigue. The idea is to internalize the "flavor" of the tracks relative to one another. Evenings may produce the best listening environment, as you'll likely get less ambient noise and less sensory competition from vision (assuming some darkness). If you've been listening to loud music or noises during the day, save the test for another day—your ears will already be fatigued. Self-run audio tests using your own hearing as a "benchmark" should always attempt to isolate variables and incorporate a best-case scenario, against which everything else should be measured.

If your home stereo or entertainment system includes an on-board digital-to-analog converter which offers settings for surround sound or for various room effects (concert hall, small room, etc.), it's important to disable these options and make sure your system is set on the defaults. If you use an equalizer or other audio "enhancing" device, disable it or run it flat (unless it's a "set-and-forget" processor carefully tuned to neutralize a particular room). You're testing your encodings here, not your equipment. Artificial interferences in signal flow will compromise the integrity of the test.

As you listen, take note of the most subtle sounds in the reference track, and try to determine whether those sounds are still present in the various encodings. Can you hear the clarinetist taking that breath just before his solo? The sound of the guitarist's hands gliding over her strings? What about the sense of space (soundstage expansiveness)? Can you pinpoint the location of various instruments in the recording studio as accurately with files that have gone through the MP3 process as you can with the reference track? What about dynamic range? Are you detecting certain subtleties of the high and low ranges being chopped off or smeared out? Perhaps most importantly, do the MP3 encodings have a sense of "presence?" Can you close your eyes and imagine that the performance is taking place there in your living room? Does a given encoding make the performer sound sort of "mechanical" or "electronic?" Does the performance seem to be more or less intense with any of the encodings? Do any of them do a better job of making you want to tap your foot or sing along or dance, or do any of them make you feel edgy, or bored? Do the MP3 encodings sound "hollow" or "swishy" to you? Chances are, you'll be amazed to discover just how much quality has been lost in the 128 kbps and lower encodings.

The bitrate at which you literally cannot tell the difference between the reference track and the MP3 encoding is the bitrate at which you should start encoding your MP3 tracks, provided you're willing to part with the extra disk space this will consume (of course, if you'll be burning a lot of tracks to writeable CD for permanent storage, this may be less of an issue). To be really safe, and to make sure your tracks will sound great no matter what playback system they end up on one day, encode at a bitrate *higher* than the one at which you couldn't tell a difference.

If you don't have access to a CD burner, then your next best bet is to connect your sound card's line-out jack to one of your home stereo's inputs, and try similar tests. However, keep in mind that unless you use a fully digital transfer system (i.e., using digital coax or optical cable from the sound card to your digital receiver or external DAC), you'll be hearing the limitations of your sound card's digital-to-analog converter.

Digital to Analog (and Back)

As you know, computers store and manipulate information—including audio information—digitally. But your ears are analog devices, and so are speakers.* Somewhere in the process of getting information out of your computer and into your ears, the signal must be converted from digital to analog. The device that affects this conversion is called the *digital-to-analog converter*, or DAC, and its role is crucial. The quality of the DAC plays a huge part in the overall quality of the resulting audio pouring from your speakers or headphones.

In the case of most stock sound cards, the DAC is a part of the card's chipset. However, some higher-end sound cards offer digital outputs, which can be hooked up directly to the digital input of a digitally equipped stereo receiver or pre-amplifier. If your receiver or amplifier doesn't have digital inputs (they'll be clearly marked), you can purchase an external DAC or use the DAC built into a DAT deck, Minidisc player, or DCC, which will then sit between your computer and your receiver/pre-amp. The quality of external DACs is often directly related to their cost; a very high quality DAC can cost $1,000 or more, though you should be able to pick up a suitable model for a few hundred bucks.

* Or, at least, most of them are; completely digital PCM speakers are beginning to appear on the market.

Audio Test Checklist

To set up a test like this in a hurry (actually, it's best not to hurry when doing things like this), remember these simple steps:

1. Choose a reference track from your collection with which you're familiar. Look for a reference track with a wide variety of instruments and not too much density. A wall of guitar distortion is not a good choice, though you'll get mixed results with straight electric guitar under various encoders. Well-recorded acoustic guitar, vocals, strings, and brass will show you more limitations in the MP3 codec than hard rock. Rip the reference track to uncompressed WAV or AIFF format.
2. Using the best encoder you can find, encode the reference track at a variety of bitrates, using default settings in the encoder. Alternatively, test various encoders at the same bitrate.
3. Decode the encodings back to WAV or AIFF (see Chapter 5 for details).
4. Burn the reference track and the decoded encodings to an audio CD. Keep a list on paper of all the tracks going onto the CD.
5. Find the best stereo you can, relax, close your eyes, and just enjoy the music. Don't start analyzing carefully or paying special attention to subtleties and nuances until after you've relaxed into the music.

In any case, by turning the job of conversion over to a dedicated unit, you can bypass the typically poor quality built-in DAC on most sound cards, and get a far better resulting sound stream out of your system.* When purchasing a sound card, make sure it has digital outputs in addition to the standard line out. If you're serious about quality, do some research online to find out which cards have the best on-board DACs. If you're really, really serious, consider a high-end outboard (external) DAC.

On the flipside, the reverse process—*analog-to-digital conversion* (ADC) may also come into play, depending on your needs. If you plan to encode music from analog sources such as tapes, LPs, or microphones, you'll also appreciate the extra input quality you'll get by making sure your source signal is stored in the computer after having passed through a high-quality ADC (either onboard or outboard), rather than the one built into your average sound card.

* Of course, the rest of your system should be of high quality before you go spending money on a DAC. Putting an external DAC in a system with poor speakers, for instance, probably won't result in noticeably better sound.

Equalization

Most GUI MP3 players include some sort of equalization mechanism, typically accessed by clicking a button on the player's interface labeled "EQ." Bringing up the equalizer usually displays a row of sliders (most often 10, though there could theoretically be any number of them). So, what exactly is equalization, and what does it get you? The simplest way to conceptualize equalization is to think of the bass and treble controls on your home stereo, and imagine that you had a separate knob for each of many much finer slices of the frequency spectrum, affording you much tighter control over the resulting output signal.

Why equalize?

The point of equalization, of course, is to make the resulting audio sound better. EQ can be applied to compensate either for poorly recorded audio, for limitations in your speaker quality or design, or for the speaker placement or acoustics in the listening room. The ultimate objective of equalization is to obtain a "flat" frequency response (hence the name "equalization")—one in which the signal does not sound "colored" by the signal processing chain, storage media, or playback equipment.* Equalization can also be used to eliminate audio artifacts stored in the original recording. For example, you may be able to partially correct a recording with an annoying 60Hz hum, or to partially remove some of the scratchiness of encodings taken from LPs (though the latter can be pretty tricky, since scratchiness typically crosses many different frequencies, and you don't want to throw the baby out with the bath water).

Despite these advanced capabilities, many (if not most) people use equalizers not so much to correct for anomalies as to boost the frequencies they like and to diminish the ones they don't. For example, hip-hop fans may want to pump up the bass frequencies, while classical listeners may want to "sweeten" the strings.

Before you go tweaking your equalizer's sliders with wild abandon, consider the equalizer as something to be experimented with gently and systematically. Listen carefully to the resulting sound after each adjustment, and pay attention to how manipulating one sub-band affects your perception of others—diminishing one sub-band can cause others to sound more prominent. Note also that if you find you have to equalize heavily to compensate for room acoustics, poor speakers, or a poor sound card, you should consider addressing these problems directly. Equalization is a band-aid, not a panacea. Be aware of the potential for over-equalization, and of the limitations of EQ principles in general:

* Note, however, that many recording studios use equalization to *add* color, not to remove it—a fact which drives some audiophiles nuts.

- Without a lot of expensive test equipment, you have no concrete goal to aim for—you're just tweaking until something in the sound "feels" better for one reason or another.
- EQ bands come in predefined widths and slopes, but your equalization needs don't—they're typically more "fuzzy" than the usual set of EQ bands allows for (it's kind of like trying to apply mascara with a hairbrush... and no mirror).
- Analog equalizers can introduce distortions, especially phase distortion. Digital equalizers can minimize some of these effects.

It's easy to confuse louder and brighter sound for better sound. Optimum quality is achieved when music sounds as natural as possible—equalize with an ear toward a natural sound atmosphere. Over-equalization can easily degrade sound quality. In other words, the answer to the question, "Why equalize?" is often "You shouldn't, unless the situation demands it." It's easier to ruin the sound than to improve it. Experiment with your EQ, but don't overdo it.

Artificial Audio Enhancers

Every now and then, you'll come across a product that claims to improve the quality of computer audio dramatically. Typically, these products (QSound's iQ, *www.qsound.com,* being an example) work by running a series of proprietary algorithms over the outbound audio stream and applying digital effects to the stream just before it heads for your sound card. Much like the sample effects that come with some sound card drivers, or the presets found in some home theater amplifiers, these products create the illusion that the sound stage extends out beyond the edge of your speakers, generating a 3D sound stage on the fly. These products can also take a monophonic sound stream, such as you'll typically get when listening to SHOUTcast or icecast streams, and "turn them into stereo."

Do these products work? Well, there's no question that the effects are quite dramatic, and you'll most certainly notice a difference over normal playback. But the fact is, these products are creating a signal that isn't present in the source by playing auditory tricks with channel shifting and phase relationships (bouncing sound waves against one another). What these products do is unnatural and unrealistic, and while they may make a cheap computer sound system sound more dramatic, they certainly won't bring you any closer to the artist's vision of how the music was intended to sound.

If you choose to experiment with products like iQ, keep in mind that the signal you'll be hearing is not "true." This may or may not matter to you, depending on your perspective. My position is that effects like these are best reserved for use with games, but should be disabled when listening to music.

The Fletcher-Munson curve

While you should, of course, equalize as necessary to hit the sweet spot that works for a given situation, there's one commonly used configuration of which you may want to be aware: The classic "Fletcher-Munson" curve, which takes advantage of the fact that the ear is not as sensitive to low and high frequencies when music is being played at low volumes. Thus, boosting these frequencies somewhat above normal can help music to sound more natural at lower volumes (Figure 4-1).

Interestingly, the function of the loudness button on your receiver is similar; apply an Fletcher-Munson equalization curve to the music makes it sound more natural at lower volumes.

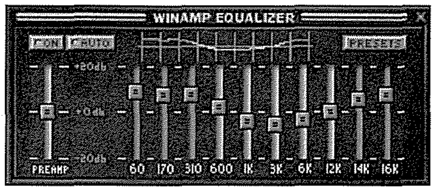


Figure 4-1. Fletcher-Munson curve, for use in low-volume situations

Note that many MP3 equalizers, such as WinAmp's, let you store equalization curves as presets, which can be loaded up again later without painstakingly re-creating them. Depending on your player, it may also be possible to store a prime equalization curve directly in the MP3 file itself, so you can optimize all of your songs' equalization curves independently. This capability is dependent on the fact that the ID3v2 specification allows for a huge array of meta-data storage capabilities in MP3 files; in other words, you must be using an ID3v2-compliant decoder capable of storing EQ presets. At this writing, this capability remained purely theoretical, though it should become a realistic possibility as the ID3v2 spec builds more momentum. See the ID3 section of this chapter for details.

Working with EQ presets

Because different tracks may require different equalization curves, WinAmp lets you associate an EQ preset with any given track. Here's how to associate an EQ curve with a track:

1. Play the track normally, and press AUTO in the EQ interface.
2. Dial in your EQ curve.
3. Click the Presets button and navigate to Save → Autoload Preset.
4. Allow WinAmp to associate the current track's filename with the current EQ curve.

After creating an association between a track and an EQ curve, the association is stored by WinAmp in a file named *winamp.q2* in the WinAmp installation directory (the file cannot be edited by hand). Any time AUTO is enabled in the equalizer and that track is played, the preferred EQ curve will snap into action.

Uncooking Bad Transfers

Every so often, you may end up with an MP3 file that's been corrupted during transfer somehow, usually by being transferred in ASCII mode rather than binary. If you should encounter one of these, search the Internet for *uncook.exe*, Uncook95, or OK Uncook, all of which do essentially the same thing—repair the damage that was done to the file during transfer and make it playable again. Linux and BeOS users may want to keep a copy of mp3asm (*www.ozemail.com.au/~crn/mp3asm.html*) around for the same purpose.

If you're using a recent browser and the web server you're accessing has been set up properly, this should never happen, though it's nice to know there are tools available to correct the situation if it does.

In most cases, the equalizers built into MP3 players will have no effect on the audio stream when playing audio CDs, since CD audio is often routed directly from the CD player to the sound card without moving through software first. In this case, the MP3 player is just used as a convenient interface onto the CD transport's control buttons. However, there are plug-ins available that will "steal" the CD audio signal and route it through the MP3 player (the NullSoft CD/Line Input plug-in for WinAmp, with the "sampling" option enabled, for example). See the "Plug-ins: Extending Your Reach" section in this chapter for details. Once the CD plug-in has been enabled, you should be able to use the equalizer and various visualizers to interact with CD audio.

ID3 Tags and Playlists: The Virtual Database

ID3 tags and playlists are two of the most important extended topics for people who take their MP3 collections seriously. Taken separately, both technologies are important in their own right, but taken together, these two functions can complement one another in very powerful ways, giving you the means to sort, store, organize, name, and build highly customized virtual collections. Because these two technologies are so synergistic, and because many tools exist to help you organize both of them simultaneously, we'll treat both of them together in this section.

What Are ID3 Tags?

Every MP3 file has the ability to store "meta-data" related to the track in the file itself, in the form of what are known as "ID3 tags." For example, a file's ID3 tags may store the song's artist, album, year, genre, and comments in ID3 tags. Many MP3 players have the ability to read ID3 data out of your files, and to display this information in the MP3 playback interface. Thus, giving your MP3 files descriptive names isn't the only—or even necessarily the best—way to identify the tracks in your collection.

Conveniently, ID3 tag information can either be included in the file at the time it's encoded or added in later. Most of the better encoders will provide a number of options to let you control whether and how ID3 tag data should be written to the file. While some MP3 players include the ability to edit, as well as to display ID3 tag information, you may want to download and install additional utilities that specialize in ID3 manipulation, for maximum control. We'll look at some examples of ID3 tag editors later on.

ID3v1 vs. ID3v2

It's important to understand that there are two basic flavors of the ID3 specification, conveniently named ID3v1 and ID3v2. Of the two, ID3v2 is vastly superior, for three reasons:

- Unlike ID3v1, ID3v2 imposes no arbitrary limitations on the amount of storage space available for meta-data, which means there's plenty of space for storing images, complete lyrics, performance notes, equalization presets, et cetera.
- ID3v2 data is stored at the beginning of the file, rather than at the end. This means that ID3v2 is much better for use in streaming situations, where it's important that users be able to see and use the meta-data while the song is being played, rather than after the download or broadcast is complete.

As useful as it can be to have ID3v2 data appear at the beginning of the file, this state of affairs also presents a problem—especially for older players. A player built to the ISO spec should simply skip over the ID3v2 data that it doesn't understand, and move ahead to the valid MP3 data. But exactly how far should the player seek through the file for an MP3 header frame before giving up and deciding that it's not currently dealing with an MP3 file after all? Unfortunately, the ISO spec is vague on the matter, which leaves the door open to interpretation by developers. Some believe that a player should seek forever (after all, ID3v2 data can occupy as much as 256 MB of space), while others feel that a player should give up after a certain number of bytes and assume that they're not dealing with a valid MP3 file after all. The user could have innocently tried to play a giant AVI movie through his MP3 player, for instance, which would cause a player that seeks forever to basically hang. Since the spec is not clear on the matter, the debate may never be resolved. The best solution may be store this data separately, in a hypothetical, centralized MP3ID database, for example, although that's just one of the ideas being floated around.

- The ID3v2 spec is open-ended and flexible, meaning it can be extended in the future to accommodate as-yet-unseen needs. Users may even be able to develop their own custom solutions, given the availability of future ID3v2 tag editors and management solutions. For example, the built-in "Popularimeter" field could be used to let users rate all of their tracks on a scale of 1 to 100, and then create custom playlists based on a scale of favorites.

The number of possible applications for ID3v2 data is staggering. Right off the bat, it gives artists and labels a place to store copyright information, terms of use, and proof of ownership. There are even possibilities for cryptographic security being embedded in MP3 files, thanks to ID3v2. For the end user, ID3v2 means the ability to store full-size, color scans of album covers, complete lyrics (even synchronized lyrics, for use in Karaoke situations), multiple URLs for further information, music reviews, extended information on the recording session, and so on.

While users of ID3v1 players sometimes discovered that the spec, with its 32-character limitation, didn't even give them enough space to store very long artist or album titles (such as Stereolab's album "Cobra and Phases Group Play Voltage in the Milky Night"), the ID3v2 spec allows for a maximum of 256 *megabytes* of additional tag storage space, meaning users won't bump into arbitrarily imposed limitations. Of course, if you fill a file with 256 MB of ID3 data, don't expect many users to want to download it from your site.

Because a surprising number of older players are *not* ID3v2 aware, and because of the problem mentioned earlier regarding the length of time an older player should seek for MP3 data before giving up, problems do arise. For example, if you use MusicMatch Jukebox to embed an image of an album cover into an MP3 file using Jukebox's ID3v2 tag editor, you'll find that the file will no longer play in the command-line mpg123 player, while Xmms may be able to play the file but won't show any ID3 tags for it.

Despite these issues, ID3v2 is a spec with huge potential. One of the most popular applications to take a stab at the ID3v2 space is MusicMatch JukeBox. At this writing, ID3v2 support in MusicMatch was limited to a minority subset of the total frames available (although these were the most significant ones). In addition, MusicMatch was taking a proprietary approach to storing some categories, including tempo, preference, and mood, which some feel could have been better handled had MusicMatch gone with standard ID3v2 frames. Finally, MusicMatch's implementation of the Content Type frame was still incomplete, and resembled ID3v1 more than ID3v2. Still, somebody had to be first on the dance floor, and the company's commitment to driving the standard forward should be appreciated. However, there's a lot of work remaining to bring their product into full compliance with the specification.

So how do you tell which ID3 spec is in use on any given file? There's no way to get this information just by looking at a file; the best way is to open the file in a tag editor capable of reading both versions of the spec, and looking to see where in the editor's fields information shows up. Any older MP3 file can be "upgraded" simply by opening it in an ID3v2-capable player or editor and adding extended information to it. In fact, some utilities will automatically migrate all ID3v1 data to equivalent ID3v2 fields. Opening an ID3v2-enhanced MP3 file with an older player will probably result in you seeing nothing in the ID3 info fields. The file itself, however, should still play properly. If it doesn't, you should consider changing or upgrading your MP3 player software.

Having accurate information in your files' ID3 fields is important for several reasons. If you ever edit or screw up your file's names, you can always just open the file in an ID3-capable player, read the author and song title, and rename the file accordingly. Some tools will even do this for you, automatically and programmatically, so you can assign ideal file names to every MP3 track on your system. ID3 tags are displayed in many playlists, so that you can display only song names, only artists, or only the comments. ID3 tags are commonly used by hardware MP3 players such as portable "Walkman"-type devices and home MP3 units. And some programs and utilities will let you treat the ID3 info in your files collectively, as if your entire collection was a huge, multi-field database.

For more information on the ID3 spec and its evolution, see the ID3 Standards Group web site at *www.id3.org.*

What Are Playlists?

After all the talk about how the MP3 revolution represents a rebirth of the "custom cassette" revolution of the '70s and '80s, the arena of playlists is where you finally get to see that promise become a reality. In their simplest form, playlists are simply text files containing paths to MP3 files on your system—one line in the file for every track referenced. The beauty of playlists is that they can be constructed to reference arbitrary tracks, rather than being limited to entire directories. For example, you might have a playlist referencing only the tracks in your collection in the Free Improv genre, or all tracks by The Carpenters, or all tracks you're temporarily storing in a holding bin until you decide what to do with them. There are many ways to construct playlists. Most users will create them by simply dragging tracks (or entire directories) out of the system's file manager (Explorer/Finder/Tracker, etc.) and into an open playlist window, then saving the new playlist to the hard disk. However, playlists can also be constructed from the command line by way of scripts, or from database searches. As you'll soon see, real playlist flexibility comes into play when you use your files' array of ID3 tags to search on specific criteria, then save the resulting found set as a playlist file.

