

Internet connection). The CD volume is then mounted with the artist and album name, and all tracks on the disc appear with their actual song names in the Tracker. This means that BeOS ripping/encoding tools don't necessarily have to do CDDDB lookups of their own—they can just extract the artist, album, and track names directly from the filesystem.

If you want to be really hardcore about it, you can rip tracks from audio CDs without the help of `cdda-fs`, by using the system's built-in "play" command (which offers a "save track to file" option), or use the BeOS port of `cdparanoia`, described earlier in this chapter.



Be has licensed the MPEG codec from Fraunhofer and Thomson, and includes MP3 encoders and decoders in the system. This means that, in addition to ease of ripping from CD, you can save directly to MP3 from any BeOS audio mixer/editor/sequencer application that works with the media kit.

CDPlayer and SoundRecorder

If you prefer to do this graphically, or just want to rip portions of tracks, the graphical *CDPlayer* application included with BeOS also has a built-in ripper and encoder. Just click the floppy disk icon in *CDPlayer*'s interface, select a track number, and use the triangular sliders to select the portion of the track you want to rip (the default setting rips the entire track). Select an output format to save your selection as WAV, AIFF, RAW, or directly to MP3. Similarly, the system's built-in *SoundRecorder* application will let you crop arbitrary segments of audio from any track and right-drag them to the Desktop. Choose an output format from the context menu and you'll be cropping and encoding audio to MP3 simultaneously.

MediaPede's UltraEncode

www.mediapede.com The easiest to use of the BeOS ripper/encoder solutions is an all-in-one package from MediaPede called *UltraEncode*, which sports an intuitive interface and fast encoding thanks to Be's inclusion of an MP3 encoder in the system. Usage is straightforward: Establish your CDDDB server, bitrate, and destination directory in the preferences, insert a CD, and its track names will appear in the Extractor Queue. Establish a genre and year for the CD from the Settings menu, and double-click any track name for which you want to edit details.

Check off the tracks you want to encode, or tap `Alt+A` to select all or `Alt+D` to select none. Click the Start button and ripping/encoding will take place simultaneously (*UltraEncode* is heavily multithreaded). You can show/hide any of the interface panels by clicking the "switch" widgets at the right of the interface (see Figure 5-6).