Because they're constructed with such simplicity, playlists are nearly universal; for the most part, a playlist created with any method will work in any MP3 player (on the same operating system). Of course, the exact format of the lines comprising a playlist depends on the path style on your operating system. For example:

```
DOS/Windows - C:\DATA\MP3\ALBUMS\CAN\DELAY\PNOOM.MP3
Mac OS - Main HD:Data:MP3:Albums:Can:Delay:Pnoom.mp3
Linux - /usr/local/home/shacker/data/MP3/Albums/Can/Delay/Pnoom.mp3
BeOS - /gorgonzola/home/data/MP3/Albums/Can/Delay/Pnoom.mp3
```

So while a playlist will work on any MP3 player on your operating system, it wouldn't on another OS. (This really isn't much of an issue though, since people organize their collections very differently and it thus makes little sense to share your playlists with others, unless you're trading around data CDs full of MP3 files). Even if you're a Linux or BeOS user storing your MP3 collection on a Windows partition and then accessing it from Linux or BeOS, the paths to the files on the Windows partition will still appear differently, so you won't be able to share playlists between different OS partitions.

It's worth noting that most users will seldom see the actual text of a playlist file. While you can open a playlist in any text editor for viewing or hand-editing, it's almost always easier to view and manage your playlists directly in your MP3 player's built-in playlist manager, since these typically give you drag-and-drop, shuffle, sort, total playback time, and other MP3-specific features you probably won't find in a text editor. Playlist manager/editors generally do not show you the full paths to your MP3 files, displaying only the actual track names. In many cases, playlist editors will read the ID3 tags out of each file in the list and display those (if found) rather than the filenames. As you'll see in Chapter 8, playlists may also contain the URLs to MP3 files on the Internet, so that you can stream known favorites "out there" directly to your MP3 player.

Because most MP3 players can accommodate multiple audio file formats, including WAV, RealAudio, AAC, and others, there's no reason you should restrict your playlists to referencing MP3 files alone. If you've been collecting multiple audio file formats on your system and your player supports them all, feel free to create playlists referencing multiple file formats. If such a playlist does get fed through a player that doesn't understand some of the referenced formats, it should gracefully skip those files (assuming it's well-designed).

Playlist formats

The form of playlist described above is known as an M3U list,* and is named as *playlist.m3u*. In some cases, you may also encounter playlists with a *.pls* extension. These files are virtually the same, with a couple of internal differences. They begin with an identifying block marking the file as a playlist, precede each track with a numerical identifier, and close by listing the total number of tracks referenced. *.pls* files don't always reference the *full* path to each track, so if you move a playlist to another directory, disk volume, or hard drive on the same system, it won't work. *.pls* files are often dished up by streaming MP3 SHOUTcast/icecast servers, since the *.pls* format allows for the distribution of meta-data.

When creating playlists on your own machine, you'll almost always want to stick with .M3U for maximum compatibility with a wide variety of MP3 players. If you do need to create a *.pls* file, you'll find the option in the "Save Files as Type" picklist in the file panel when you access your playlist's Save function.

* For "MPEG Layer 3 URL"; the URLs contained by these files can be either local files on the user's system or remote files "out there" on the Internet. The MIME type for M3U files is *audio/x-mpegurl*.

Finally, there are a few applications that create their own custom playlist formats. A good example of this is the Apollo MP3 player for Windows (*apollo.mp3-2000.com*), which combines an MP3 player and sophisticated playlist manager into a single application. While Apollo offers a number of cool playlist features that others do not (such as ID3v2 support and the ability to create lists from arbitrary ID3 criteria), its playlists are only useful with Apollo itself.

Loading and manipulating playlists

Playlists can be loaded into your MP3 player in several ways, depending on the platform and the specific player. In most cases, the easiest way to load a playlist is to simply double-click its icon. If it's been properly associated with your MP3 player, it will launch the player with that list preloaded. Playlists can usually be dragged onto your MP3 players' interface, or loaded from the File → Load button in the player's playlist editor. If you're a command-line junkie, you can usually load a playlist by typing:

```
<path to player> <path to playlist>
```

WinAmp users can right-click a playlist file in Explorer and choose Enque from the Context menu. If a playlist is already playing, this one will start playing when the current one has finished. You can queue up a pile of playlists to play sequentially in this way.

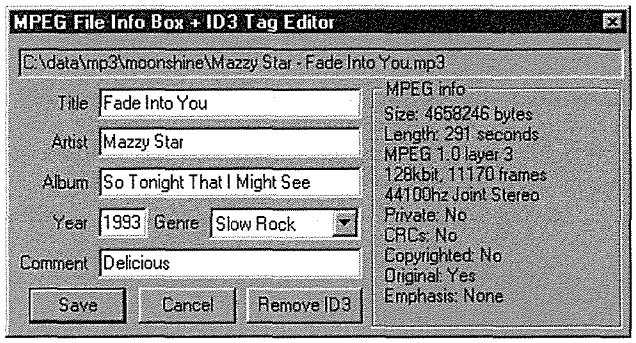
Most playlist editors include a number of functions designed to let you sort the order of tracks by song name, artist, length, and other criteria, though "Randomize" seems to be far and away the most popular means of "organizing" playlist tracks. If you're using Windows and are dropping files into a playlist editor, note that a peculiarity of Explorer could end up frustrating your attempts to get them to appear in same order you selected them in. However, if you select the last file first in Explorer, hold down the Shift key, and then highlight the first file; you can drag the whole block into your playlist editor and have the selections appear in the right order. In other words, the order in which you begin your selection in Explorer is the reverse of the order in which the selection is passed to other applications. You gotta love it.

Most playlist editors include a set of buttons (or a pull-down menu) separate from that of the MP3 player itself. WinAmp's playlist editor, for instance, includes buttons that will let you add entire directories or URLs to the playlist window, remove all selected files or all "dead" files (i.e., references to files which no longer exist at the specified location), sort your playlist by a variety of criteria, or save your playlist. To access the ID3 info for any file in WinAmp, select it in your playlist and tap Alt+3.

You can also generate HTML playlists directly from WinAmp's playlist editor. With the editor open, click Misc Options and select Generate HTML Playlist (or just tap Ctrl+Alt+G). The current list will be sent directly to your default browser. You can then pull down File → Save in the browser to store the resulting HTML document for future reference.

ID3 and Playlist Editors

There are seemingly infinite numbers of ways to create interfaces onto ID3 tag data (Figure 4-2), and we can't hope to cover a fraction of them in this book, but here are a few examples of the creative ways ID3 tags can be accessed, displayed, used, and edited, in various operating systems.



C:\data\mp3\moonshine\Mazzy Star - Fade Into You.mp3

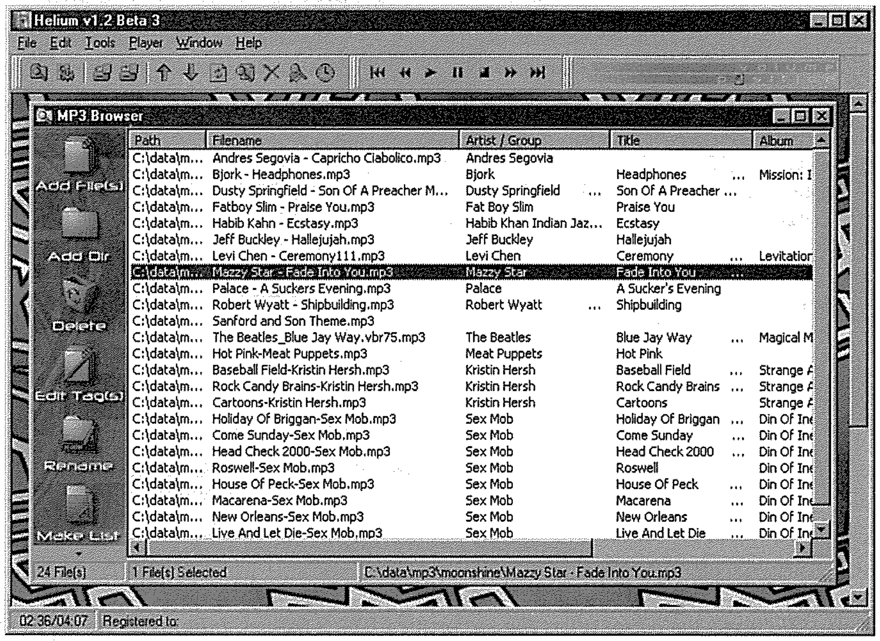
Title: Fade Into You
Artist: Mazzy Star
Album: So Tonight That I Might See
Year: 1993 Genre: Slow Rock
Comment: Delicious

MPEG info
Size: 4658246 bytes
Length: 291 seconds
MPEG 1.0 layer 3
128kbit, 11170 frames
44100hz Joint Stereo
Private: No
CRCs: No
Copyrighted: No
Original: Yes
Emphasis: None

Figure 4-2. A WinAmp's built-in ID3 tag editor

Windows

MP3 collections are becoming larger, and large projects lend themselves well to database-like solutions. Since the array of all tags in all of your MP3 files, considered together, pretty much amounts to a database, the real trick is in building a good interface onto that database, letting you view and edit ID3 tag data, rename files based on existing ID3 tags, see the collection of all ID3 tags in a database-like manner, search that database on custom criteria, and generate playlists from the results. If you want to do all of these things, you have two choices. You can scan your favorite software library for individual tools that serve one or more of these functions (and probably get it all done for free, but create a lot of busy work for yourself), or you can grab an all-in-one tool like Helium (*www.citymob.rit.se/helium/*). While most of the all-purpose tools will probably cost you a few bucks, they'll more than pay for themselves in the time they'll save you in the long run. Since, at this writing, Helium is one of the best all-purpose ID3/database/playlist tools available, and because it covers the general principles involved in performing all of these tasks, we'll focus on Helium in this section. Note, however, that there will likely be other, perhaps better, tools available for your operating system by the time you read this.



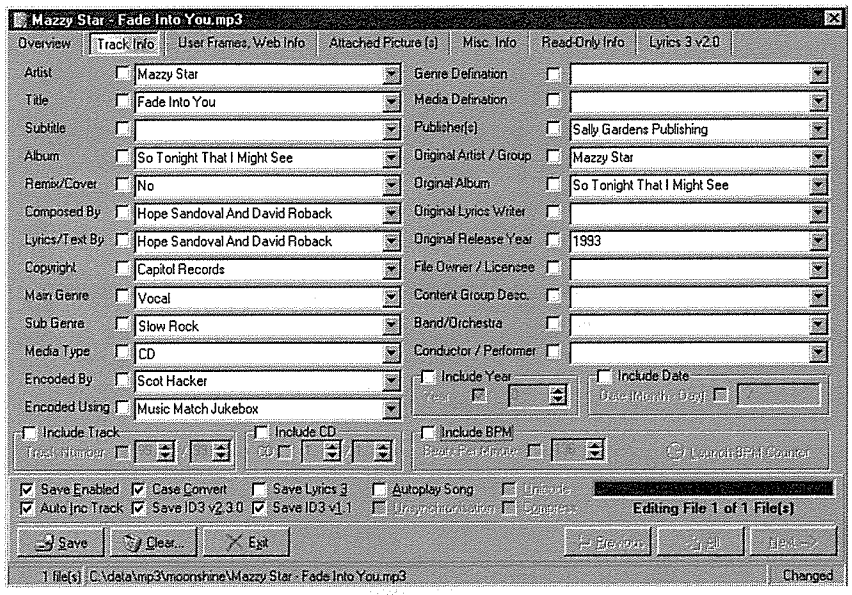
Path	Filename	Artist / Group	Title	Album
C:\data\m...	Andres Segovia - Capricho Clabolico.mp3	Andres Segovia		
C:\data\m...	Bjork - Headphones.mp3	Bjork	Headphones ...	Mission: I
C:\data\m...	Dusty Springfield - Son Of A Preacher M...	Dusty Springfield ...	Son Of A Preacher ...	
C:\data\m...	Fatboy Slim - Praise You.mp3	Fat Boy Slim	Praise You	
C:\data\m...	Habib Kahn - Ecstasy.mp3	Habib Khan Indian Jaz...	Ecstasy	
C:\data\m...	Jeff Buckley - Hallejujah.mp3	Jeff Buckley	Hallejujah	
C:\data\m...	Levi Chen - Ceremony111.mp3	Levi Chen	Ceremony ...	Levitatior
C:\data\m...	Mazzy Star - Fade Into You.mp3	Mazzy Star	Fade Into You ...	
C:\data\m...	Palace - A Suckers Evening.mp3	Palace	A Sucker's Evening	
C:\data\m...	Robert Wyatt - Shipbuilding.mp3	Robert Wyatt ...	Shipbuilding	
C:\data\m...	Sanford and Son Theme.mp3			
C:\data\m...	The Beatles_Blue Jay Way.vbr75.mp3	The Beatles	Blue Jay Way ...	Magical M
C:\data\m...	Hot Pink-Meat Puppets.mp3	Meat Puppets	Hot Pink	
C:\data\m...	Baseball Field-Kristin Hersh.mp3	Kristin Hersh	Baseball Field ...	Strange A
C:\data\m...	Rock Candy Brains-Kristin Hersh.mp3	Kristin Hersh	Rock Candy Brains ...	Strange A
C:\data\m...	Cartoons-Kristin Hersh.mp3	Kristin Hersh	Cartoons	Strange A
C:\data\m...	Holiday Of Briggan-Sex Mob.mp3	Sex Mob	Holiday Of Briggan ...	Din Of Ine
C:\data\m...	Come Sunday-Sex Mob.mp3	Sex Mob	Come Sunday ...	Din Of Ine
C:\data\m...	Head Check 2000-Sex Mob.mp3	Sex Mob	Head Check 2000 ...	Din Of Ine
C:\data\m...	Roswell-Sex Mob.mp3	Sex Mob	Roswell	Din Of Ine
C:\data\m...	House Of Peck-Sex Mob.mp3	Sex Mob	House Of Peck ...	Din Of Ine
C:\data\m...	Macarena-Sex Mob.mp3	Sex Mob	Macarena ...	Din Of Ine
C:\data\m...	New Orleans-Sex Mob.mp3	Sex Mob	New Orleans ...	Din Of Ine
C:\data\m...	Live And Let Die-Sex Mob.mp3	Sex Mob	Live And Let Die ...	Din Of Ine

Figure 4-3. The main Helium window

The main Helium window plays host to any number of what are known as "Browsers," where a browser is simply a table of MP3 files: a sort of fancy playlist (Figure 4-3). To create a new browser window:

1. Double-click on the Helium background. Point the empty browser to an MP3 directory or collection of files by clicking the Add Files or Add Dir buttons on the left.
2. All MP3 files in that directory and its subdirectories will appear in a table, along with any ID3 tags stored in those files. If the referenced files don't include MP3 tags and you want to give them some, select those files in the browser and pull down Tools → Edit Tags (or tap Ctrl+E).

3. An ID3v2 tag editor will appear for the first selected file, offering tons of fields, picklists, and text boxes into which you can type or paste lyrics, select genres, and so on (Figure 4-4).
4. When finished editing a file's tags, click Save, then Next to move to the next file in your selection.



Artist	Mazzy Star
Title	Fade Into You
Subtitle	
Album	So Tonight That I Might See
Remix/Cover	No
Composed By	Hope Sandoval And David Roback
Lyrics/Text By	Hope Sandoval And David Roback
Copyright	Capitol Records
Main Genre	Vocal
Sub Genre	Slow Rock
Media Type	CD
Encoded By	Scot Hacker
Encoded Using	Music Match Jukebox
Genre Defination	
Media Defination	
Publisher(s)	Sally Gardens Publishing
Original Artist / Group	Mazzy Star
Orginal Album	So Tonight That I Might See
Original Lyrics Writer	
Original Release Year	1993
File Owner / Licensee	
Content Group Desc.	
Band/Orchestra	
Conductor / Performer	

☐ Include Year ☐ Include Date ☐ Include Track ☐ Include CD ☐ Include BPM

☑ Save Enabled ☑ Case Convert ☐ Save Lyrics 3 ☐ Autoplay Song
☑ Auto Inc Track ☑ Save ID3 v2.3.0 ☑ Save ID3 v1.1

Editing File 1 of 1 File(s)

1 file(s) C:\data\mp3\moonshine\Mazzy Star - Fade Into You.mp3 Changed

Figure 4-4. Helium's tag editor offers the majority of ID3v2 frames; buttons along the top let you navigate amongst major classes of allowable ID3v2 data

Helium is also capable of renaming files singly or in batches, based on the contents of their ID3 tags (though the version we tested did not offer special character or blank space substitution, which we feel should be a mandatory feature). To access the renaming function, select a file or files and pull down Tools → Rename Files. As with other ID3 renaming tools covered in this section, you have partial control over the format of the filenames that will be created, via a template function accessed from Helium's Output Editor. Conveniently, Helium lets you store as many naming templates as you like. A couple are built in, but you'll probably want to create your own as well. Click the New button, and then the Tag Help button to see a list of all allowable field strings. Artist/Group is represented by the symbol *%00*, Title by *01%*, Album by *05%*, and so on. So if you've got a file called *file1.mp3* and its ID3 tags specify that this is "Blue Jay Way" from the Beatles' *Magical Mystery Tour*, you could set up a template of the form:

```
%00%05%01
```

Renaming your file with this template selected will result in a file called:

```
BeatlesMagical Mystery TourBlue Jay Way.mp3
```

That, however, is probably not what you want. To add separators between the fields, enter them in the Prefix and Suffix fields.

Helium is capable of exporting lists in two ways. Before you do, however, you'll probably want to use some of Helium's filtering and sorting capabilities—here's where the real power of a database begins to come to light. Let's say you've got hundreds of folders full of MP3 files scattered around on your hard drive, and you want to make a playlist of all songs with their Genre tag set to "Booty Bass." If you haven't already assigned Genres to your tracks, use the Tag Editor to do so. You'll also need to make sure the Genre column is showing in the Browser table—you can enable this from Helium's preferences panel. Now pull down Edit → Search, type in "booty," make sure the "Copy matches to new window" checkbox is selected and click Search. All songs in the main browser window that are tagged as Booty Bass songs will appear in a new Browser. Now all you have to do is pull down File → Save Playlist, select either the *.M3U* or *.pls* format, and save. Bingo—instant custom playlist.

In addition to playlists, Helium is capable of generating HTML or text documents from its browsers, so you can keep documents on your desktop that list your entire collection of MP3 files, or any custom subset thereof. Select a set of files or create a new browser containing customized search results, and click the List button. The output format of Helium's list generator can be completely customized by specifying HTML or other header data, and then inserting ID3 tag symbols into the Main List section of the List Generator.

Because of the inherent power of database-driven tools, you'll find that applications like Helium become increasingly indispensable as your MP3 collection grows larger.

One-click master list

How would you like to be able to double-click a single icon on your Desktop, have it scan your system for MP3 files, generate a master playlist, and launch that playlist into WinAmp or another MP3 player? A batch file (that's DOS-ese for shell script) like the following one can make this easy. To do this, you'll need to invoke the ancient masters of your early DOS training.

Generating a playlist from the DOS command line (i.e., DOS prompt) is a matter of constructing a command that will find all files ending in *.mp3* in a specified directory and all of its subdirectories, that will not spew out all kinds of extra data you don't need, and that won't list directory names themselves. Here's what you want:

```
dir c:\data\mp3\*.mp3 /s/b/a-d > c:\data\playlists\thislist.m3u
```

If you scatter MP3 files all over your system (which wouldn't be very good housekeeping), you can change the initial path to simply *C:\*. In case you're curious about those switches, */s* tells *dir* to recurse down through subdirectories. */b* tells *dir* to output only a simple list of pathnames, without file sizes, or foreshortened filenames. */a* tells *dir* to handle special attributes, as specified in its subsequent arguments. *d* is one of the attributes */a* can handle, and since we precede it with a - instead of *:*, we're telling *dir* to ignore directory names themselves (directory names in playlists will simply be ignored by most players, but they're still messy). The > symbol is a redirect, and tells DOS to dump the output of the command to a new file. Of course you'll need to substitute in the path to your collection of MP3 files, as well as the preferred path to and name of your generated playlist file.

If you want to do this sort of thing in a batch file, so you can simply double-click an icon on your desktop to have the whole thing run automatically, you'll want to create something like this:

```
@echo off
echo.
echo Creating playlist from d:\data\mp3 ...
:: Remember to edit the paths below to match
:: those on your machine! The second line is optional,
:: and shows how you can have this batch file scan through
:: multiple drives and directories and append the results
:: to the same playlist file. Note that if any of your directory
:: or filenames are longer than 8.3 you'll need to surround
:: them in quotes so DOS doesn't get confused.
dir d:\data\mp3\*.mp3 /s/b/a-d > "d:\data\playlists\alltracks.m3u"
dir e:\hold\music\mp3\*.mp3 /s/b/a-d >> "d:\data\playlists\alltracks.m3u"
echo.
echo Launching the new list in WinAmp...
:: The next line can be the the full path to any MP3 player
:: on your system that accepts a playlist as a command line
:: argument. WinAmp handles this just fine.
start "C:\Program Files\Winamp\winamp.exe" "d:\data\playlists\alltracks.m3u"
exit
```

To test your new system, double-click the batch file in Explorer. Did it work? Good. Unfortunately, you will have noticed that a DOS shell is launched automatically when you run the batch file, and doesn't close itself when you close WinAmp. The trick to fixing this annoying behavior is to click the small Properties icon in the DOS shell's menu bar and check the "Close on exit" checkbox (you may also want to select "Run: minimized" to keep the DOS window from flashing briefly on screen). Close the Properties panel, and a *.PIF* file will be created in the same directory as the batch file, appearing as a standard Shortcut file. Drag this shortcut to your Desktop or wherever, and from now on you can just click that to launch your script. The DOS shell will close as soon as WinAmp launches.

Do it with Perl

Search the Internet and you'll find tons of homebrew playlist solutions created by ordinary people, using free tools. One of the most common and useful such solutions is the Perl-based playlist creator, such as the one created by Patrick Hearon and available at *www.owlnet.rice.edu/~patrickh/playlist.html*. Point the script at a directory tree and it will "walk" through its contents, taking note of any MP3 files it finds in the path. When finished, it will generate either a plain text *.M3U* playlist file, suitable for use with any MP3 player, or a collection of HTML files, one in each directory, each listing the contents of that directory. Because each file automatically links to the others, you can browse through your MP3 collection easily, or post the HTML documents on a web site for others to peruse.

Perl junkies may also want to search CPAN (*www.perl.com/CPAN/*), a.k.a. the Comprehensive Perl Archive Network, for all sorts of useful Perl modules. At this writing, MP3.com was in the process of contributing some of their own modules into the CPAN database, including modules to give Perl built-in ID3v2 and MPEG Layer 3 info support (a less complete module already exists).

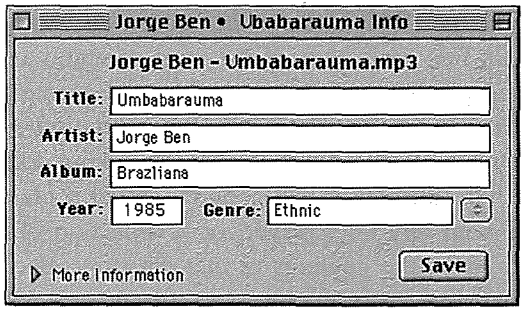
Because Perl is available for nearly every platform known to humankind, these scripts can be easily tweaked to work with your platform (though all you'll have to change are the magic cookies pointing to the Perl interpreter, and the path style used on your system).

Mac OS

Most of the popular MP3 players for the Mac include built-in ID3 tag editors, and usage is straightforward, as you might imagine. In MACAST, for example, make sure the playlist is showing, select a track, and click the Info button. Fill in or edit the fields as desired, then click Save to save your changes directly into the file, as shown in Figure 4-5. At this writing, MACAST supports only ID3v1 tags, though ID3v2 support may be available by the time you read this.*

Do not attempt to edit ID3 tags in a currently playing track. MACAST will issue a warning and offer to move to the next track in order to make changes to the current track, but in our experience this froze the Mac.

* In fact, a Lyrics field was available in the ID3 tag panel in the version of MACAST we looked at, and lyrics are officially a part of the ID3v2 spec. However, ID3v2 offers a much broader array of options than just lyrics, so we consider this to be only a partial implementation of ID3v2 support.


Jorge Ben • Ubabarauma Info

Jorge Ben - Umbabarauma.mp3

Title: Umbabarauma

Artist: Jorge Ben

Album: Brazliana

Year: 1985 Genre: Ethnic

Figure 4-5. A MACAST window's Info button produces fields to fill in or edit; just fill in the fields and close the MATE window to save your changes

More powerful than the built-in tag editors is a piece of freeware called mp3tool, which will scan a specified folder for MPEG files and let you edit ID3 tags in any files it finds along the way. While mp3tool won't create playlists from the files it finds, it offer very straightforward tag-editing capabilities. Search your favorite Mac OS software library for mp3tool.

Since Mac OS doesn't have a command-line environment, it's not possible to create scripts that perform similar functions to the DOS batch file above or the Unix shell script below. However, if you're familiar with AppleScript, it should not be difficult to store references to your MP3 files in a commercial database (such as FileMaker Pro) or spreadsheet and extract the results of database searches to a plain text file. Alternatively, you can use the Finder's Find panel to scan your system for all MP3 files, then drag files in the resulting list onto the Desktop or into a folder. From there, you can drag files directly into your MP3 player. However, because you can't drag files directly out of the Find panel and into a player, this process essentially means moving your MP3 files around, which is probably not what you want to do. However, you *can* drag files out of the Find panel and onto the MP3 player application icon. In the end, you're probably better off using the MP3 finding tools built into the players themselves.

Linux

The power of Unix-based operating systems is often at the command line, and ID3 handling is no exception. One of the most popular tools available for handling ID3 tags under Linux is id3ren (*tscnet.com/pages/badcrc/apps/id3ren/*), a multi-purpose tool that can not only read and write ID3 tags, but can also rename files based on the ID3 tags those files contain. While id3ren is most popular under Linux, note that BeOS and Windows ports of id3ren do exist, and can be found at your favorite software library. The real power of a tool like this is in its ability to be embedded in shell scripts and run over batches of files at once.

To see the tags currently attached to `filename.mp3`, type

```
id3ren -showtag filename.mp3
```

To rename a file with the default (built-in) naming template, just type:

```
id3ren filename.mp3
```

If the file's Artist tag was "Beatles" and the Title tag was "Taxman," `filename.mp3` will be instantly renamed to `[Beatles].[Taxman].mp3`. Don't ask me where that goofy default format comes from, but fortunately, you can easily change this by means of the `-template` argument. Templates in id3ren are a means of defining the naming structure of the files to be renamed, where the following identifiers are available.

Tag	Identifier
%a	Artist name
%c	Comment
%s	Song name
%t	Album title
%y	Year
%g	Genre

So, for example, if your file's Artist tag is set to "Beatles" and the Title tag is set to "Taxman," you can have the file renamed as *Beatles--Taxman.mp3* with the following command:

```
id3ren -template="%a--%s.mp3" filename.mp3
```

If you always want to use the same template, and don't want to have to specify it every time, create a text file called *.id3renrc* and place it in your home directory (see id3ren's documentation for more on allowable locations). In this file, specify the same template parameters, but leave out the quotes. id3ren will now always use your preferred template, unless you use the `-nocfg` flag in your command. Windows users will need to replace the % symbols with $ in the template examples here.

The simplest way to add to or change the tags associated with a given file is to type:

```
id3ren -tag filename.mp3
```

You'll be prompted to enter a string for each of the possible tags. Alternatively, you can add multiple tags at once with a single command, like this:

```
id3ren -tag -song "Mink_Shmink" -artist "Eartha_Kitt" filename.mp3
```

You'll then only be prompted for any tags you don't enter in the command (including `-comment`, `-genre`, and `-year`). If you don't want to be prompted for other tags, add the `-edit` argument. If all you care about are the Artist and Title tags, and don't want to be prompted for others, use the `-quick` flag. If you don't want the file to be renamed, and only want to edit tags, use `-tagonly`. To simply remove all tags, use `id3ren -striptag filename.mp3`. You get the idea. There are many more options available, but they all follow this basic model—just use `id3ren -help` to see them all.

The real power of id3ren, of course, comes into play when dealing with batches of files at once. In any of the example commands, wildcards or variable names can be used in place of `filename.mp3`. Thus, you can easily run through an entire directory of files at once. Since a directory might very well represent an entire album, you'll probably want to specify the artist and album names in the initial command, so you don't have to type them in over and over again. For example, let's say you just ripped and encoded all of *Magical Mystery Tour*, but the MP3 files you've created have been given names like *Track1.mp3*, *Track2.mp3*, and so on. The following command will give the files appropriate names, add the Artist and Title tags, and prompt you only for the song titles, skipping the year and comments:

```
id3ren -tag -artist "Beatles" -album "Magical Mystery Tour" -nogenre \
-nocomment -noyear -template="%a-%s.mp3" *.mp3
```

id3ren will prompt you for the name of each song in the current directory, write the song, album, and artist tags to each file, and rename each song as *Beatles-Taxman.mp3* and so on. If you don't mind being prompted for the genre tag with each song as well, you could use the `-quick` flag for a somewhat shorter command.

The following script can be used to sift through all of your MP3 files for just those files whose ID3 tags match criteria you specify. This sample script looks for matches on the Artist tag, though you can tweak it to search on the Genre tag, for example, "find me all songs that are filed under the Genre ('Gangsta'). Further adjustments could yield a script that searches on two criteria at once ("all Soul tracks from 1971"), or just about anything else you can imagine:

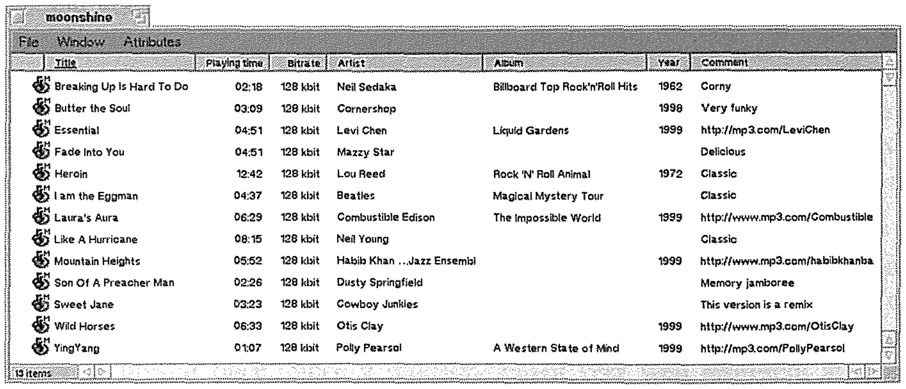
```
#!/bin/sh

# Get the artist name passed in from the command line
Artist="$*"

# Edit the next line to equal the full path to your main MP3 directory.
# To search your whole system, you could even make it " / ",
# But the more specific you are, the faster the script will be.
MP3Path=~/mp3
```

```
# Set this to the location in which you want your playlist files
# to appear. This universal HOME variable should work on all
# Unix-based systems, though you'll probably want
# to customize it.
PlayListDir=${HOME}

# Now we'll get a list of all MP3 files living in the MP3 hierarchy.
# The list will be saved to a temp file for further parsing.
find "$MP3Path" -name *.mp3 > $PlayListDir/.id3-tmp

# Now we're going to open that temp file and examine each track
# it references, trying to determine whether its Artist tag matches
# the one the user specified on the command line. We have to do
# a little fancy work with grep and sed to get just the info we need
# and nothing else.
{
while read ThisFile; do
    ThisArtist=$(id3ren -showtag "$ThisFile" | grep "Artist\:" | sed s/Artist\:\
//)
        if [ $ThisArtist = "$Artist" ]; then
            echo "$ThisFile"
        fi
done

# Input is the temp file, output is the final playlist
} < $PlayListDir/.id3-tmp > $PlayListDir/$Artist.m3u

# Remove the temp file
rm $PlayListDir/.id3-tmp
exit
```

BeOS

There are a number of tools for BeOS ID3 tag display and manipulation. All of the major MP3 players support normal ID3 display within their own interfaces, but BeOS also supports some ID3 manipulation not possible on other platforms. Because the Be file system supports "attributes" (data associated with a file that is not a part of the file itself), ID3 tags can be read out of MP3 files and displayed directly in the Tracker (the BeOS file manager), as shown in Figure 4-6. This provides an excellent database-like interface you can use to sort, query, and organize your MP3 tracks along user-selectable criteria.

One of the most useful tools you'll find out there is a collection of Tracker add-ons and applications by Jonas Sundström (see *www.be.com/beware/*). Install Tag2Attr and Attr2Tag in your */boot/home/config/add-ons/Tracker* folder, then right-click any MP3 file or collection of MP3 files and select Tag2Attr. All of the selected files will be examined for embedded ID3 tags, and attributes will be written to the file system. You can now enable the MP3 attributes you want to view from the Tracker's Attributes menu, and edit your attributes directly in the Tracker. Because other operating systems don't recognize BeOS attributes, you'll want to make sure

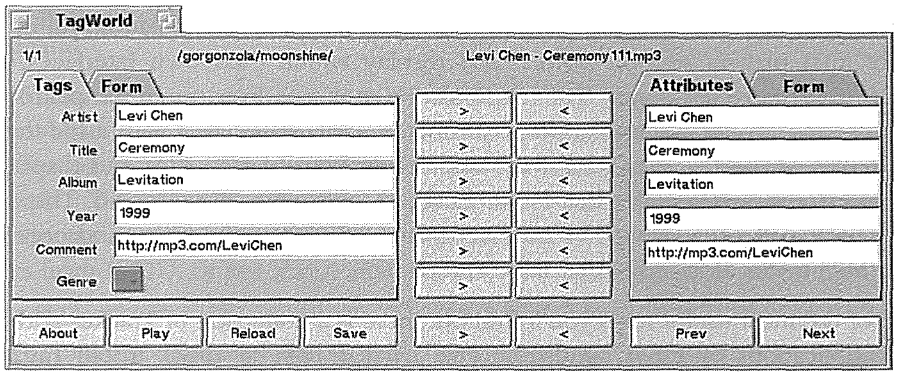


Title	Playing time	Bitrate	Artist	Album	Year	Comment
Breaking Up Is Hard To Do	02:18	128 kbit	Neil Sedaka	Billboard Top Rock'n'Roll Hits	1962	Corny
Butter the Soul	03:09	128 kbit	Cornershop		1998	Very funky
Essential	04:51	128 kbit	Levi Chen	Liquid Gardens	1999	http://mp3.com/LeviChen
Fade Into You	04:51	128 kbit	Mazzy Star			Delicious
Heroin	12:42	128 kbit	Lou Reed	Rock 'N' Roll Animal	1972	Classic
I am the Eggman	04:37	128 kbit	Beatles	Magical Mystery Tour		Classic
Laura's Aura	06:29	128 kbit	Combustible Edison	The Impossible World	1999	http://www.mp3.com/Combustible
Like A Hurricane	08:15	128 kbit	Neil Young			Classic
Mountain Heights	05:52	128 kbit	Habib Khan ...Jazz Ensembl		1999	http://www.mp3.com/habibkhanba
Son Of A Preacher Man	02:26	128 kbit	Dusty Springfield			Memory jamboree
Sweet Jane	03:23	128 kbit	Cowboy Junkies			This version is a remix
Wild Horses	06:33	128 kbit	Otis Clay		1999	http://www.mp3.com/OtisClay
YingYang	01:07	128 kbit	Polly Pearsol	A Western State of Mind	1999	http://mp3.com/PollyPearsol

Figure 4-6. BeOS lets you store ID3 tag info as filesystem "attributes," which can then be displayed and edited directly in the Tracker

these.attributes are written back into the file itself if you edit any of them (and plan to make them available to users of other OSes); hence, Attr2Tag works exactly the same, but in the opposite direction.

While this pair is excellent for batch jobs, you may find you want more control. Sundström is also the author of TagWorld, which is also a Tracker add-on, but lets you examine and edit the collection of tags for each file individually (see Figure 4-7).



1/1 /gorgonzola/moonshine/ Levi Chen - Ceremony111.mp3

	Tags	Attributes
Artist	Levi Chen	Levi Chen
Title	Ceremony	Ceremony
Album	Levitation	Levitation
Year	1999	1999
Comment	http://mp3.com/LeviChen	http://mp3.com/LeviChen

Figure 4-7. TagWorld lets you copy ID3 tags to attributes and back again

In order to query your BeOS system on ID3 data, you'll need to tell the filesystem to maintain an index of the ID3 attributes. Sundström also provides MP3 AI to help you create and manage the necessary indexes, as well as MP3 Flashlight, which makes finding MP3 files even easier than it is from the system's Find panel. Taken together, Sundström's collection of tools exploits native BeOS features to gain functionality similar to that found in Windows with tools like Helium (covered earlier in this chapter). Many BeOS encoding tools will write ID3 tags to attributes as they're encoding, and can even create custom folder layouts based on GUI templates. More on that in Chapter 5.

If you prefer to work from the bash shell, or if you'd like to script some of these behaviors, download a copy of Ari Pernick's id3attr from *www.be.com/beware/*. Once installed, you can copy ID3 tags to attributes by launching a Terminal and typing:

```
id3attr *
```

The author also keeps this alias in his *~/.profile*:

```
idren='id3ren -template="%a - %s.mp3" -space=" "'
```

By using the `id3attr *` command followed by the custom `idren` command, he can "database" and give ideal names to a batch of freshly downloaded MP3 files in seconds.

Once your MP3 files have attributes attached, there's almost no limit to the ways you can create custom playlists. For example, if you want a playlist consisting of all the MP3 files in your collection written in 1971, just do a normal system query (pull down the Be menu and select "Find") on "MP3 → Year contains 1971." Select All in the query results window and drag the selection into an open playlist window. Save the new playlist, and you're done. Of course, you can extend the same principle to create playlists of any complexity. For example, you could create a playlist in three seconds flat consisting of all Neil Sedaka tunes that have a Comment field including the word "smarmy." Pretty cool, huh?

Of course, if you want to get more hands-on from the command line, you can also use bash shell scripts like the one shown in the Linux section earlier in this chapter, though the BeOS query function gives you the same functionality directly in the GUI, but without the need to edit configuration files.

Skins: Dressing Up MP3 Players

As shown throughout this book, WinAmp and other MP3 players can be "costumed" with alternate interfaces called "skins." Because skins are merely collections of bitmap images with pre-specified names and sizes, skins are not tied specifically to WinAmp, or even to Windows. While WinAmp is by far the most common place to find WinAmp skins in use, you'll also find MP3 players for Mac OS, BeOS and Linux that wear WinAmp skins. In addition, there other MP3 players out there with even more radical interfaces (such as Sonique and K-Jöfol) that also wear skins. Note, however, that these players don't wear WinAmp skins; see their sites for availability of skins for those players.

How to Get WinAmp Skins

There are a number of large skin repositories on the Internet. The definitive collection is on WinAmp's own site, at *www.winamp.com/skins/*. There you'll find nearly 3,000 downloadable skins made by WinAmp users like yourself, categorized into groups like "Computer," "Game," "Stereo," "Anime," and even "Ugly" (and believe me, some of them deserve their place in that category). You'll probably find the highest quality skins in the "Best Skins" category, a group that has been selected by WinAmp employees for their finesse and professional look. There are (at this writing) three other notable skins sites out there: *www.1001winampskins.com* is a site run by WinAmp fans for other fans (not connected to NullSoft), while *www.skinz.org* and *www.customize.org* offer skins for dozens of products, not just WinAmp (though their WinAmp skins collection is smaller than that of the other two sites).

Each skin collection arrives as a single zip file, and becomes accessible to WinAmp simply by living in the skins directory specified in the WinAmp preferences (which by default is in *C:\PROGRAM FILES\WINAMP\SKINS*). If you have an older version of WinAmp, you'll need to unzip each collection into a subdirectory of the skins directory. For example, a skin archive called "Evo" might need to be decompressed in *C:\PROGRAM FILES\WINAMP\SKINS\EVO*. However, if you're using WinAmp 2.04 or higher, you don't need to unzip the archive at all, since WinAmp is smart enough to unarchive the files it needs on the fly (there is no noticeable performance penalty for doing this).