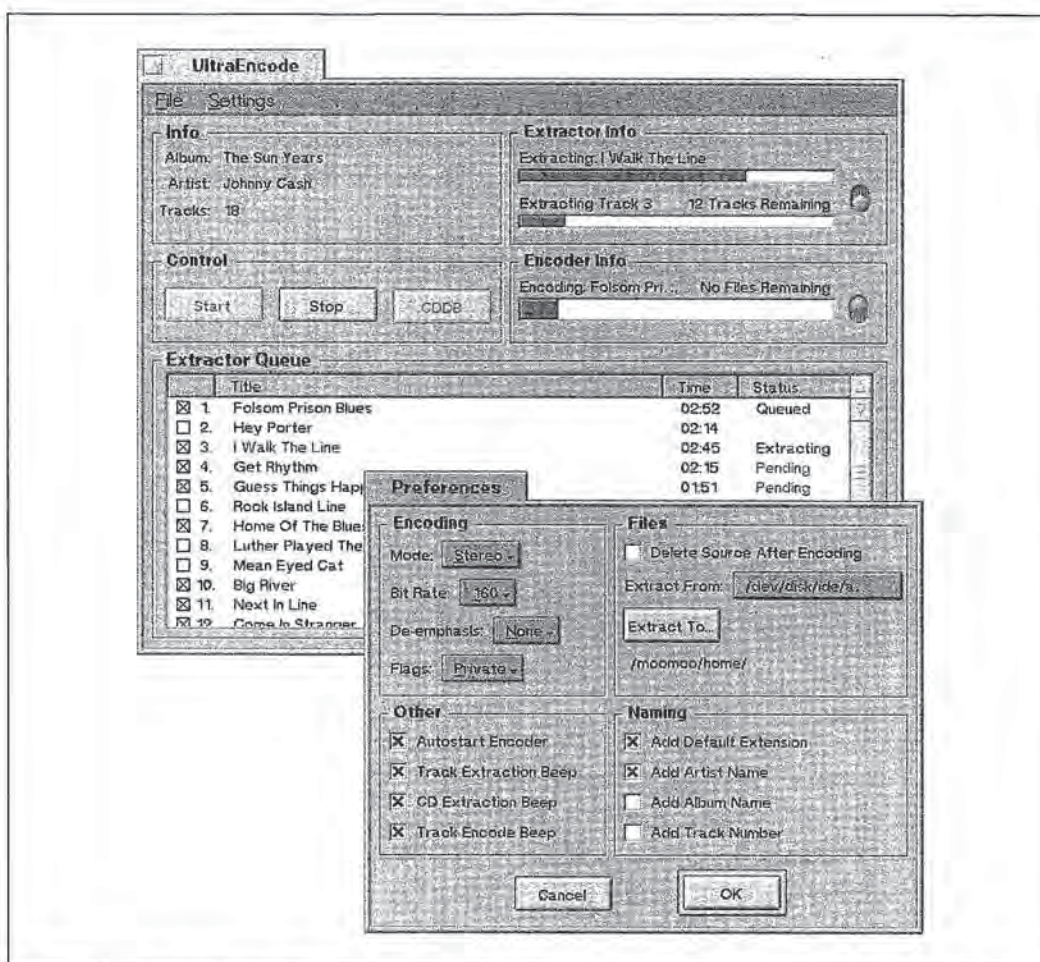


Figure 5-6. MediaPede offers an all-in-one ripping and encoding package for BeOS users

RipEnc

www.betips.net/software/ RipEnc for BeOS is a distant cousin of the Linux version mentioned above. The BeOS version (by this book's author) started life as a straight port of the Linux version, but was later gutted and rewritten from scratch to take advantage of cdda-fs. The BeOS version is therefore "ripless"—tracks are encoded directly as they appear in the Tracker, without being extracted from the CD first. Like the Linux version, the BeOS version is a "frontend" to a small collection of command-line tools, including id3ren (covered in Chapter 4), the GOGO encoder (covered earlier in this chapter), and a tool called id3attr, which copies ID3 tag information into BeOS filesystem attributes which can be queried on to create custom playlists quickly. RipEnc for BeOS also makes use of "folder templates" so that generated folders full of MP3 files are pre-organized in a customizable fashion, showing ID3 tags as attributes in the Tracker (as shown in Figure 5-6). RipEnc also supports optional manual track naming, a "play when done" option, variable bit rate mode, CD automounting, and other goodies.



The nice thing about using frontends such as RipEnc for Linux or BeOS is the fact that the user can choose their favorite tools to be used with the system. For example, you can choose to use a different encoder while still retaining the same overall encoding framework. And because shell scripts are easy to edit, users can dig in and make modifications if they wish.

Other popular ripping and encoding tools for BeOS include RipChord, KizunoMP, GOGO Gadget, 8hz, and KonaKoder. You'll find these tools and more at www.be.com/beware/ or at www.bebits.com.

Ripping from Other Sources

Of course, the audio you want to encode isn't always going to originate on audio CDs. Many of us still have huge music collections stored mostly on LPs, cassettes, DAT, 8-tracks, reel-to-reels, and possibly other media. MP3 represents a great opportunity to finally create a permanent digital archive of these collections... or at least the best of them (most people would have to be horribly masochistic to dedicate themselves to the digitization of every track they own). The good news is that encoding your existing collection in this way is do-able. The bad news is that it's going to require quite a bit more manual labor than encoding from CD. Since you can't stick an LP into your computer and let it twirl, you've got to go to the extra step of patching your home stereo into your sound card and manually cueing everything up properly. And of course, there are no batch-ripping/encoding solutions available for digitizing 8-track tape collections. Finally, you may have to separate songs manually when encoding from analog sources—LPs don't have any real awareness of when one track ends and the next begins. Without special software (see the following section), you'll have to do all of that yourself in an audio application prior to encoding.