If you do choose to decompress your skins, note that many skin authors carelessly zip up their collections without also zipping the parent directory, which means you can easily overwrite skin components of the same name if you don't manually create an appropriate subdirectory first. If you use WinZip, a good way to do this is to right-click the zip archive and select "Extract To..." from the Context menu.

Once installed, skins can be activated by right-clicking anywhere in WinAmp's interface and choosing Skin Selector from the context menu (tapping Alt+S will also bring up the Skin Selector). A panel will appear listing all installed skins. Single-clicking an entry will cause WinAmp's interface to instantly adopt the new look. After donning a new skin, try enabling the WinAmp playlist and equalizer functions. Depending on how the author created the skin, these modules may or may not also take on the theme of the new skin. If skin components for these modules were not provided, they'll appear with the default WinAmp look and feel.

Please note that some skins are distributed as shareware, rather than as freeware. You should find a file called *readme.txt* in each skin's directory. If the skin is shareware, it will say so in this file. If you like the skin and decide to keep it, pay the author his or her due.

If you have multiple operating systems installed on your computer, there's no need to store the same skins collections multiple times. For example, since both BeOS and Linux can read Windows partitions, you may want to store all of your skins on your Windows partition and just create a symlink in your BeOS or Linux MP3 player directory called "skins" pointing to the location of your Windows skins directory. Of course you'll need to make sure your Windows partition is mounted automatically at boot time if you want this technique to work transparently.

WinAmp Easter Eggs

Like most heavily evolved software, WinAmp has a few "Easter Eggs" tucked away—non-obvious "features" that one might only stumble upon by accident... or if one read about them in a book. Try these:

- Right-click in WinAmp's interface and choose "NullSoft WinAmp" to bring up the About panel.
- Click the WinAmp tab to see the animated WinAmp character.
- With that animation running, hold down Ctrl+Alt+Shift and click on the copyright notice at the bottom of the panel. The character will be overlayed with a bitmap of WinAmp's original developer, Justin Frankel.

Older versions of WinAmp had additional Easter Eggs, though these seem to have disappeared after NullSoft was acquired by AOL.

Creating Your Own Skins

If you really want to call yourself a WinAmp pro, using other people's skins won't cut it—you've got to create a few of your own. While creating original skins can become tedious, time-consuming work—especially when you get into customizing regions and colors—the payoff can make the effort worth it. The only software you need is a good image editor, such as Adobe Photoshop, Paint Shop Pro, the GIMP (Linux and BeOS only), or something similar. You can even create skins in Microsoft Paint, though you'll probably find Paint underpowered for the job.

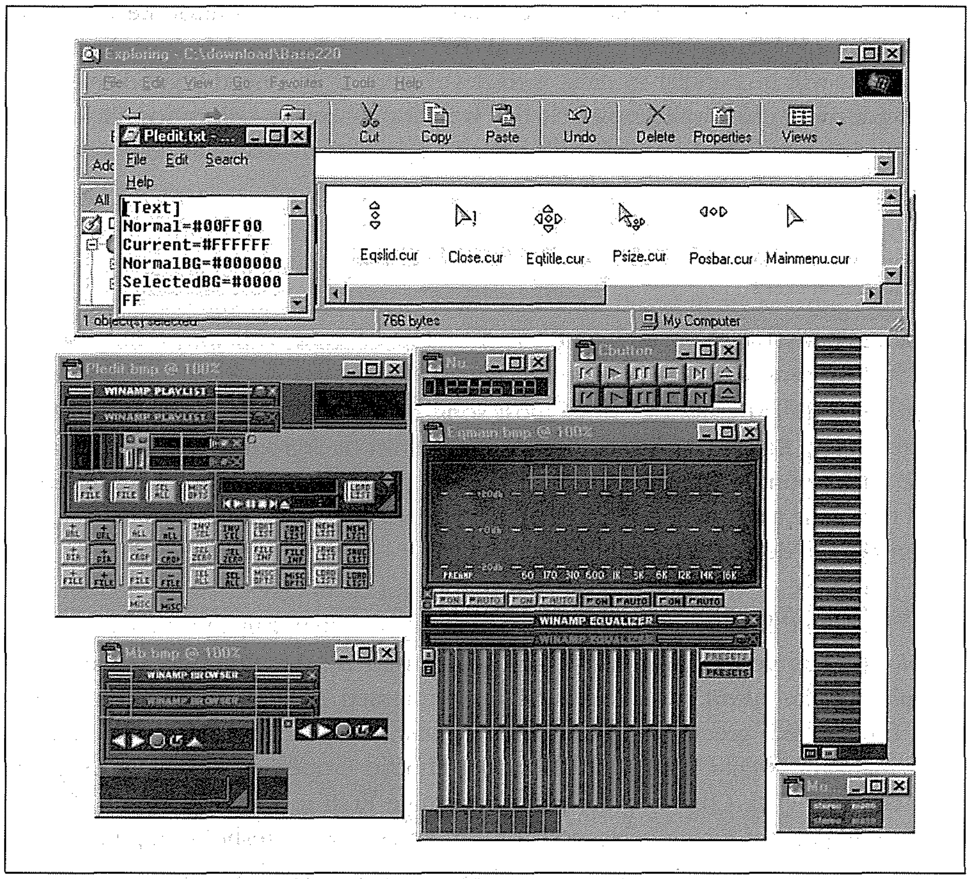
Next, you'll need to understand the purpose of each of the *.bmp* files in a skins directory. While you can use any skin as a starting point, the definitive collection of templates can be downloaded as *Base.<version>.zip* from *www.winamp.com/skins*. You'll also find a skin-creation tutorial at that site. Unpack *base.zip* and you'll find three type of files inside: *.bmp* files, which are the actual graphics that will comprise your skin's interface; *.cur* files, which are standard Windows cursor files to be activated when the user's mouse hovers over different parts of the interface (ignored when the skin is used by another operating system); and *.txt* files, which can be customized to specify the colors of non-graphical elements and transparency behaviors.

You do not need to edit—or even include—a version of every single file in this directory. The behavior of elements for any files you don't include will be supplied by WinAmp itself. In other words, the default WinAmp interface is always present behind your creation.

Skin graphics

Open the various *.bmp* files in your image editor, and the purpose of each will become immediately obvious—you'll see fragments of familiar interface elements in each one, while each file's purpose is also declared by its file name (Figure 4-8). For example, *volume.bmp* displays a column of colored horizontal glowing strips, representing the colors of the volume slider as you move it right and left. *Titlebar.bmp* likewise shows all possible incarnations of the WinAmp titlebar (e.g., as it appears when maximized, when in zoom mode, and with and without the visualizer enabled). All you have to do is edit these graphics to your liking, taking care to preserve their dimensions and division lines precisely. Because many of the elements, such as the fonts in *text.bmp*, may appear very, very small, you'll probably want to get familiar with your image editor's zoom (magnifying glass) function.

Because just creating new bitmaps for every possible interface element in WinAmp can be a huge job, most skin creators call it a day after finishing the graphics portion. If you really feel like gettin' jiggy wid' it though, you'll want to dig into the *.cur* and *.txt* collections as well.



```
[Text]
Normal=#00FF00
Current=#FFFFFF
NormalBG=#000000
SelectedBG=#0000
FF
```

Figure 4-8. Bitmaps, base configuration files, and Windows cursors, all with specific filenames that must not change

Cursors and control files

The *.cur* files used by WinAmp are standard Windows cursor files, and can be generated in any number of Windows shareware programs designed for the purpose. You may want to look into the excellent Microangelo (*www.impactsoft.com*) for some serious cursor customization action. You should be able to determine the purpose of each cursor file by its filename; for example, *eqslid.cur* is the cursor that goes into effect when the mouse is using the equalizer's sliders, while *volbar.cur* takes hold when adjusting WinAmp's volume.

Manipulating fonts and colors

The most technical aspect of skin creation (and again, this is totally optional) may be the configuration of specific aspects of WinAmp appearance and behavior from the text-based configuration files *region.txt*, *viscolor.txt*, and *pledit.txt*.

The easiest to edit of these files is *pledit.txt*, which lets you control the fonts and font colors used in the playlist panel. All values are expressed in hexadecimal, a format that will be familiar to most web developers. If you need a good reference of hexadecimal color values, search the web for "netscape colors" or "hexadecimal colors." You'll find both web pages and applications that will help you look these up easily. The format of *pledit.txt* is simply:

```
[Text]
Normal=#00FF00
Current=#FFFFFF
NormalBG=#000000
SelectedBG=#000080
Font=Arial
```

Where `Normal` is the font color of most song entries, `Current` is the color of the selected track, `NormalBG` is the color of the playlist background, and `SelectedBG` is the background color for the selected track. If you specify fonts here, keep in mind that your users may not have this font on their system. If the specified font is not found, the default font will be used.

Manipulating the visualizer

viscolor.txt lets you control the colors being used by the WinAmp visualization module. Unlike *pledit.txt*, colors here are expressed as RGB triplets (one value each for the red, green, and blue color values). You can use Windows' built-in color manager (right-click on the Desktop, choose Properties, select the Appearance tab, and click Colors → Other) to determine RGB triplets. There are 24 lines in this file, each of which corresponds to an audio level threshold. Experimenting with this file is probably the best way to get a feel for which entries take effect at which audio thresholds. If you want to keep track of your changes, you can succeed each line with a comment:

```
24,33,41, // This corresponds to mid-volume signals
```

Manipulating skin regions

Perhaps one of the most interesting interface controls you have as a skin developer is afforded by *region.txt*, which lets you use masks to "carve out" regions of WinAmp to be considered invisible. By using a system of X,Y coordinates, you can tell WinAmp only to draw specific sections of the interface, and to leave the rest transparent. The results can be dramatic. Working with *region.txt* is somewhat more complex than working with the other text files. The principle here is that you specify the shapes of polygons by telling WinAmp how many points each polygon has, followed by a series of X,Y coordinates for each polygon. Since WinAmp's dimensions are 275×116 pixels, the X coordinate values can range from 0 to 274, and the Y values can range from 0 to 115. In order to determine the exact coordinate points you need, you'll want to find the tool in your image editor that reports the exact coordinates beneath the mouse cursor. (In Photoshop, pull down Window → Show Info.) WinAmp simply draws all regions that fall within the polygons you've defined and leaves the rest invisible. Open *region.txt* in the base skin collection and you'll see lines like this:

```
;NumPoints=4
;PointList=0,1, 275,1, 275,14, 0,14
```

Any line beginning with a ; is ignored, so to enable any series of sample coordinates, just uncomment those lines, save, and load the skin to see the effect. Of course, you'll probably want to define more than one polygon, but for ease of development, just work on perfecting one polygon at a time (comment out your perfected polygon and start a new one). Once all polygons have been perfected, you'll need to conjoin them all into a single line, like this:

```
; NumPoints = 4, 4 ; In other words, define two squares
; PointList = 0,1, 275,1, 275,14, 0,14,   3,15, 272,15, 272,113, 3,113
; The space between the two sets makes the coordinate series easier to read.
```

See the comments in *region.txt* for further details. If you don't want any transparency in your skin, just delete *region.txt* or make sure all of its lines are commented out.

Mac users! MACAST supports a skin format that is incompatible with WinAmp skins. Fortunately, this also means that MACAST users are not limited to the size and shape constraints of WinAmp skins. If you do want to use WinAmp skins with MACAST, you can download a tool called SkinConverter from *www.solsticetechnologies.com/skinconverter/* that will do the job nicely. If you use SoundJam, you'll find a copy of SkinConverter in the SoundJam installation directory—just drag WinAmp skins onto the SkinConverter icon and they'll be transformed into SoundJam skins and placed automatically in SoundJam's skins folder, ready for use.

Plug-ins: Extending Your Reach

Just as the capabilities of programs like Adobe Photoshop or Netscape Navigator can be extended through the addition of third-party extensions called "plug-ins," so can many MP3 players. The beauty of the plug-in model is that the developers of the MP3 application get to have their product's capabilities extended indefinitely by the contributions of the larger developer community, and a cottage industry can be established for plug-in programmers. As a user, you'll find that your MP3 player may actually be far more capable than it originally appeared, thanks to these additions. While skins let you customize the look and feel of your MP3 player, plug-ins let you extend its actual functionality. Unlike skins, however, which are generic enough to work across multiple operating systems and players, plug-ins must be written specifically to work with each MP3 player. While you must have programming experience to write plug-ins, anyone can use them. Some are free, some are shareware, and some are available only by paying a fee up front.

Some MP3 players, like WinAmp, are built modularly, around the very concept of plug-ins. Because much of these programs' functionality is derived from the presence of certain crucial plug-ins, you should *not* delete the plug-ins that come with the WinAmp distribution. You may very well end up disabling your player's ability to export any sound at all.

There are a seemingly infinite variety of third-party plug-ins available, but they all fall into a few overarching categories:

Input plug-ins
: Allow your player to play audio file formats that are not built into the player itself, or to accept signals from external devices such as keyboards or stereo components.

Output plug-ins
: Allow your MP3 player to export signals to other audio file formats.

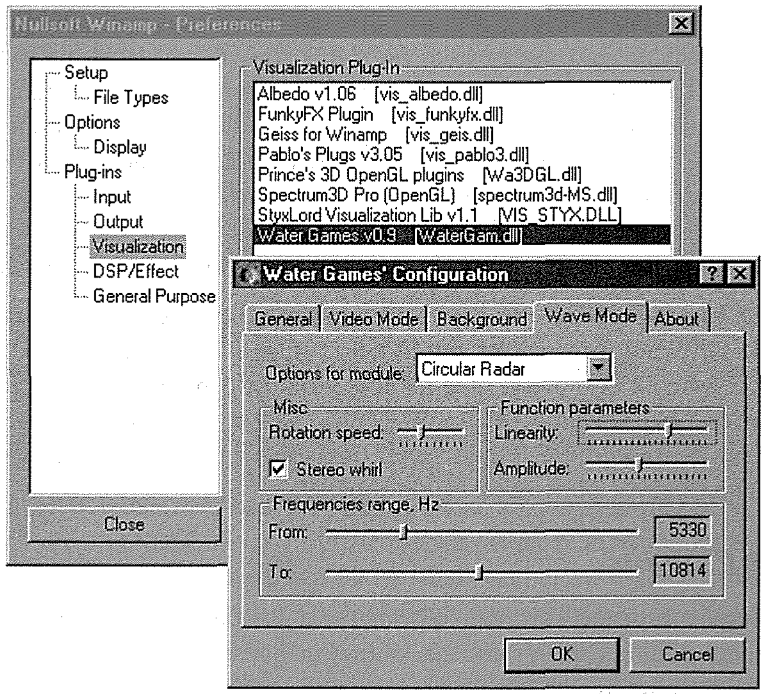
DSP/Effects plug-ins
: Let you apply any number of audio special effects, such as distortion, echo, reverb, and more.

Visualization plug-ins
: Perhaps the most popular of the lot, let you "see" your music in displays far more sophisticated and psychedelic than the simple visualizers built into most MP3 players.

Other plug-ins
: Let you do things that aren't easily categorized, such as handling incoming signals from universal remote control units.

The best place to find plug-ins for your MP3 player is usually at that MP3 player's web site. You'll also find a good cross-platform plug-in collection in the Software section at *www.mp3.com* and other popular MP3-related web sites. While the specifics of plug-in installation and configuration differ greatly from one player to another, you'll almost always need to install plug-ins in a subdirectory of the MP3 player's installation directory called "plug-ins" or "plugins." In most cases, you should *not* create a new subdirectory for each downloaded plug-in under the *plugins* directory (yes, this is inconsistent with the skins installation model). WinAmp plug-ins are stored in *dynamically linked libraries*, or *.DLLs*. To access or activate a given plug-in, try right-clicking your M3 player's interface or looking for a Plug-ins menu. In WinAmp, just tap Ctrl+P to bring up the Preferences panel and expand the Plug-ins section of the hierarchy. To configure any individual plug-in, double-click its entry in this panel, as shown in Figure 4-9. Because most plug-ins are actual programs that run in conjunction with the MP3 player itself, most of them sport their own configuration interface.

Before starting to download and install third-party plug-ins, you should spend some time experimenting with the collection of pre-installed plug-ins, just to familiarize yourself with the huge number of options already at your disposal.


Nullsoft Winamp - Preferences

- Setup
  - File Types
- Options
  - Display
- Plug-ins
  - Input
  - Output
  - Visualization
  - DSP/Effect
  - General Purpose

Visualization Plug-In

```
Albedo v1.06   [vis_albedo.dll]
FunkyFX Plugin   [vis_funkyfx.dll]
Geiss for Winamp   [vis_geis.dll]
Pablo's Plugs v3.05   [vis_pablo3.dll]
Prince's 3D OpenGL plugins   [Wa3DGL.dll]
Spectrum3D Pro (OpenGL)   [spectrum3d-MS.dll]
StyxLord Visualization Lib v1.1   [VIS_STYX.DLL]
Water Games v0.9   [WaterGam.dll]
```

Close

Water Games' Configuration

General | Video Mode | Background | Wave Mode | About

Options for module: Circular Radar

Misc
Rotation speed:
[✓] Stereo whirl

Function parameters
Linearity:
Amplitude:

Frequencies range, Hz
From: 5330
To: 10814

OK Cancel

Figure 4-9. Double-click any entry in WinAmp's Plug-ins preferences panel to launch a configuration panel specific to that plug-in

Without question, the greatest number of plug-ins are available for WinAmp; at this writing, there were more than 140 WinAmp plug-ins available at *www.winamp.com/plugins/*, with more being added on a regular basis. That site, by the way, also includes a category called "Best Plug-ins," containing those deemed by WinAmp's staff to be of the highest quality or usefulness. You may as well start there. Let's look at a few of the most popular WinAmp plug-ins available, keeping in mind that plug-ins for other players and other operating systems perform similar functions but aren't available in such large number or variety.

Input Plug-ins

This category includes plug-ins designed to let your MP3 player handle arbitrary file formats and signal types as input. You'll find input plug-ins that let WinAmp handle MIDI, MPEG, Waveform, and CD/Line input, for example. Double-click any of these for further options relating to the way these input streams are handled. Most important for MP3 players is, obviously, the MPEG input plug-in. Owners of older machines and crappy sound cards will find a number of options here that let you adjust the performance of the MPEG decoder to use fewer resources. All users will appreciate being able to customize the way song titles and artist names are displayed as they're read out of each file's ID3 tags. If you have a very slow or eternally busy hard drive, you may get better MP3 performance by setting the Full File Buffering size to a number higher than the default of 128 K (a buffer here is simply a space in program memory dedicated to holding audio data before it's processed; by using a larger buffer, you guard against the possibility of skipping audio due to the hard disk's read head not being able to gather data from disk fast enough to satisfy the player). Streaming Data Buffers will be covered in Chapter 8.

If you start looking around for third-party input plug-ins, you'll find enablers that let you play obscure formats like sound tracks ripped from video games, plug-ins that let you watch AVI movies, plug-ins that work in conjunction with radio tuner cards, listen to high-quality AAC or VQF files, Amiga SoundTracker files, or the ever-popular RealAudio file format. Each of these will come with its own documentation and unique preferences/settings panels.

Output Plug-ins

Output plug-ins function as a sort of bridge that sits between the MP3 player and its output, controlling exactly how that output manifests. For general playback, the basic WaveOut plug-in (for WinAmp) is the only one you'll need, and you probably won't need to mess with its default settings unless you've got an older sound card (such as SoundBlaster 16), or need to perform some fancy footwork, like reversing the left and right stereo channels.

Decoding vs. Playback

Throughout this book, you'll see the terms "decoder" and "player" used pretty much interchangeably. Technically speaking, anything that plays MP3 audio is a decoder, since it's basically running the codec in reverse, re-assembling intelligible audio from a pile of bits in accordance with the MPEG specification. However, the term "decoder" is sometimes used specifically to refer to any process that takes an MP3 bitstream and converts it back into an uncompressed PCM audio file such as WAV or AIFF. Whether the MP3 file is converted to audible signal or to an uncompressed file, the exact same processes are in use.