The term "ripping" doesn't exactly apply when talking about sources other than compact disc—nothing needs to be ripped since the source isn't hidden, as it is with .CDA files. We use the term "ripping" in a somewhat looser sense in this chapter, though it might be more accurate to refer to "extraction."

If you're a recording artist, you'll probably already have your repertoire stored on tape, DAT, or another medium, so this section applies to you as well. Of course, it's possible to connect a microphone or other instrument directly into your sound

card and save the signal direct to disk, and you may even have a fancy-schmancy sound card that handles multiple inputs and lets you do all kinds of preprocessing with dedicated DSPs, etc. But even if you are a digital-friendly musician, you're still going to want to do all of your mixing and mastering in dedicated multi-track audio software, creating a final stereo mix before going to MP3. In other words, while recording live to MP3 is a possibility, it's not often done unless you're capturing a live performance—and even then, you'll probably want to store that unencoded first.



When recording from LP, cassette, or other analog sources that don't have the same crispness at the high end, try using an equalizer to boost the higher frequencies by a couple of dB before encoding. This can help to alleviate some of the "swishy" effects for which MP3 is known. However, increasing high frequencies on LPs and tapes will also result in more noise, so there's a trade-off here. As always, you'll get better results from analog sources by using higher bitrates.

Direct to Hard Disk

As described in Chapter 4, a lot of audio hardware, including most sound cards, deal with two possible input specifications: Line level and Mic level. Mic input is used for amplifying low level input, which mostly consists of that coming from microphones. Line input handles signal from audio equipment with line-level outputs, such as CD players, tape decks, tuners, and so on. Most sound cards offer two or more input jacks, one Mic and one Line. If you're not sure which input jack on your sound card is which, consult its documentation. Note that phono cartridges need a unique amplification stage that converts their output to line level.

There are two ways to connect audio hardware to a sound card. You can either connect an instrument, microphone, or a stereo component's output directly to the appropriate input jack on your sound card, or you can run a line from your stereo amplifier/receiver's output terminal to your sound card (if your amplifier has an output jack—not all do). With the first method, you'll have the very slight advantage of eliminating your stereo's circuitry from the recording chain for slightly better input quality, but have to deal with the physical awkwardness of yanking stereo equipment and placing it near your computer. The second method incurs a little extra circuitry, but may be more convenient. In all cases, you'll get better results whenever you can minimize the length of cable runs.

Record What You Hear

Note that most of the same techniques outlined above also apply if you want to record other audio streams moving through your system, such as those coming from Internet broadcasts (any format). The only difference is that rather than fiddling with cables and jacks, you'll need to set your system's audio mixer to trap all signals. Look for a setting labeled something like "Speaker Out" or "Record what you hear." This is different in every operating system, and some sound card utility installations override your system's default settings, so you may need to use your sound card's software rather than the operating system's mixer to establish this setting. In any case, it should be possible to save any incoming signal to WAV or AIFF, clean it up in an audio editing application, and encode the resulting audio file to MP3 normally. Some audio pirates use this technique to circumvent protection schemes built into Internet broadcast formats.



Alternatively, you can use the tape output jacks on your amplifier, which will let you use the amp's switching facilities (such as the Tape Monitor button). If you do this, the volume control on the amplifier won't affect the level of the signal being sent out, so you can change the volume of the music you're hearing without affecting the level of the recording being made.

Once you've got your equipment hooked up, you need to be able to save the signal coming into your sound card to hard drive. While most operating systems come with basic audio recording software, you'll probably find that this software isn't quite up to the task. Since you may need to edit out silent spaces at the beginning and end of various tracks, or apply noise-masking, equalization, or other effects to the tracks you record prior to encoding, I strongly recommend looking into the third-party audio software options available for your platform. Usage of general audio recording software is outside the scope of a book on MP3, but popular applications include CoolEdit Pro or SoundForge for Windows, Pebbles or 3dsound for BeOS, Deck or Peak for the Macintosh, and fOX Mixer for Linux. See the section "Encoding from Analog Sources" later for more information.



If you do want to encode directly to MP3 from audio input, some of the more complete tools—such as MusicMatch Jukebox—are capable of doing so. In MusicMatch, pull down Options → Recorder → Source and choose from Line In, Mic, or System Mixer. Set your encoding preferences as usual, hit Record, and you'll be encoding from that input.

About S/PDIF

If you want to make sure your transfers stay 100% digital, or if you need to get audio data out of digital devices such as DAT players, look for a sound card with a Sony/Philips Digital Interface Format (S/PDIF) port, which usually looks like a standard RCA jack and can be connected directly to the S/PDIF port on the device. You'll also need special software (generally supplied with the digital device) to manage the transfer. All transfers will be 100% perfect; none of the issues that make CD ripping sometimes problematic affect S/PDIF transfer.

Ripping from DVD

The audio tracks on most DVD disks are stored in the copy-protected VOB format to prevent ripping. If you should be in possession of a DVD disk to which you have legal copyright access, you can bypass this copy protection by going through the sound card—just plug the DVD player's output directly into the sound card and record from that channel with an audio-editing application. If you want to go for higher quality, run the S/PDIF output of the sound card into a digital amplifier, then from that amplifier's output to an external CD recorder. You can take the analog output from a DVD player, run it through analog-to-digital conversion to output a WAV file, and encode that. If you have a digital sound card, you can take the digital output from the DVD player, run it through the sound card's digital input, and generate a WAV from that.

The fact that a group of Linux hackers had successfully unlocked the DVD security mechanism in late 1999 may mean that tools for getting signal from DVD to hard disk may become available by the time you read this; it was too early to tell at this writing

Encoding from Analog Sources

Many thanks to technical editor Bruno Prior for his contribution of this section.

There are two basic ways to encode from analog sources such as LP or tape: Either use an "all-in-one" solution to encode directly from the audio source to MP3, or record the source to an uncompressed audio file (WAV, AIFF, etc.), pre-process in a dedicated audio editor, and compress it with an MP3 encoder. While it may seem at first like all-in-one tools would be the easiest route, this often turns out not to be the case, for a variety of reasons. For the purposes of illustration, we'll look at one example of each type.

MusicMatch Jukebox

www.musicmatch.com We've already covered the basics of MusicMatch Jukebox in Chapter 3, but here we're going to take advantage of one of Jukebox's more advanced features—its capability of encoding from your sound card's line-in jack. To be more consumer-friendly, MusicMatch refers to encoding processes as “recording,” so we'll use that term here, even though it technically isn't quite correct.

To enable the Recorder, make sure the “Show Recorder” option is selected under Options → View. To configure the Recorder for line input, navigate to the Recorder configuration page via Options → Recorder → Settings and select Line-in from the Recorder Source picklist. Select your Recording quality (bitrate) from the options on the left. Click the Songs Directory button and specify the folder and filename formats under which your recordings should be stored.

You may also want to click the Advanced button to specify whether tracks should fade in or out, which can be useful if the actual fade in or out on the source material is corrupted with vinyl crackle or tape hiss. You can also choose to offset the start of each track, (which is useful to account for gaps between tracks), and to normalize your recordings.