Converting MP3 to WAV files

One of the most useful alternative outputs offered by the Output Plug-ins collection is the NullSoft DiskWriter, which lets you transform your outgoing MP3 signal into WAV files on your hard drive, rather than to audible signal. All you have to do to enable this function is double-click its entry, select an output directory on your system, and click Close in the WinAmp preferences panel. Playing MP3 files will now result in them being written to hard disk as WAV files. You will *not* hear output from WinAmp while this is happening, since only one output plug-in can be active at a time, and selecting the Disk Writer plug-in disables the WaveOut plug-in by definition. Of course, the resulting WAV files will be 10–12 times larger than the MP3 files they originate from. Remember to change back to WaveOut when you're done exporting to disk! You'll probably use this option a lot if you decide to burn audio CDs from your MP3 collection.

Use Microsoft Audio optimizations

If you want your system to be able to play sounds from multiple sources at once, or if you want to launch multiple instances of WinAmp simultaneously, you'll need to take advantage of capabilities offered only in later versions of Windows, or installed with upgrades to the Windows MediaPlayer. The technology that allows this is called DirectSound, and WinAmp must be told specifically to take advantage of it. Select the DirectSound output plug-in rather than WaveOut if you want to mix multiple audio streams down into a single output, or if you just want to see what it's like to play a bunch of MP3s simultaneously. Keep in mind that Windows may still balk if you tax its audio system too hard, since the operating system is not inherently optimized for this kind of maximum audio throughput.

Cross-fading your files

There aren't a ton of third-party output plug-ins floating around. The most popular is Justin Frankel's Crossfading output plug-in, which lets you start one MP3 file playing while the previous one is on its way out, with the first one's volume going down while the next one comes up. In other words, a cross-fade plug-in can make your MP3 audio stream sound as if it were coming off a DJ's mixing bench. Cross-fade timing is controlled by adjusting the size of the buffer, measured in milliseconds (i.e., a 5,000 ms buffer will give you a 5-second cross-fade). Note that using a larger buffer will help prevent skips on slower machines, but consume more memory.

BeOS users: You can attach cross-fade settings to attributes of individual MP3 files, or to entire playlists at once, without needing a plug-in. Just click and hold on any currently playing track and select "Crossfade settings" from the context menu to adjust the settings for that file. If a playlist is currently being played, just select all tracks in the list and make your fade-out/fade-in changes, then click "Apply to Selection."

Linux users: If you use Xmms and have a multiple-CPU machine, try downloading the XAudio output plug-in from *www.xaudio.com* and using that instead of Xmms's default output—you'll feel the difference in performance!

DSP/Effects Plug-ins

This sophisticated (and often complex) breed of plug-in lets you apply digital signal processing and other effects to the audio signal. While you may enjoy simply tinkering with some of the amazing effects you can produce with these, many of them are geared specifically toward the more high-end user: radio stations, DJs, and sticklers for absolute quality and customization.

Virtual DJ

One of the most amazing DSP plug-ins available is Leif Claesson's PitchFork, which will literally let you treat your MP3 collection as if it were a pile of vinyl records, to be cued up, nudged, and synchronized. Pitchfork's controls are rather complex, and you'll definitely want to step through the included tutorial if you want to get the most out of the program. Of course, you'll need to be running at least two instances of WinAmp simultaneously, which means you'll need to be using the DirectSound output plug-in. If you really want to do it right, you'll want to have two sound cards installed in your computer, both of them running their line outs into an external mixing board. Needless to say, PitchFork's hardware requirements are a little beefier than your usual plug-in (but not overwhelming)—you'll need at least a Pentium 200 and 64 MB of RAM.

If you're serious about DJing with MP3 files and want something more than a plug-in, check out virtual dj at *www.virtualdj.com*, a complete playlist organization/automation system that handles MP3 files (along with other formats). Linux users may want to check out DigitalDJ, at *www.nostatic.org/ddj/*. BeOS disk jockeys use FinalScratch (*www.n2it.net/finalscratch/*), which uses physical turntables and special records containing nothing but microsecond timing marks—the system depends on Be's extremely low audio hardware latencies to let DJs "scratch" MP3 files in SoundPlay from the helm of actual turntables.

Normalizing playback

More down-to-earth is another of Claesson's creations, called AudioStocker. This gem provides a solution to the annoyance of having to reach for the volume control between tracks to compensate for the fact that MP3 files often demonstrate substantially different volume levels. Since most MP3 files are encoded digitally, without signal tweaking by the person doing the encoding, this artifact is due to the fact that different audio engineers and different recording labels tend to record albums at different levels to begin with, for whatever reason. You've probably noticed this effect before on your home stereo, but with MP3, where you're often jumping quickly between tracks extracted from different albums, the effect becomes much more noticeable.

The general term applied to the process of making multiple volume levels peak at similar thresholds is called *normalization*, and this is exactly what AudioStocker does. AudioStocker's controls consist of a set of push buttons that let you control the thresholds of the plug-in's intervention in the output signal. Tell the plug-in how much it's allowed to amplify a tune, how much frequency balance (equalization) can be applied, and a degree of audio compression (not to be confused with file compression) that can be applied. With a little tweaking, you should be able to get all of your tunes coming out of WinAmp at virtually the same perceived volume.

However, it's worth pointing out that this normalizing process is bound to result in some degradation of quality. Normalizing usually relies on the fact that the whole track can be examined in advance in order to establish peak levels. Normalizing on-the-fly, though, doesn't enjoy the luxury of a complete pass—it has to work by trimming the peak volumes and boosting the volume of quiet sections. The final output will therefore differ somewhat from the original by definition. While this is unavoidable in some situations, keep in mind that you'll always be better off doing your normalizing during the encoding process (from a quality standpoint).

In the future, it may become possible to store relative volume levels in a file's ID3v2 tags. Players would then be able to examine that data and adjust volume dynamically for each track. No known players were capable of working with ID3v2 tags in this way at this writing.

Other DSP/Effects

Take a look around and you'll find a number of simpler DSP/Effects plug-ins available for many MP3 players, including modules that let you bend the pitch of the current signal up or down (without speeding up or slowing down the track), modules that let you apply standard effects like chorus, reverb, flange, or echo, and even modules that let you "stack" multiple plug-ins on top of one another, so that you can create cumulative effects.

While not technically a plug-in, WinAmp does offer a cool little output feature that will let you generate pure tones for testing purposes. Turn down your system's volume control so you don't accidentally hurt your ears or speakers, then tap Ctrl+L (for Open Location) in WinAmp and enter *tone://1000.* WinAmp will play a 1kHz tone until you press the Stop button. Of course, you can substitute any frequency you need. If you'd like to save the tone to disk as a WAV file, enable the DiskWriter plug-in in WinAmp's Output Plug-ins section.

Visualization Plug-ins

This variety of plug-in is probably the most popular because they're the most fun. If you thought WinAmp's built-in visualization module was cool, you ain't seen nothin' yet. B-b-b-b-baby, you ain't seen nothin' yet! These modules analyze the signal flowing out of the player and into the system and run spectrum analyses, mapping changes in frequency and volume to rules that control a visual display. Because there are infinite numbers of ways to construct these rules, the appearance of these visualizers is virtually unlimited. Some of them are soothing, offering effects as subtle as drops of waters landing in a pool, while others can only be described as seizure-inducing, with wild psychedelic light shows emanating from every delta point. While some visualization plug-ins run in small or resizeable windows, others will optionally run full screen, which is perfect for use at parties, or late-night, low-light zombie sessions.

The screenshots shown in Figure 4-10 are of visualization plug-ins for WinAmp, but there are visualizers available for many different MP3 players, and for virtually all operating systems. These screenshots don't do justice to the experience of seeing these for yourself—you've just got to download a few and play with them. While all visualizers come with good default settings, most of them also include settings or preferences panels that let you customize the colors, responsiveness thresholds, and visualization forms. Dig around, turn the knobs, push the buttons, check and uncheck the checkboxes, and drag the sliders around. If you're a customize-a-holic, be prepared to set aside several hours of twiddle time.

Some visualization plug-ins are very CPU-intensive, and can have a noticeable effect on system performance. You probably won't want to run visualizers as a matter of course, reserving them for times when you're not trying to do other tasks in the background while listing to MP3 files. Not to mention the fact that many of these are wild enough to completely distract you from getting any real work done while they're running!

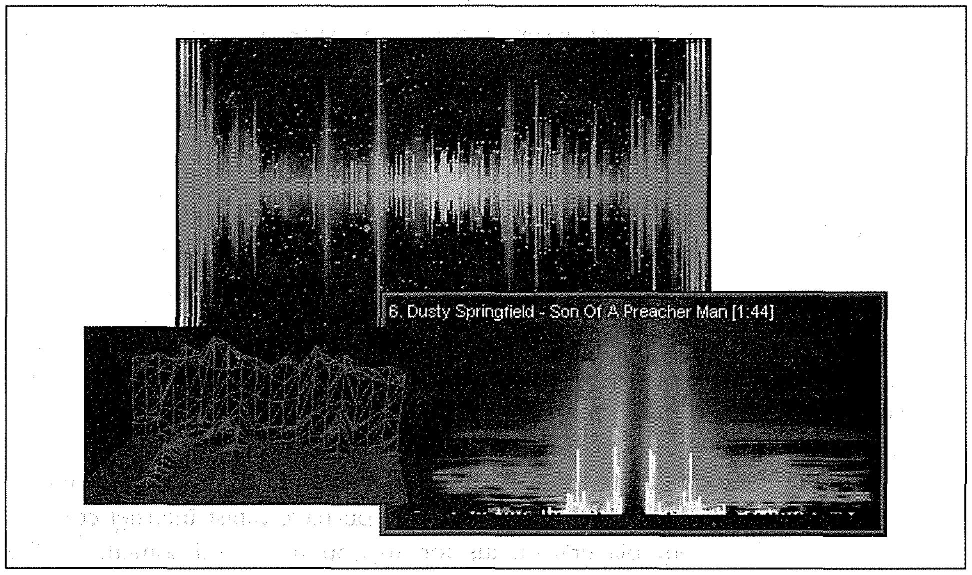


Figure 4-10. A few of the many visulization plug-ins available for WinAmp

General Plug-ins

The category "general" is really just a nice way of saying "miscellaneous," referring to plug-ins that do odd-jobs around the house. Did you know, for instance, that you can control WinAmp with a standard, $25 universal remote control? You'll need to purchase a separate infrared receiver to connect to your computer, and then install a corresponding IR plug-in to process the invisible signals it generates. This is the kind of connectivity that people are talking about when they refer to "convergence"—the computer becoming the center of the home infotainment system. See Ampapod (*www.v.nu/core/ampapod/*) or Irman (*www.evation.com/irman/*) for more information.

Other general plug-ins let you output your current playlist to an HTML file for use with SHOUTcast or icecast, some let you operate WinAmp controls from the Windows TaskBar tray, or even control WinAmp over a local area network (LAN).

Listening to MP3 Streams

MP3 files don't have to be downloaded to your hard drive, necessarily. In many cases, you'll be able to play them back in your favorite MP3 player directly from the server on which they're located. As you'll discover in Chapter 8, there are several ways in which MP3 webmasters can dish up files for "streaming." As an end user, you won't need to worry much about the particular streaming technique in use, though it's interesting to know the difference. In some cases, you may have to tweak a few settings in your browser or player to make sure streamed files are handled by your operating system properly.

Types of Streaming

There are two primary ways in which MP3 files can be streamed to users without being downloaded: MP3-on-demand and MP3 broadcast.

MP3-on-demand

In this form of streaming, control of the download is in the hands of the MP3 player, rather than the browser. Because this capability is built into most MP3 players, users can choose at any time to listen to an MP3 file directly from a web server, without saving it to their hard drives first. Of course, this assumes that the user has sufficient bandwidth to listen to the file in real time without it skipping or halting, but we'll get to bandwidth issues later. If you have a fast Internet connection, look around in your player's menus for an option labeled something like "Open Location" or "Play URL" and enter the URL of any MP3 file on the web. The easiest way to get this information is to right-click a link to an MP3 file in your browser and choose "Copy Link Location" from the context menu, then paste the URL into the Open Location dialog in your player.

In addition, MP3-on-demand can be forced by the webmaster, so that clicking a link normally will cause MP3 files to be pulled down by the player and played directly, rather than saved to hard drive as with a normal download. To do this, the webmaster creates an "M3U" (MPEG URL) playlist file, which is a plain text document containing the full URL to an MP3 file (or list of files) on a web server. Because the text file is tiny, the browser can download the M3U file to the user's hard drive nearly instantaneously. The web server sending the M3U file should (if it's configured correctly) dish it up with the MIME type *audio/x-mpegurl*. This MIME type should, in turn, be associated in the user's browser or operating system with a preferred MP3 player capable of handling MP3 streams.

Once the M3U file is downloaded, it's launched in the preferred MP3 player, which reads URLs out of the file and takes over control of the actual download.

Note the difference here: when you download an MP3 file normally, the browser itself handles the entire download, and users have to then launch the MP3 file in an MP3 player manually. The MP3 file is stored on the user's hard drive for future use. With MP3-on-demand, the MP3 player handles the download, not the browser. The MP3 player plays the file *as* it's being downloaded, not later on. And unlike a standard download, the MP3 file is not present on the user's system after they've finished listening to the track.

The advantages of using the MP3-on-demand technique are:

- The user does not have to wait for the download to complete before beginning to hear music.
- The user has more control over playback than with real streaming (e.g., the user can skip around between songs or fast-forward through songs at will).
- The webmaster has a degree of protection against MP3 files being stored permanently on the user's system.
- The publisher or webmaster does not have to set up any special MP3 serving software by having the MP3 player "suck" the file down. A plain vanilla web server running on any operating system is capable of serving up MP3-on-demand.

Because MP3-on-demand offers so much flexibility to both the webmaster and to the user, it may eventually become more popular than true MP3 streaming if and when all of us have lots of bandwidth. As long as our bandwidth is limited, however, true streaming solutions will continue to outweigh on-demand systems in popularity.

An important aspect of the MP3-on-demand technique is that it's "asynchronous," or outside of time. In other words, it doesn't matter what time of day the user accesses the file—he'll hear it from the beginning. This is very different from TV, radio, or MP3 broadcast, where you get whatever is being broadcast at the moment in time when you tune in to the channel. For this reason, the MP3-on-demand technique is also sometimes referred to as "pseudo-streaming."

A good example of MP3-on-demand can be found at MP3.com. Access any artist's page and access one of the Hi-Fi or Low-Fi links. Rather than being prompted for a download location, your favorite MP3 player will be launched and the file should start playing immediately.

See Chapter 8 to learn how to set up your server for MP3-on-demand.

MP3 broadcast

In contrast to pseudo-streaming, MP3 broadcasting (or real streaming), is "synchronous," and thus more akin to TV and radio broadcasting. In this case, the user tunes in to a channel or station which is playing an MP3-encoded bitstream much like a radio station, sometimes complete with live announcements and commercials. The person running the MP3 server is running the show in real time, and the listener only hears the portion of the show currently being dished up. When you tune in to an MP3 broadcast, you can't just pick an arbitrary tune from the show, any more than you can with radio.

Running an MP3 broadcast station is a fairly complicated matter, and you'll learn all about that in Chapter 8. As a user, however, listening to streamed MP3 is rarely more than a matter of point and click. To find MP3 broadcasts, check out sites such as *www.shoutcast.com*, *www.icecast.org*, *www.radiospy.com*, *www.mycaster.com*, *www.greenwitch.com*, or *www.live365.com* and you'll find dozens—or hundreds—of ongoing broadcasts. If your system is configured properly, clicking a link to a stream in progress will cause your MP3 player to be launched and (after a short delay) for that stream to begin playing. If it doesn't, see the following section, "Configuring Your System to Handle Streaming MP3."

Real MP3 streams are usually sent as *.pls* files (MIME type `audio/x-scpls`), rather than *.M3U*. The difference between these two playlist types is described earlier in this chapter.

The advantages to real-time MP3 streaming are:

- Much greater control for the webcaster (voice, live mixing, etc.)
- Webmaster can send a stream to many people without needing tons of bandwidth on the playback machine (though they still need access to a server with a fast connection)
- Difficult for user to save MP3 bitstream to hard drive
- Optimization of bitstream for various client bandwidths

Of bandwidth and buffers

To receive MP3 streams from MP3 broadcasts or pseudo-streams, your player must be capable of managing downloads and buffering streams over the Internet on its own. The vast majority of popular MP3 players are stream-enabled—even many of the command-line players for Unix/Linux.

The biggest concern for most users, of course, is the speed of their Internet connections. If you're on a slow connection and the stream being served up (or pulled down) carries more bits per second than your modem is capable of delivering, you'll experience choppy, halting playback.

This can be mediated somewhat by two solutions. On the client (user) side, a process called *buffering* can be used. In the buffering process, the MP3 player grabs a good chunk of data before it begins to play, and continues to read ahead in the stream. The music being played is thus delayed by a few seconds. The slower the connection, of course, the larger the buffer required. Theoretically, one could utilize a buffer so large that the entire song was downloaded before a second of music was played. This would guarantee perfect playback over even the slowest connections, but would undermine the advantage of listening to streams.

Most users, however, require more modest buffer settings. If you find that your MP3 streams are skipping or pausing as they're played, dig around in your player's options and preferences for something like "Streaming Preferences." In WinAmp, tap Ctrl+P to bring up the preferences screen and navigate to Plugins → Input → NullSoft MPEG Audio Decoder. Click the Configure button and select the Streaming tab, where you'll find an array of buffering options. Most likely, you'll just want to change the numerical value for kilobytes of prebuffered audio (try increasing it by 25% or 50% for starters). You can also control how much of a track will be grabbed before a single byte is played.

On the server side, webmasters can do a number of things to make things easier for their modem-connected users, including downsampling MP3 files to lower frequencies, encoding files at lower bitrates, and sending mono, rather than stereo, streams over the Internet. All of these, of course, degrade sound quality, but are necessary to deliver acceptable streams to modem users. A sophisticated MP3 server will offer high-bandwidth and low-bandwidth options so users can select the best possible quality for their connection speed (such as the Hi-Fi and Low-Fi pseudo-streaming options found at MP3.com).

Technically speaking, on-demand servers are capable of doing some of the things that broadcast servers do, such as downsampling MP3 audio on the fly. For the technically inclined, this would be a matter of creating a CGI interface that would invoke a downsampling program when the user accesses a link on the server, and sending the output of that program to the user rather than the actual file. This is really a CGI implementation, rather than an MP3 issue. While there aren't many sites doing this currently, the technique would have certain advantages for some users, since it wouldn't require the installation and configuration of special broadcast software. More on this in Chapter 8.

Configuring Your System to Handle Streaming MP3

Regardless whether you're accessing real MP3 broadcasts or pseudo-streams, the ideal is to have your browser, your operating system, and your MP3 player all configured to interoperate correctly so that accessing a link to an MP3 player automatically results in the right thing happening—your player gets launched and begins to play the stream with no further intervention on your part. In most cases, simply installing an MP3 player capable of handling MP3 streams is all it takes to set things up properly, but it's possible for the necessary associations to become broken if you fiddle around with a lot of software or tweak your browser settings. In addition, you may at some point want a different player associated with MP3 streams.

The easiest way to create an appropriate association is to look in the options and preferences of the preferred player for something like "Make preferred for all types." Better players will let you establish associations on a per-filetype basis so that you can, for example, make Sonique your preferred MP3 player for regular MP3 files, and WinAmp the preferred player for streamed *.M3U* and *.pls* files.

If you have multiple MP3 players and browsers on your Windows system, the easiest way to establish associations may be to download the MP3Fix utility from *help.mp3.com/help/diagnosis/*. Although MP3Fix is packaged in a Windows InstallShield package, running it will not in fact install anything on your system. Rather, it will scour your system for known MP3 players and allow you to make one of them the preferred player for all browsers.

Netscape Navigator/Mozilla

If Netscape Navigator is your primary browser, you can tell the program exactly how to handle any incoming MIME type (file type) by pulling down Edit → Preferences → Applications (this may be slightly different in various versions of Navigator). Scroll through the list of known file types for something like MPEG Audio File or WinAmp Media File and click the Edit button. To change the associated MP3 player, click the Browse... button and navigate to the location of your preferred MP3 player. If you don't find an entry in the list for the file type you want to associate, you can create a new one by clicking New Type... and filling in a description, file extension (e.g., MP3), a MIME type (e.g., *audio/x-mpeg*), and the path to an MP3 player.

Internet Explorer

If you use Internet Explorer as your browser, remember that Explorer is integrated into Windows itself. Therefore, it does not use a separate MIME association database. Instead, it uses the operating system's FileTypes panel. To change an association manually, open Windows Explorer (not Internet Explorer) and pull down View → Folder Options → FileTypes. Navigate to the Playlist file type (which may be called, for instance, "WinAmp Playlist"), click Edit, and use the resulting dialog to change the description and program association. More details on this procedure can be found in the sidebar "Configuring the Default Handler for MP3 Files in Windows" in Chapter 3.

Performance Considerations

The average consumer machine of today is much more efficient than it was a few years ago, and most people have gobs of underutilized computing horsepower to spare. Nevertheless, there are still plenty of older machines hanging around out there, and owners of those machines will want to make sure they've got the most efficient decoder available. And, of course, true geeks will want access to the most efficient decoder whether they need it or not. Here are a few techniques you can use to determine the efficiency of your MP3 player. The same techniques apply to MP3 encoders, by the way.

Benchmarking Decoders

As mentioned in Chapter 2, MP3 decoders have a lot less work to do than encoders, since the task of MP3 playback is far less CPU-intensive than encoding. All the decoder has to know is how MP3 files are structured, and how to reconstruct a coherent signal from the combination of audio data and "side information" stored in the files' frames. In days of yore, when a Pentium 90 was considered top-of-the-line and most people were using 486-based computers, playing MP3 files was proportionally intensive enough to be cause for concern, and many experienced a significant impact on system responsiveness when MP3s were playing in the background.

However, modern CPUs have so much processor bandwidth to spare that decode speed isn't much of an issue for most users. For example, playing an MP3 stream on a Pentium 233 with the average decoder for Windows will generally consume only 5% of your processor speed.

If you're experiencing a noticeable performance impact when playing MP3 files, you might want to do a little benchmarking of your own. CPU/resource monitors are available for most operating systems, either as part of the system itself or as a separate download. A few of the more popular options are listed in this section. If you can't find a CPU monitoring app for your system that breaks up CPU usage by task (as opposed to giving you an overall rate), just note CPU consumption with and without your MP3 decoder running, then calculate the difference.

Once you've got a resource meter up and running, try several different MP3 players and take note of how many system resources (CPU and memory) they consume. Keep in mind, however, that even if a particular MP3 player consumes less resources than another, it may do so at the expense of sound quality or important features. You'll have to establish your own criteria for the compromises you're willing to make.

Windows 95/98

Windows 95/98 both ship with a resource meter bundled in the system. If *C:\WINDOWS\SYSMON.EXE* is present on your system, you're all set. You may find a shortcut for Sysmon pre-installed in Start → Programs → Accessories → System Tools. If not, go to Control Panel → Add/Remove Programs → Windows Setup → Accessories and add it. Sysmon's default display shows the percentage of total CPU available being consumed by all currently running tasks, but does not tell you exactly how many resources are in use by each individual process. The large spikes in Figure 4-11 occurred when launching applications, and resulted in WinAmp skipping. The flat areas represent straight MP3 playback. Several other applications were open, but no work was being done. Sysmon reports approximately 15% processor consumption, but gives no indication of exactly how much of this is being consumed by MP3 playback. For a more specific breakdown or resource consumption, download Microsoft's "Kernel Toys," right-click *top.inf*, choose Install, then click Start → Run, and type *wintop*.

Popular third-party resource meters for Windows include L-Ement and sysmeter, both downloadable from *www.skinz.org*.

Windows NT

Windows NT 4.0 has a similar performance monitor built into the operating system. To access it, tap Ctrl+Alt+Del, click Task Manager, and select the Performance tab. Alternatively, use Start → Programs → Admin Tools → Performance Monitor.

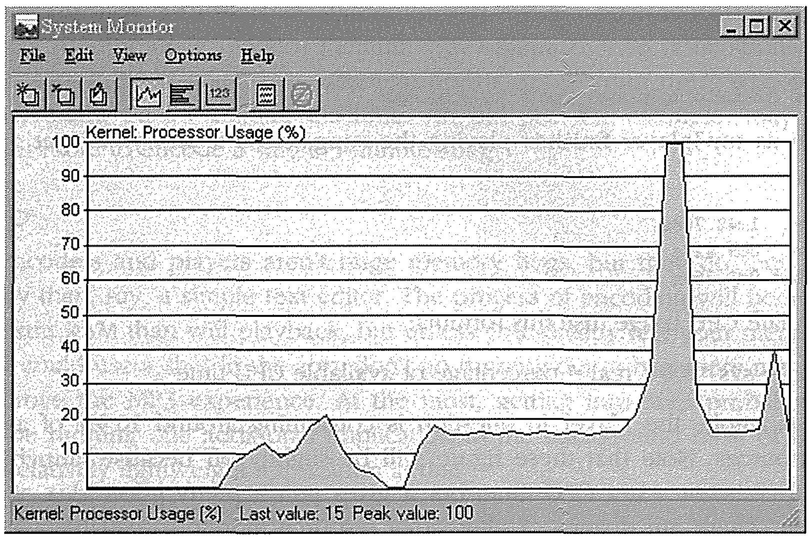


Figure 4-11. Using Sysmon to track CPU resource consumption as various tasks are performed

Mac OS

Search your favorite Mac software library for a utility called Process Watcher by Hugues Marty, which will allow you to view information on all currently running processes. However, Process Watcher will not show you a breakdown of CPU activity *per application*, so its ultimate usefulness is somewhat limited. You'll find a copy at *perso.magic.fr/suli/pw.html* if you want to check it out. Users may also want to experiment with Process Monitor or Process Spy, though this utility suffers the same limitation.

Linux

There are dozens of options for Linux users. Typing *top* at any prompt will give you a detailed overview of resource consumption, broken down by process. By default, top will update its report once per second, which may be too rapid to study properly. If you want a better view, use:

```
top -d 5
```

to have top update itself every five seconds. The problem with top is that it isn't always easy to figure out exactly how much CPU a given application is consuming, because many apps do their work in worker threads that may be difficult to identify. Fortunately, you can launch an app through the "time" utility. The operating system will calculate the total time the app was running and report the total number of seconds of processor time consumed in system space and in user space. From that, calculating the percentage of available CPU consumed by that app time's syntax is pretty simple:

```
time /path/to/mp3player /path/to/Song.mp3
```

As soon as mp3player finishes playing the song, close it and you'll get a report like this:

```
real  1m42.708s
user  0m9.427s
sys   0m1.769s
```

To calculate CPU usage, use this formula:

(user + sys)*100 / real = percentage of available CPU time

In this example, the player in question is consuming around 10.9% of available CPU resources. Note that these figure will be slightly off because launching the app will consume extra CPU time, but it gives you a pretty good idea of what's going on. There are dozens of alternative resource monitors available for Linux. Search your favorite software archive for GUI resource meters such as xosview.

BeOS

In addition to the Pulse application built into the system (which shows CPU load indicators for each detected processor), there are quite a few third-party tools available, downloadable from *www.be.com/beware/*, such as the graphical process controllers TManager and ProcessWatcher. The "top" and "time" utilities described for Linux are also available for BeOS, and function identically.

System Requirements

A few MP3-related hardware issues were covered in Chapter 2. To summarize and put it all in one place, here are the most important things you need to consider when contemplating hardware upgrades for your MP3 playback machine.

CPU speed

While you can theoretically use a 486 for MP3 playback, the experience is likely to be painful, as even the comparatively low-resource act of decoding MP3 is going to consume a good deal of horsepower that you'll probably want to give to the operating system itself. Trouble-free MP3 playback really demands a Pentium-class computer or better, where processor consumption may be around 5–10%, rather than 50–75%. Mac users, similar deal: Anything slower than a PowerMac or PowerMac clone is going to give you problems with MP3 playback, if you can even find an MP3 player that works on your machine at all.

Encoding is another matter altogether. Because encoding requires so much math, you'll be happiest with the fastest machine you can get your hands on. While fast encoders running on a modern Pentium can encode a song faster than it would take to play it back, you'll hear absolute horror stories from users who try to do their encoding on an antique machine. An old 386, for example, may take 12 hours or more to encode a single 3-minute song.

Memory

MP3 encoders and players aren't huge memory hogs, but they do require more memory than, say, a simple text editor. The process of encoding will benefit more from extra RAM than will playback, but unless you already feel your memory subsystem could use a shot in the arm, don't go running out to buy more memory just to improve the MP3 experience. At the most, getting into MP3 probably means you'll be running one additional application in the background most of the time, and a relatively lightweight application at that.

Sound cards and speakers

Again, this is a critical area. Most people have traditionally thought of their sound cards as a vehicle for games, or for listening to streamed news broadcasts. But as you get more involved in MP3, you'll start to think of your computer as a cousin of your home stereo in some ways, and will want to get the best quality you can. Do some research online to find out what the audiophiles are recommending this month as the best-sounding card at an affordable price. In the sub-$100 range, top cards from CreativeLabs and Yamaha were among the most popular cards among MP3 buffs at this writing.

Chances are, the speakers that came with your computer aren't of the best quality. Again, look around online for recommendations. Get yourself a nice set of powered subwoofer/satellites, connect them to your new sound card, and be prepared to be blown away by the difference they make over the default audio hardware that probably shipped with your computer, as shown in Figure 4-12. Because the system has a built-in amplifier and runs on standard wall power, you don't need to rely on the amplifier built into your sound card. If you're going speaker shopping for your computer, you'll get better results by going to an audio store and telling them you want speakers for use with your computer than by going to a computer store and telling them you want high-quality computer speakers.

Of course, you can always route audio signal from your computer directly to your home stereo by purchasing a length of RCA cables and a simple adapter. See Chapter 6, *Hardware, Portables, Home Stereos, and Kits* for details.

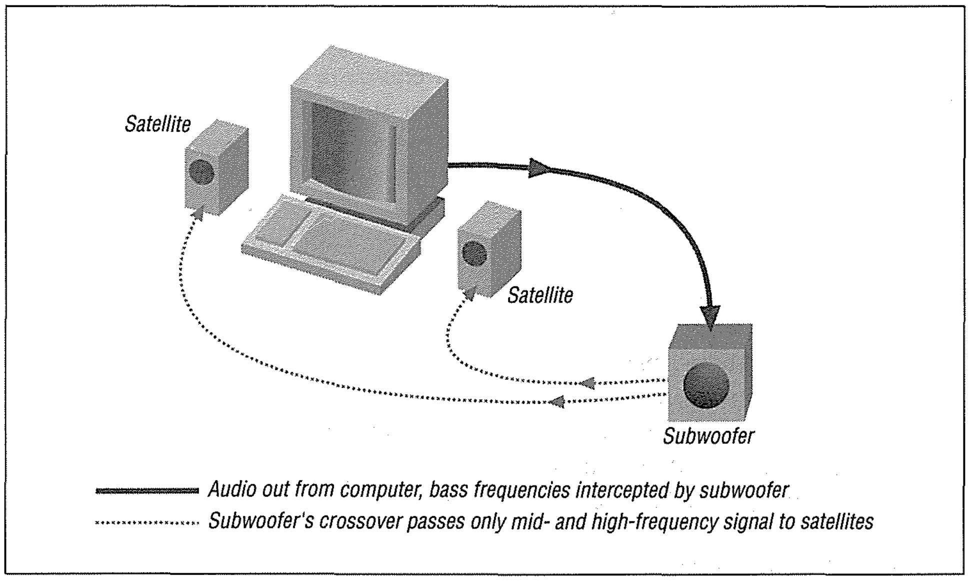

Satellite
Satellite
Subwoofer
Audio out from computer, bass frequencies intercepted by subwoofer
Subwoofer's crossover passes only mid- and high-frequency signal to satellites

Figure 4-12. Mid-range and high frequencies are handled by satellite speakers

Disk speed and space

People sometimes wonder whether their hard drive is fast enough for good MP3 playback, or contemplate going with a high-end SCSI adapter/disk to optimize their systems for MP3 use. For the most part, this is piffle. At the default bitrate of 128kbps, a mere 16 K is being scooped from disk every second—even the lowly floppy drive can keep up with that. All of the work is going on in the CPU. The fact that MP3s are so highly compressed not only means you save on disk space, but that the hard drive has to do very little work.

However, there are some caveats to this point. If your all-IDE system has more than one hard drive and one CD-ROM drive, you'll end up putting two devices on a single channel with IDE. But IDE, unlike SCSI, cannot transfer data back and forth between two devices on the same bus or channel simultaneously—IDE accesses are serial. Therefore, you can end up with your MP3 player requesting data from a storage volume while the operating system is requesting data from the boot volume simultaneously, which can potentially lead to skips or pops in the audio stream. But IDE disks are so much more affordable than SCSI that most users in this position will simply have to deal with it. If you need to run several IDE devices and want to ensure optimum performance, you might want to consider purchasing inexpensive IDE controller cards for the additional drives, so that each one gets a channel all to itself.

Note also that IDE disks are more CPU-intensive than SCSI disks, which can have an impact on low-performance systems. If you have UDMA IDE drives, make sure they're running in UDMA mode to lessen the processing impact on the CPU.

Once you start on the task of encoding your entire collection, or downloading lots of tracks from the Internet, you'll likely find disk space being chewed up faster than you ever dreamed possible. Suddenly those 30 GB drives on sale down at the local computer superstore don't seem obscenely huge after all. Of course, it won't be long before you're looking for some kind of external storage solution, and most people start burning data CDs full of MP3 files to deal with the space issue. See Chapter 5 for more on that.

When playing MP3 files from CD rather than from your hard drive, you may experience skipping, or "halting" playback, especially if you're using an older or cheapo CD-ROM drive. This is probably not happening because the CD-ROM drive isn't fast enough, but because it isn't accurate enough and has to do a lot of error correction to grab data accurately. You may be able to correct for this by increasing the "buffer" or "preload" size in your MP3 player. Dig around in the options and look for a control that will let you establish the size of the "preload" or "buffer" size, and turn it up until the problem goes away. By doing this, you're telling your MP3 player to read data from disk in larger "gulps," which gives the drive more time to retrieve data from CD successfully before the amount of music stored in RAM is exhausted.

Sound/video card interactions

One of the most frequently encountered hardware-related problems users find is that even very fast machines can "skip" or "pop" MP3s when switching tasks, scrolling windows, or doing other things that require a lot of screen redraw. This typically occurs for users with PCI graphics cards, because some card manufacturers have attempted to increase card performance by letting video cards take control of the entire PCI bus when necessary, which can prevent an even stream of audio data from reaching a PCI sound card at an adequate rate. Many people have noted that this does not occur with AGP graphics cards or ISA sound cards (since they're on a separate bus), which you may want to keep in mind when shopping for new hardware. If your audio and video cards are both on the same bus, you may be able to tweak some of the video card's driver or BIOS settings to disable or lessen this effect, though possibly at the expense of some graphics performance.

Obsequieum: A networked MP3 Jukebox for Linux
obs.freeamp.org

PC104: Small, stackable motherboards for use in dedicated units of all kinds
www.pc104.com/

Wired remote control products for car players
www.mp3ondemand.com

XIMP
ximp.iscool.net

In this chapter:
- *The Rules of Engagement*
- *The Players*

7 The Not-So-Fine-Print: Legal Bits and Pieces

The legal side of the MP3 world can be either utterly simple or unbelievably complex, depending on how you approach and use the technology and how thoroughly you'd like to understand the inner machinations of the U.S. legal system (which is increasingly tied into the legal systems of other countries, thanks to the fact that "bits don't stop at the border"). Balancing the need to protect consumer rights with the rights of artists and the recording industry has opened up a Pandora's box of conflicting interests that may never be fully resolved. In this chapter we'll take a look at some of the common myths surrounding the transfer of digital audio files (The Rules of Engagement) and then examine some of the finer points of copyright law (The Players), along with several initiatives, technologies, and specifications that have come into play, including the Audio Home Recording Act, the Secure Digital Music Initiative, the No Electronic Theft Act, the Serial Copy Management System, the MP3 patent, and more.

This chapter represents our best attempt to summarize and put into plain language the myriad legal aspects surrounding the MP3 scene. As we are not lawyers, and because the scenery changes on a near-weekly basis, this chapter does not represent the final word. If you need to make critical legal decisions regarding MP3 usage, consult with an attorney specializing in copyright and digital issues.

The Rules of Engagement

Throughout this book I've said the same thing over and over again: You are entitled to create MP3 copies of the music you own for your own personal use. You are not entitled to make accessible digital copies of music to which you do not hold the copyright. While the ramifications of this are fairly clear (i.e., you can't just copy tracks from a CD and put them on your web site), there's a whole lot of misinformation floating around out there, and many people believe there are numerous exceptions to this rule. Let's try and set some of these common misconceptions straight. Some of the content in this section is paraphrased from the "Top 10 Myths" section of the RIAA's ancillary web site, *www.soundbyting.com.* Please see that site for the recording industry's perspective on artist and consumer rights in the digital arena.

Everything in this section may seem like small potatoes to you. It may seem trivial and unimportant. You may think that your personal activities are too small and insignificant to be noticed. While that may be true, remember that if you are caught, copyright violation that involves more than 10 copies and a value exceeding $2,500 is *a felony* in the United States. OK, chances are pretty slim that you'll go to jail over it. Very few people ever see anything but a cease-and-desist letter, or have the plug pulled on them by their service provider. But it's important to realize the potential gravity of what you're dealing with here.

Copyright

Virtually everything in this chapter depends on a solid definition of copyright, so let's get that out of the way first. We'll focus on the aspects of copyright that apply to music copying and distribution.

In a nutshell, copyright and copyright law are means of making sure that the person or agency who creates a creative work has control over who gets to make copies or derivations of that work and how. Copyright is, quite simply, the right to copy, and the creator is the person who holds that right. Of course, the owner of a copyrighted work also has the right to sell the right to someone else. In the music industry, artists typically sell their copyright to the label that publishes their work.

In order for a work to be copyrighted, it must exist tangibly in the real world—not just in your head. A song you make up and hum to yourself is not copyrighted. A song you make up and write out on paper or record to DAT tape or post to Usenet *is* copyrighted. Computers and the Internet have somewhat complicated the notion of making copies. There are obvious copies, such as those created when you download a song from a web site to your hard drive, and less obvious copies, such as the copy of a song that's loaded into RAM from your hard drive when you go to play a song. But that sort of "ephemeral" copy is seldom of concern to anyone, as it's not likely to affect the artist's right to earn money from the work.

Copyright law differs in some specifics from one country to the next, but most of the world's countries have signed the Berne copyright convention, which specifies that a work becomes copyrighted the moment it's put into tangible form. Creators are not required to affix a copyright notice to their creation, though they do if they're wise—it can help in resolving disputes should they arise later on. But the important thing is, copyrights aren't like patents; artists don't have to file them with any legal bureau or do anything special. If you create something tangible, you just earned the copyright to that thing.

Of course, money is the reason most copyright holders want to retain control over their right to copy. If that right slips away, the creator is deprived of potential earnings they might make from controlling the right to copy. That's why digital music distribution is such a problem for artists—it grants them the opportunity to reach a wide audience, but simultaneously weakens their control over the copying process in a way that compact disc distribution (for example) does not.

A great place to learn more about copyright is "Ten Big Myths About Copyright Explained," at *www.templetons.com/brad/copymyths.html/*.

Fair Use

The notion of "fair use" is an important element of U.S. copyright law, and allows some parties to use portions of some copyrighted works in a limited way in some situations, without obtaining prior permission. Fair use is designed to make sure that intellectual or artistic expressions aren't totally blocked from being entered into the public discourse by overly strict copyright enforcement that could limit the free flow of ideas. By necessity, fair use has always been the source of gray areas in copyright law. However, many people misunderstand the concepts of fair use and mistakenly think it means they can trade copyrighted MP3 files around freely.

When courts look at cases where the notion of fair use is brought to bear, they consider whether the property in question was used for commercial purposes, the length of the excerpt, whether the excerpt utilizes the most distinctive portion of the piece, how distinctive the original work was to begin with, and how the usage in question will affect the market for the original work. Courts will usually grant fair use protection when the excerpt is made in the context of commentary and criticism, or for educational purposes, although there are no hard and fast rules here. In any case, making available to the public an entire track of digital audio will almost certainly not qualify for fair use protection in almost any circumstance.

Personal Use

Do not confuse the terms "fair use" (described previously) and "personal use." The latter is a right granted specifically to users of digital audio recording devices to guarantee that they can make copies of the original work on other media. For example, if you have a CD player in the home but a cassette player in the car, "personal use" guarantees you the ability to listen the music you purchase on CD while you're in your car, since you can legally make a cassette copy. The line is still drawn at handing that music out to people who didn't purchase the music.

The devil's advocate will note that the cost of a blank cassette tape includes a small royalty that flows back to the recording industry on the assumption that a certain number of blank cassettes will be used to create illegal copies of copyrighted works. One interpretation of this is that even when you create a cassette copy of a work for personal use, you still pay the recording industry for it, albeit a small amount. But note here that no part of the price you paid for your hard drive flows back to the recording industry, so the situation is not identical.

Personal use derives from the Audio Home Recording Act of 1992, described later in this chapter. The important bit for MP3 users is the distinction between a computer hard drive and a digital recording device. Even though a computer can digitally copy, a computer is considered a "multipurpose device," not a recording device. Computers *are not* covered by the AHRA, which means that no royalties flow back into the recording industry from computer-created copies. Furthermore, the AHRA does not require that computers include SCMS protection (also covered later in this chapter). Because computer-based copying does not generate royalties for and offers no protection to the recording industry, creators of computer-based copies (i.e., you) do not enjoy immunity from suits over copyright infringement. However, this does not imply that copies you make of your own music, for personal use only, are infringing copies. It only means that if you are trafficking music you do not own, the AHRA won't protect you.

The short answer: If you made the digital copy on a computer, you do not qualify for the personal use clause of the AHRA.

Note that the limitations of personal use apply to people downloading MP3 files as well to webmasters—if you download a copy of a copyrighted song from another site, you can't hide behind the "personal use" clause, because the devices being used for the transfer are computers and therefore aren't protected by the AHRA.

Keep an Eye out for Bogus "Permissions"

When surfing for MP3s, it's not uncommon to find notices like these on Web sites offering MP3 downloads:

- Please support the artist and buy the CD.
- This site is for promotional purposes only.
- You must already own these CDs to legally download these tracks.
- If you download these files, you must delete them from your hard drive within 24 hours.

All of them are nonsense. Unless the webmaster owns the copyright or has explicit permission from the copyright holder, none of the above caveats make a wit of difference. The site is still illegal. As the RIAA says, "If you reproduce, offer to distribute and/or distribute full-length sound recordings without a license, you are violating copyright law." So how can you tell whether a site is legitimate or not? Unfortunately, it's not always so easy, and it's always possible that the webmaster of a site may be lying when he says you have permission to download a given file. You'll have to make judgment calls based on the appearance of integrity the site displays. One good technique you can use to determine whether a site is hosting legitimate downloads is to head over to an online compact disc store such as CDNow.com and search for corresponding artist/track titles. If you can find the same track on an existing CD, you can be fairly certain that the files being offered for download are illegitimate.

Using International Servers

Some people are aware that copyright laws are different in non-U.S. countries, and thus assume that if they host their files on a server located in another country they will have circumvented U.S. copyright law. There are two problems with this line of thinking.

First of all, if you upload or download to or from a machine located in the U.S., you'll still be in violation of the law. And no, sending removable cartridges or disk volumes by mail isn't going to get you around this issue. Second, copyright laws in many countries are very similar to those in the U.S. In some cases, they're more strict than U.S. laws, and in others they're more lenient. You'll need to do careful research into the laws of the country hosting the server to find out how any given country is handling the situation.

The Berne Convention guarantees that a copyright violation committed in one country will be subject to governing laws in another member country. The convention makes it much easier to prosecute copyright violations internationally, and virtually guarantees that member countries will cooperate to prosecute copyright violators regardless their country of citizenship and the country in which they commit the crime. Since most major nations of the world have signed the Berne Convention, attempts to work around copyright laws of one country by moving the base of operations to another country usually doesn't work.

The First Amendment

Some people believe that the First Amendment, which guarantees every American the right to free expression, will protect them from charges of copyright violation. Keep in mind, however, that copyright itself is even more fundamental to the Constitution than free speech; it was written before the Framers even got around to the first ten amendments. But more importantly, the First Amendment does not override the principles of copyright. Free speech does not give anyone the right to deprive an artist of their right to market their works on their own terms.

Linking to MP3 Downloads

What if you don't have any interest in offering pirated music for download, but do want to create an "index" site pointing to other sites where illegal files are available for download? Here we get into ambiguous territory. Linking-related legal questions have come up on the Internet again and again, and no hard-and-fast answers have come to the fore. In the case of MP3, the question revolves around the legality of linking to unauthorized MP3 downloads even if you don't take direct responsibility for hosting those files. The RIAA has traditionally taken the tack that such sites are guilty of "contributory copyright infringement"—kind of like being a willing accessory to a crime.

However, the RIAA has not (at this writing) made significant headway in enforcing this viewpoint. As you know, many linking sites are run by big-name companies that would stand to lose a lot if forced by law to stop linking altogether. The RIAA is only able at this point to say "we believe" that such sites are guilty of contributory infringement. The question may boil down to whether the linking site has "the right and ability to control the activities of the direct infringer and also receives a financial benefit from the infringing activities." Furthermore, "Liability may be imposed even if the entity is unaware of the infringing activities."

In other words, this one is technically up in the air. It's commonly done, there is to date no court precedent to settle any suits that might arise from the practice, and even some of the Internet's largest companies (such as Lycos) do it. Lycos, for their part, claims to act quickly when reports of copyright infringement are submitted, and deletes links to those sites from their database. Lycos also provides a complete and detailed disclaimer of liability (*mp3.lycos.com/disclaimer.html*), and makes it patently clear that they are not providing the content being linked to. The RIAA doesn't like it, but at this writing, there's no proven law against it. The Internet is just too fuzzy for the law sometimes. You'll have to draw your own bead on this one.

In late 1999, the International Federation of the Phonographic Industry (IFPI) arrested a 17-year-old Swedish citizen for maintaining a site consisting of links to illegal MP3 files. The youth was later acquitted on a technicality, though IFPI still maintained that he was guilty of complicity to crime against copyright law, i.e., contributory infringement. To date, there are still no known cases of people being prosecuted for linking to MP3 files, but it's probably only a matter of time before a precedent is established.

Not-for-Profit Sites

Some people believe that if they don't seek to gain any profit by offering copyrighted music for download, then their files are legal. This is most definitely not true. What this viewpoint fails to take into account is the fact that your site can potentially result in lost sales for the artist and the record label representing the artist. Whether you profit from it or not does not exempt you from the responsibility to respect the copyright holder's rights to do as they please with their own material. This stance is clearly laid out in the No Electronic Theft Act, which amended Section 506 of the Copyright Act.

Free Advertising

Some people believe that they're doing the copyright holder a favor by promoting their material for free. Who could argue with free advertising? The problem with this line of thinking is that it's up to the copyright holder to decide whether he wants free advertising or not. You can always ask the copyright holder whether you can advertise for him or not, but you can't make these kinds of assumptions on his behalf.

Of course, it can't hurt to ask; a little communication can go a long way. But if you want to distribute the works of a major artist, it's usually not going to do you any good contacting them directly, as the copyrights to their work are probably held by a record label. If the label is small enough, there's always the chance they may say yes... but don't count on it. If you're talking about a major label artist, your chances of getting permission to redistribute works by their artists are nil to zero.

The Players

As soon as you start trying to look into the current state of MP3 law as it pertains to copying, distributing, digital devices, and patents, one thing becomes immediately apparent: The whole situation is a mess and a minefield. Dozens of laws, acts, protocols, specifications, and precedents come into play. Most of them are interdependent in some way, and some of them come close to contradicting one another. One of the reasons why most people are so lackadaisical toward MP3 copyright issues is because it's all so difficult to figure out. While several sites exist for the purpose of explaining it all to the layman, many of those sites also have their own interests at stake, and may not give you the straight dope. And God help anyone who tries to go straight to the laws as written. The author of this book has determined that trying to make sense of all that legalese can cause brain damage.

The best you can do is to start sifting and sorting through the various published opinions and summaries presented out there and try to piece it together for yourself, keeping in mind that many documents conceal their own hidden agenda. Here, then, is a quick guide to "the players"—the most significant laws, acts, recommendations and specifications that affect the legality of MP3 copying and distribution.

Again, keep in mind that the law is a fluid thing, and may very well have changed by the time you read this.

The Audio Home Recording Act (AHRA)

www.hrrc.org/ahra.html/ The AHRA was passed in 1992 and is based on the recording industry's assumption that many digital audio recording devices are used for the copying and distribution of pirated music. While the industry had once feared that the cassette tape would kill music sales (clearly, this did not turn out to be true), the advent of Digital Audio Tape recording/playback devices and units like the Sony MiniDisc rekindled those original fears. However, this time the industry was able to make a better case. Unlike the analog cassette tape, digital devices give the user the ability to spawn a potentially infinite chain of copies where every copy enjoys the same audio quality as the original. Where cassette copiers had to keep making copies from the original source in order to keep audio quality decent, it was now suddenly possible to make copies of copies of copies of copies ad infinitum without ever sacrificing quality.

In an attempt to balance the threat to the recording industry of digital copying techniques with the need for individuals to retain the right to make their personal copies, a compromise was reached regarding digital recording devices. The AHRA requires that all digital audio recording devices include a mechanism or system that will allow first-generation copies to be made, but not further generations. The AHRA thus requires digital devices to be incapable of spawning the infinite copying chain. You would, for instance, be able to make a first-generation copy and send it as an email attachment to your sister in Poughkeepsie. But if your sister then tried to transfer that copy onto her own DAT player, she would be unsuccessful. The system works by setting a "copy bit" in the file's header to the "on" position. Any AHRA-compliant device looks for this bit, and refuses to allow a copy to be made if the bit is found to be switched on. This serial-copy prevention mechanism, found in all DAT, MiniDisc, and legitimate home stereo or portable MP3 *recording* devices is known as Serial Copy Management System (SCMS).

The AHRA doesn't stop there. The act also requires that vendors of such devices register with the Copyright Office, and that 3% of the cost of blank digital audio media be collected as a "tax" and turned over to the RIAA. This tax was intended as a means for the recording industry to recoup some of the losses they expected to incur as a result of rampant piracy.

But there's a *really* big problem with the AHRA tax: It operates on a presumption of guilt. Even if you're adamantly in favor of recording industry and artist rights, even if you're a recording artist yourself and never, ever try to copy a file illegally, you have to pay the tax anyway, because the industry has convinced Congress that you're probably a pirate. Possibly even worse, there are tens of thousands, perhaps millions of legitimate MP3 files floating around out there that are not attached to any particular label. All of the songs downloadable from MP3.com, for example, are placed into the public sphere by the artists who created them. Those artists *want* you to copy their songs around. They *want* to use the power of the Internet to get their music into as wide a distribution net as possible. They have blessed the limitless copying of their own music. But certain aspects of the AHRA make this difficult, because the RIAA has convinced Congress that most MP3 files are pirated, totally ignoring the completely legitimate distribution of thousands of works of art by the artists on their own terms.

In exchange for their agreement to abide by the terms of the AHRA, hardware vendors gain legal immunity from any lawsuits that might arise from any illegal copying done for noncommercial purposes using their equipment. This does not mean that all copying suddenly becomes legal, only that manufacturers of the devices cannot be held responsible if their customers use the devices for illegal copying.

So, what are the specific ramifications of the AHRA for MP3 users? First of all, the act includes a provision that allows users to *space shift* the music they own. Just as television users have the right to "time shift" programs, i.e., tape programs to watch them later, digital audio users have the right to make copies of the music they own in order to listen to them in other locations: in the car or at the beach, for example. Space shifting may necessitate the creation of multiple copies of a track in different formats; i.e., the user uses a cassette deck at the beach and a DAT player in the car. This is a key consumer right, but it's also the very right most often abused, leading to a tendency to make personal copies public.

It's important to understand that the act applies only to certain types of devices. No analog devices or analog transfers are covered, so this has nothing to do with standard cassette players, for example. The analog clause, by the way, opens the door for manufacturers to create MP3 devices that accept analog input. More importantly, "general-purpose" devices such as computer hard drives and CD-ROMs are not covered by the act, so the AHRA has no bearing on users who burn audio or data CDs of audio material originating from MP3. And since these devices are not covered, the AHRA provides no immunity to makers of CD burners.

Where the AHRA does make contact with MP3 is in the realm of devices specifically designed to record MP3 files. This was the crux of the RIAA's suit against the Diamond Rio portable MP3 player in 1999. The RIAA tried to make the case that the Rio was a recording device and therefore subject to the terms of the AHRA. Since Diamond did not implement SCMS or pay royalties back into the industry, the RIAA felt that Diamond was flouting the law. However, Diamond was able to make the case that the Rio is not in fact a recording device, but rather a simple storage and playback device. The Rio doesn't make digital audio, it just stores it and plays it back. Diamond even went as far as to implement security measures that would prevent the transference of audio files out of the Rio and into other devices. Diamond rightfully won their case on these grounds.

However, it is entirely possible that future portable MP3 players may gain the ability to record from external input, or from a built-in radio tuner. In this case, the AHRA would clearly apply to vendors of these devices, and those vendors would need to implement other anti-copy mechanisms to stay legal. At the moment, no such portable devices exist. But there are an increasing number of home stereo MP3 components that can both play and record MP3 files.

AHRA in the real world

To give you an example of how the act applies to a real life vendor of an MP3-recording device, let's take a look at ReQuest's AudioReQuest (see Chapter 6, *Hardware, Portables, Home Stereos, and Kits*). The machine is capable of accepting an audio CD and generating a pile of MP3 files for playback. This means several things for users of AudioReQuest (and similar devices):

- If a CD has its copy bit set to "high," only one digital copy can be made, and that copy is encrypted and marked so that it cannot leave the device.
- If a CD has its copy bit set to both high and low, the device recognizes the disk as a digital copy, and will refuse to encode its tracks. It will, however, play the CD normally.
- The AHRA states, "As long as the copying is done for noncommercial use, the AHRA gives consumers immunity from suit for all analog music copying, and for digital music copying with AHRA covered devices." To respond to this clause, AudioReQuest's Line-in jack is analog. Music entering through the analog port is digitally encoded, but by this time it has already gone through an analog phase and has therefore lost a (very) small amount of fidelity in the conversion process, thus responding to one of the fears the AHRA intends to address. Future versions of AudioReQuest will have digital ports, and those ports will comply fully with the Serial Copy Management System described later in this chapter.
- When you copy digital files from a PC to AudioReQuest, it becomes an "interface device" as well as a recording device. Since MP3 itself doesn't have SCMS capabilities, AudioReQuest will play it safe and assume that the files are copies and will encrypt them. This potentially saves AudioReQuest's butt, but note that this practice unfairly assumes that the user is guilty of piracy even if the files in question are not pirated. If you're the artist who created the MP3 files in question, you're unfairly disadvantaged.
- Even more significantly, all files transferred from a PC to AudioReQuest are moved, rather than copied. That's right—the files will be deleted from your computer's hard drive after copying, again because ReQuest has to assume the worst to stay on the good side of the law. This saves AudioReQuest from functioning as a digital recording device in this case, and avoids "serial copying"; the transfer becomes analogous to carrying a CD from your house to your car. If you don't want files deleted from your hard drive, your only choice is to make backup copies of them first.

All of this is messy, but necessary from the vendor's perspective. Either they expose themselves to lawsuits, or they comply with AHRA and enjoy its immunity clause. Simple as that. Note that some of the points above could change when SDMI (discussed later in this chapter) is finalized.

The Digital Millennium Copyright Act (DMCA)

www.digmedia.org/DMCAexp.htm/ The DMCA, signed by Bill Clinton in 1998, is a portion of the Copyright Act that implements a pair of international treaties from the World Intellectual Property Organization (*www.wipo.org*). The act has two main components: 1) To make it illegal to circumvent technologies created to protect copyrighted works, and 2) To define the terms by which Internet Service Providers may or may not be responsible for copyright infringements.

The DMCA makes it a crime to create, sell, or use technologies that can be used to circumvent copyright protection software devices. This applies equally to things like digital watermarks, encryption techniques, and even firewalls. However, the DMCA also takes care to provide an exception to its anti-circumvention rules:

> ...a person may develop and employ technological means to circumvent a technological measure, or to circumvent protection afforded by a technological measure... for the purpose of enabling interoperability of an independently created computer program with other programs, if such means are necessary to achieve such interoperability, to the extent that doing so does not constitute [copyright] infringement.

In other words, the DMCA wisely allows for the possibility that it may be necessary for copyright techniques to be used to protect a consumer's right to make copies of legally purchased works. For example, if you purchase a song in a secure format that cannot be played on another computer, what does this do to your right to play music you own on multiple devices? A copyright circumvention tool may be necessary to guarantee that copyright protection mechanisms don't interfere with your right to create multiple personal copies of a given work.

The DMCA also entitles webcasters to a "statutory license" to perform copyrighted sound recordings over the Internet, and an exemption to let webcasters make a single copy of a recording (an "ephemeral" recording) and a statutory license for storage on servers for the purposes of webcasting. The DMCA also launched a two-year study on how current law regarding making copies of audio music should apply to the Internet (the Internet, as you know, changes everything).

The DMCA provides for the notion of "safe harbors" for ISPs. In particular, a service providers' liability is limited when the ISP is acting as a "mere conduit" for the data in question, when such data is just being cached (so long as the caching mechanism is designed to meet copyright owners' rights), when the ISP does not know that their equipment was harboring illegal copies, when the ISP does not benefit financially, if ISPs respond "expeditiously" to notifications of illegal activity, and when search engines provided by ISPs unknowingly link to illegal material (so long as they then "respond expeditiously" to the offending material). These protections ensure that ISPs are not legally saddled by circumstances outside their control, but they still place a burden on ISPs that didn't exist just a few years ago.

Webcasting

The exact implementation details of the DMCA as it applies to webcasting have yet to be worked out at this writing, and will ultimately be the product of lengthy negotiations between DiMA (The Digital Media Association), who represents webcasters, and the RIAA. Both sides see the value and the intractability of webcasting, but there is some disagreement over how licensing issues should best be decided. It is likely that webcasters will be able to obtain a license to broadcast files over the Internet by paying a single fee and deal with a single point of contact. This will help to avoid the impossible situation of webcasters needing to work out royalties with the label of each artist whose work they intend to broadcast, which is technically the situation at this writing (though few, if any, independent webcasters are bothering to actually do this).

After five months of talks, the RIAA and DiMA were unable to reach an agreement, and talks had officially stalled. Meanwhile, heavyweights such as America Online, broadcast.com, Eclectic Radio Corporation, MTV Networks, NetRadio, RealNetworks, SonicNet, Spinner.com, Tunes.com, and Westwind Media had signed on with DiMA. As the stalled talks moved inevitably toward arbitration, DiMA was stating that they wanted the following goals recognized:

- Webcasting provides promotional value to new artists.
- Webcasting allows for the creation of partnerships with the recording industry; users will be able in some situations to purchase the music they're currently listening to.
- Rate models already exist in the marketplace (such as in traditional radio) and that the Internet should not be considered categorically different from existing markets.
- Undue restrictions or requirements on webcasters will impose hardships that are not imposed on traditional broadcast models.

In any case, you'll need to pay royalties to ASCAP and BMI if you want to run a legal webcasting operation. BMI requires a base rate of $264 per year if your site garners less than $12,500 per year in revenue. ASCAP appears to have lowered their rate from the $1,000 per year they once charged, though the exact amount was not available at press time. See *www.bmi.com* and *www.ascap.com* for details. Also unclear at this writing was how artists not covered by those organizations were to be compensated. A good deal of useful information on the subject can be found at *www.kohnmusic.com.* If your webcasting operation is for-profit, note that ASCAP and BMI have a reputation for vigorous prosecution. If you're a small or non-profit operation, you're probably safe, though that doesn't mean you're legal without going through the proper channels.

The No Electronic Theft Act

Passed by the House of Representatives in late 1997, the NET Act amends copyright law to:

> define "financial gain" to include the receipt of anything of value, including the receipt of other copyrighted works. Sets penalties for willfully infringing a copyright: (1) for purposes of commercial advantage or private financial gain; or (2) by reproducing or distributing, including by electronic means, during any 180-day period, one or more copies of one or more copyrighted works with a total retail value of more than $1,000. Provides that evidence of reproduction or distribution of a copyrighted work, by itself, shall not be sufficient to establish willful infringement.

How's that again? Quite simply, the NET Act means that pirates who make money from their practice—whether that money comes from selling files, advertisements, or any other related means—can be sent to the clink for three to five years. In addition, it means that if the total value of the files posted exceeds $1,000, you can go to jail even if you don't directly profit from having made the files available.*

Those are strong words, and appear to be a strong tool the recording industry can leverage against pirates. Of course, enforcing the law is a more difficult matter, and so far we haven't seen a whole lot of pirates prosecuted in accordance with the terms of the NET Act. But that's not to say there haven't been any. In August 1999, a 22-year-old University of Oregon student pleaded guilty to charges of criminal infringement of copyright law thanks to a collection of MP3 and other illegal files he had made available on his web site. The perp faced 3 years in prison and a $250,000 fine. While this was the first actual arrest under the NET Act, it most certainly will not be the last. Don't let anyone tell you the industry isn't serious about this stuff.

The First Sale Doctrine

This doctrine comprises the section of the Copyright Act providing consumers with the right to sell or otherwise transfer a copy of an audio recording to another person, following the same logic that allows you to sell a copy of a book or compact disc to a third party. However, note that file-based media raises an interesting problem: When you sell a book you've purchased to someone else, you no longer have a copy of that book. When you sell an MP3 file you've purchased, what's to prevent you from keeping a copy of that file around on your hard drive?

* How is the value of illegally available tracks calculated? That's for the courts to decide. Do not automatically assume that a CD with 13 songs and costing $13 retail has a value of $1 per track. Because every download of that track is potential lost revenue, the courts may conclude that even a single illegal copy on your site is valued at over $1,000.

Theoretically, a system could be established that would let you "check in" your copy of the file with some central database, or to log in and register the transfer of ownership. Such a system would be dependent on software that would only allow you to play files bearing a valid registration number. Needless to say, such a system would be fraught with complications, but this is at least possible in theory. In any case, it's important to distinguish between selling a copy of *a* copyrighted work from selling *your* copy of that same work.

This right meshes with traditional notions of property ownership, but potentially conflicts with some copyright protection mechanisms, which explicitly prevent such transfers through the identification of digital watermarks or other means. In fact, some believe that the First Sale Doctrine even specifically legitimizes the use of some copyright circumvention devices and applications, since it is possible that someone might procure a product like TotalRecorder or AudioJacker to save encrypted music to a non-encrypted format in order to transfer that piece of music to another person. Thus, tools like these may actually become necessary in order to preserve consumers' established rights. Otherwise, you'll never be able to sell a song you purchase from emusic.com or other online music vendors because SDMI (discussed later) will prevent its transfer.

International Issues

Some issues in the MP3 universe are complicated by the fact that the Internet is international in scope, but people live in individual countries and are subject to the laws of those countries. As you may have noticed, almost everything in this chapter is U.S.-centric, and refers to American organizations and laws. This raises some important questions for MP3 users in other countries. For example, once licensing fees and terms are established by the RIAA and DiMA, they're likely to feed royalties back into American organizations like ASCAP and BMI. What will that mean for Canadian webcasters? The Canadian equivalent of ASCAP is SOCAN, but SOCAN has, as of this writing, not yet weighed in on the webcasting issue. Repeat this confusing scenario for every wired country in the world and things get very messy very fast.

Another good example of the kind of confusion that can be caused by this disjunction is the case of the BladeEnc encoder (Chapter 5, *Ripping and Encoding: Creating MP3 Files*). BladeEnc's author lives in Sweden, and Swedish law prevents the patenting of algorithms. But MP3 *is* patented by its inventors, in both Germany and the U.S. BladeEnc's author distributes the BladeEnc source code under the terms of the LGPL (Lesser General Public License), under the advice of his ombudsman,* who has suggested that it's perfectly legal to distribute BladeEnc in Sweden without paying Fraunhofer their royalties. No attempt is made to prevent

* An ombudsman is a counselor or mediator, often involved in helping to resolve legal issues.

users in other countries from downloading and compiling the BladeEnc source code from the Swedish server, and it's not difficult to find binaries on servers in other countries, created from that source code. At this writing, the BladeEnc issue is officially unresolved. Keep on eye on *BladeEnc.cjb.net* for updates.

It is likely that similar international copyright questions will arise in the future, involving other codecs, technologies, and products. Unfortunately, this book can suggest no easy answers. It's a wild time for legal professionals these days.

The MP3 Patent

http://www.iis.fhg.de/amm/legal/ A great deal of confusion and misinformation surrounds the question of the MP3 patent itself, as it's involved in some way in every MP3 encoder and player available. Because many of these utilities are free for the download, many people automatically assume that MP3 must be a free specification. In fact, developers *can* freely download encoder source code offered by the International Standards Organization. Encoders based on this source are known as ISO-based encoders, and are referred to throughout this book.

It's worth noting that the ISO source is only a set of guidelines or suggestions. The ultimate goal of the source code is to help developers create applications that are capable of producing an MP3-compliant bitstream. The ISO code will get you there, but it won't necessarily offer the most efficient route. In any case, just because the source is available doesn't mean that developers are free to release utilities based on that source. If you compile and distribute binaries from the ISO source code, you may owe a fee to Fraunhofer IIS. The Fraunhofer IIS Institute, who spent nearly a decade developing MP3, demands that developers who have released ISO-based encoders pay licensing fees back to Fraunhofer IIS.

MP3 is an ISO-approved standard, but the intellectual property behind it is still owned and patented by the Fraunhofer IIS Institute (patents were registered in Germany in 1989 and again in the U.S. in 1996, while Thomson Multimedia owns a different set of patents on MP3). MP3 is not, at this writing, patented in other countries. In exchange for use of their intellectual property, Fraunhofer IIS (and THOMSON, a company with which Fraunhofer IIS partners) ask for something back from those who use the MP3 spec in their applications.

However, there's a difference between what they ask for and what they demand. If you merely distribute a free decoder, Fraunhofer IIS asks for nothing in return. If you sell your decoder, Fraunhofer IIS asks that you pay them $1.00 per copy.* However, Fraunhofer IIS's patent does not cover decoding, so they can't demand

* Although Fraunhofer IIS does not assert their patent against decoders of which less than 10,000 units have been sold.

this payment; it's voluntary on behalf of the developer. If you create a hardware-based decoder, the price doubles to $2.00 per unit, but again, this is a request for a voluntary payment. If users are required to pay for songs or tracks, Fraunhofer IIS demands that the agency offering the song for download pay $.01 per track.

The situation gets a little different with encoders, as this is where Fraunhofer IIS's patent kicks into gear. All distributors of MP3 encoders are required to pay patent licensing fees according to the schedule in Table 7-1. Developers of MP3 encoders must be prepared to pay Fraunhofer IIS substantial fees in exchange for the use of their intellectual property. Free encoders are not exempted from the responsibility to pay licensing fees.

Table 7-1. Fraunhofer IIS Licensing Fees

Number of encoders shipped	Fee per encoder
1–1,000	$25.00 per unit
1,001–2,000	$20.00 per unit
2,001–3,000	$15.00 per unit
3,001–10,000	$10.00 per unit
10,001–100,000	$5.00 per unit
More than 100,000	$2.50 per unit
PLUS a flat yearly fee of $15,000	

Some people think that it would be possible to create an encoder that doesn't use the scheme laid out in the ISO sources and thus circumvent the requirement to pay licensing fees. However, the patent is on tools that create an MP3-compliant bitstream. It doesn't matter how you get there; if your encoder can create MP3 files, you still owe money to Fraunhofer and THOMSON.

You know that "freeware" encoder you've been using for the last six months? Unless the developer has really deep pockets and has been paying Fraunhofer IIS out of their personal bank account, the vendor is probably in violation of the law. Typically, only encoders released by established companies such as MusicMatch, Real, Microsoft, or Be are fully paid up and licensed. Pretty much any freeware encoder you download and use is technically cheating Fraunhofer IIS out of the money they ask for the intellectual property they've contributed to the standard.

If a content provider charges for the download of songs, and if the encoder used to create those songs is covered by the Fraunhofer IIS patent (and it most surely is), Fraunhofer IIS is within their rights to demand (not just request) a payment. The required payment is currently $.01 per song.

The Serial Copy Management System (SCMS)

SCMS is described in the discussion of the AHRA earlier in this chapter. In essence, SCMS is a means by which manufacturers of digital audio recording devices can earn immunity from copyright infringement lawsuits if users of those devices distribute illegally copied music. An SCMS-compliant device must:

- Be registered with the Copyright Office
- Pay a royalty fee to the RIAA
- Disallow the creation of second-generation copies

SCMS has deeply ranging implications for MP3 recording devices (and, claims the Electronic Frontier Foundation, may have repercussions for Internet distribution as well). Many people have philosophical problems with the fact that SCMS presumes guilt on the part of the person using the device. An SCMS-compliant device places what many feel are unfair burdens on legitimate artists by disallowing them from copying their work around as they see fit. In this regard, SCMS rewards artists who are already signed to labels and penalizes unsigned artists who will not benefit from the royalty payment flowback; the government thus becomes a virtual agent for large artists and an adversary of small artists. SCMS also makes some hardware vendors jump through hoops to guarantee compliance and keep themselves out of hot water, and inconveniences users of those devices in several ways.

Nevertheless, the alternative is untenable to both large artists and labels, who do indeed stand to lose revenue if perfect and unlimited digital copying were to be allowed. Despite the best efforts of SCMS, however, the Internet has all but dwarfed concerns attached to external devices. The Internet is not a recording device, but it allows for nearly unfettered copying of copyrighted works, and cannot be as easily tamed as can hardware vendors.

The Secure Digital Music Initiative (SDMI)

www.sdmi.org In 1999, the recording industry decided to fight back against the MP3 phenomenon in a big way. More than 50 companies from all aspects of the recording industry, including the Recording Industry Association of Japan, tech industry leaders, the International Federation of the Phonographic Industry, and the five major record labels (BMG, EMI, Sony, Universal, and Warner) banded together to form what has become known as the Secure Digital Music Initiative, or SDMI. The group, realizing that MP3 was not a passing fad, that digital music distribution was not going to go away, and that competition among the many "secure" formats offered by various companies was likely to lead to chaos sooner than it was going to lead to solutions, decided to approach the problem from another angle and get copy protection mechanisms built into a much wider array of devices than those covered by SCMS. Because the Internet phenomenon is bigger than any one organization, and clearly can't be cut off at the knees, SDMI goes after what it can control: All of the mechanical means by which digital audio data enters and leaves devices, including portable devices, home stereo components, CDs, and yes, personal computers.

It's important to understand that SDMI does not represent any single technology or implementation. Rather, it's a framework in which many (possibly competing) technologies can interoperate within a single system. The goal, as stated by the industry, is to create technologies that will:

> ...respect the usage rules embedded in music by its creators... The Secure Digital Music Initiative brings together the worldwide recording industry and technology companies to develop an open, interoperable architecture and specification for digital music security. The specification will answer consumer demand for convenient accessibility to quality digital music, enable copyright protection for artists' work, and will enable technology and music companies to build successful businesses.

That's the rhetoric, but what's the reality? What exactly does SDMI mean for MP3 users? At this writing, it's a little difficult to say exactly, as full details of the SDMI implementation have not yet been fully worked out. What we do know, however, is that your portable MP3 player, your PC, your sound card, and your CD player will all conspire to control what you can and cannot copy into other audio formats.

SDMI faces an immense uphill battle. They must not only convince hardware manufacturers to implement SDMI capabilities in the equipment they produce, but they must convince consumers to adopt this equipment. The first part is easier than the second, because the forces behind SDMI have all the money and all the clout. They will be able to force legislation and wield threats. Convincing consumers that SDMI is a good thing will be a far more difficult task, as consumers aren't stupid, they know what they want, and they have already voted on unfettered MP3, even though a variety of security-minded formats already exist (see Chapter 9, *Competing Codecs and Other File Formats*). Consumers have a tendency to sensibly adopt the simpler, more open solution when provided with a choice. The battle between DVD and DIVX, which was formally won by DVD when DIVX withdrew from the battle in mid-1999, provides ample evidence of this tendency.

Phases I and II

However, the SDMI team has a pretty powerful ace up its sleeve. Because of the way SDMI is slated for roll-out, consumers will eventually find themselves unable to listen to the latest content without an SDMI-capable device. As much as we might like to fight the power, it may not be as simple as saying "I won't buy any SDMI-compliant hardware." The SDMI roll-out is scheduled to occur in two stages.

During Phase I, SDMI-compliant devices will be able to play either unprotected or SDMI-compliant music. This phase mostly represents the seeding of the market with SDMI-compliant devices, and consumers won't notice that anything has changed. However, all Phase I hardware and software will include a "time bomb." When a certain date arrives, Phase II will kick in and users will be prompted by their devices and applications to download new "screening" software. Without this screening update, users will not be able to play pirated versions of music created on Phase II SDMI hardware. Users will, however, still be able to play older unprotected music, just as they do now.

That scenario sounds fairly painless on the surface. After all, even Phase II SDMI devices will be able to play plain vanilla, unprotected MP3 files. However, consider the fact that computer hardware goes out of date rather quickly. Eventually, people will discard their old hardware and replace it (perhaps unwittingly) with newer, SDMI-compliant hardware. Copies of digital music made on that hardware will not be playable by other users of SDMI playback devices. And because record labels will be including digital watermarks on every track of every new CD they create, devices and applications will be able to easily distinguish between "marked" and "unmarked" content.

Over time, the pace of technological change will result in fewer and fewer people having access to non-SDMI hardware. Zillions of existing MP3 files will still be playable, but new music played on new hardware will soon be pirate-proof.

Holes in the facade

Or so the industry would have us believe. Of course there's always a way around, over, or under any barrier, and it probably won't be long before myriad hacks and workarounds are created to circumvent SDMI, both through hardware and through software. Present the computing world with a cracking challenge, and you've virtually thrown down the gauntlet. Hackers will see SDMI as just another challenge, and will no doubt discover myriad ways to circumvent SDMI's protection mechanisms. Illegal? Perhaps, but the fact remains that circumventions will exist. With the rising popularity of Linux, computer jocks find themselves in an environment that's not accountable to any corporation, and lacking Microsoft-style supervision and/or cooperation with the SDMI group. Linux machines may in fact become a sort of haven for users wanting to circumvent SDMI. And since Linux is also a server platform, SDMI-protected music will quickly find its way back out of Linux boxes and onto the Internet, in unprotected format.*

Of course, there are philosophical and political problems with SDMI as well. While SDMI claims to represent the interests of artists, membership in the SDMI working group has been both exclusive and expensive. No small-time artist, and few independent labels could afford the membership fee required to have their voice heard in SDMI decision-making sessions. One must keep in mind, however, that use of watermarks on audio CDs is not going to be forced on labels. If they choose to release compact discs without watermarks, they're free to do so.

In fact, some believe that SDMI is unfeasible at a technical level. For example, in order for SDMI to work on computers, each machine must be uniquely identifiable. But recall the public outcry that went up last year over the unique identifying string built into some Pentium chips. There are clearly some significant privacy issues at stake here. If files are tied to unique machine IDs, you run into another hurdle every time you upgrade your processor or motherboard (depending on where the identifying string is stored). Will you lose and have to replace all of your music? Will you have to re-register your machine with some central authority? There's a potential Pandora's box here.

Of course, any barrier that can be erected by the industry can be transcended by the clever user, and the organizations behind SDMI realize this. Their only hope is to erect a barrier high enough that the majority of consumers won't be bothered to try and jump over it. If they can block the majority of the population from easily pirating music, they will be better off than they are now (or, at least, that's their presumption).

Will SDMI fly, or will it die an ugly death at the hands of consumers? After all, it was consumers who killed DIVX in its crib by mobilizing the Internet to mass protest and educating millions of potential adopters. Whether you believe SDMI is good or bad, there's no question it faces a long and winding road to mass acceptance. However, remember too that DIVX was offered as an *option* to consumers. SDMI is not being put forth as an option; it may be forced upon the public, who will have no choice but to adopt it when their hardware goes out of date. If the forces behind SDMI are able to convince enough manufacturers to stand behind them, that is.

* I'm not suggesting that SDMI circumvention will only occur in the Linux world; only that many hackers gravitate toward Linux and that Linux does not live under corporate control like other operating systems do. SDMI circumvention techniques will undoubtedly exist on all platforms.

Technical difficulties and possible solutions

Thanks to technical editor Bruno Prior for his contributions to this section.

One of the proposed linchpins of SDMI is that it will allow users to create only so many copies of a given track, even for personal use. For example, you may be allowed to make four copies of a track: One for your computer, one for your home stereo, one for your car stereo, and one for your portable device. So what happens if you have two cars? What happens if the hard drive on your home stereo crashes? What if you want to store multiple copies of your music at different bitrates, or encoded with different encoders, for use in different situations? Theoretically, it should be possible to submit the original back to some governing organization and "check out" a replacement copy. But if you have more than four devices, you're going to have to do this over and over again, potentially each time you leave the house. Clearly, nobody is going to have the patience for that. At a certain threshold, such a plan becomes too inconvenient to be taken seriously.

Faced with barriers like these, customers are likely to seek out illegal circumvention techniques, potentially forcing law-abiding citizens to become scofflaws. Barriers this high will stifle the digital music distribution industry, rather than promote it.

More palatable—if it can be made easy enough to use—would be some form of public/private key system, similar to systems used in privacy protection mechanisms such as Pretty Good Privacy. A purchaser of music would submit their identity to an organization like Verisign, who would issue a public key to that person. The vendor would watermark and encrypt the audio tracks with the purchaser's public key before forwarding the encrypted file to the purchaser. The purchaser would then be able to decrypt the file with their private key and save the decrypted (but still watermarked) file to disk. Compliant software/hardware would require the availability of the private key to match the public key in the watermark in order to play the track.

Under such a scheme, users could create as many copies, permanent or temporary, as they liked, but would only be able to play them in the presence of the private key. Distributing one's private key to other users is not something most people would do, and any attempt to allow wholesale piracy by issuing protected files on the Internet would be easily traceable through the public key in the watermark. Any public keys so discovered could have their certification instantly revoked by the certification authority.

Such a schema could be taken even further by taking the 16-bit original and stripping every other bit to yield a lower-quality 8-bit version, which could be offered for free to the public as a sample download. Consumers who liked the track and wanted to purchase a high-quality version could submit their personal ID and receive the stripped bits, watermarked and encrypted, to rebuild the track at its original quality level. This would increase the attractiveness to the consumer and reduce the encrypting overhead for the server.

As with the officially proposed scheme, there are inconveniences to the user inherent in a key system as well, and it would be essential that all of the software and related tools required for such transactions be utterly simple and painless to use. However, such a scheme would not impose arbitrary limitations on the consumer's ability to create an unlimited number of digital copies of the music they already own.

More information

For more information on intellectual property, copyright, SDMI, and the political ramifications of MP3 issues in general, please visit the following sites, keeping in mind that most of these sites have their own agendas to protect. The News and Opinion sections of MP3.com are often good sources for counter-opinions:

10 Big Myths About Copyright Explained
www.templetons.com/brad/copymyths.html

The Digital Media Association (DiMA)
www.digmedia.org

The Electronic Frontier Foundation (EFF)
www.eff.org

Fraunhofer's take on Intellectual Copyright issues
www.iis.fhg.de/amm/techinf/ipmp/

The Recording Industry Association of America (RIAA)
www.riaa.com

SDMI: Boom or Bust for the Music Industry?
www.musicdish.com/downloads/sdmireport.pdf

Secure Digital Music Initiative (SDMI)
www.sdmi.org

SoundByting (RIAA subsidiary site)
www.soundbyting.com

World Intellectual Property Organization
www.wipo.org