The MP3 encoding selection box will let you choose between Joint Stereo (the default), Stereo, and Mono. As described earlier, you probably want to select Stereo here, unless you want to keep file sizes down. This can be particularly important when recording from tape, since left and right stereo channels on some tapes can be slightly out-of-phase, especially with older music. Combining the lower frequencies of out-of-phase channels, as Joint Stereo does, can produce significant distortion—the only way to avoid this is to use simple Stereo encoding.

The most important section in Advanced dialog is the Auto Song Detect option, which is intended to let MusicMatch Jukebox detect gaps between songs so you don't have to stop and start recording for each track on an LP or tape. If you know the approximate or minimum length of gaps between songs you're recording, you can help MusicMatch make a better educated guess as to where songs begin and end.

You'll notice that the Jukebox recording interface now looks slightly different than it does when ripping from a CD. Sample album title and artist names will have been added, and only one track will be displayed, awaiting the name to be entered. Enter real album and artist names and give a meaningful name to the first track. With your turntable or tape deck attached to your machine as described earlier, start the music source playing and hit Start on the Recorder. If you chose

the Auto Song Detect option, it will attempt to break the input into tracks wherever it finds a gap in the music. If it doesn't find gaps, Jukebox will output the entire audio stream until you press Stop.

There are a number of problems you may encounter when recording from an analog source. For example, Jukebox does not start recording immediately when you press Start. Instead, it listens for the start of the song before recording begins. This is fine if the song starts assertively, but if it fades in or starts with a quiet intro, you'll probably lose the start of the song. You can get around this by selecting the Delayed Record option in Recorder Settings and then specifying Start Immediately. The side effect of this technique is that you'll have to stop and start recording for each track, whether or not you have selected Auto Song Detect.

Regardless whether you have Start Immediately enabled, Auto Song Detect may introduce other problems. For instance, ASD simply looks for gaps longer than the specified Gap Length which do not contain sound levels above the specified Gap Level. Unfortunately, most analog sources contain quite a bit of noise, so you'll have to specify a fairly generous Gap Level—probably at least 10%. But if your music encompasses a significant dynamic range, you may end up with split tracks, as Jukebox may mistake quiet passages for song gaps (try recording Queen's "Bohemian Rhapsody" from vinyl for a splendid example of this). Fade-outs may also be prematurely truncated.

Even if you can get past these gotchas, you'll still have some work to do when recording is complete. You now have a series of MP3 tracks, but only the first one will have a meaningful name, while the remaining tracks will have names like "Line-in track 2," etc. You're going to have to rename the files and edit the ID3 tags by hand for them to be of any use. With any luck, a future version of Jukebox will offer a more intelligent interface for handling this in advance of recording.

While Jukebox does a pretty good job as far as it goes, you may end up deciding that you need a more sophisticated tool for reasons like this. LineRipper, for example, may be better suited to meet your needs.

LineRipper

leo.worldonline.es/mpicarth/ LineRipper exists for the sole purpose of encoding from Line-in (although it is also capable of conventional encoding from WAV files, and supports a variety of encoders, including LAME and GOGO. Conveniently, LineRipper lets you specify track names for all the tracks to be recorded beforehand. More importantly, LineRipper lets you specify different artists for each track, which gets around problems commonly encountered by encoders trying to deal with compilation albums. For the real power user, LineRipper can separate tracks either by listening for gaps or by reading a table of track lengths input by the user before encoding begins. While this takes more effort up front, it offers a better chance of accurate track separation.

LineRipper is clearly superior to Jukebox for the purpose of encoding from Line-in, but it has its own weaknesses. Even if you do type in track lengths beforehand, some slippage is still likely to occur due to differing gap lengths. If you use the auto-detect option, there's still no way to avoid the problem of truncating tracks that fade in or out when there's significant noise in the signal, and there's no way to avoid noise from sources like vinyl or tape unless you preprocess the output. This is no fault of LineRipper, but a fundamental weakness in the process of encoding from analog sources in general.

Notes on manual track separation

It may seem like a lot of effort, but in the end, the traditional route of recording to one giant WAV file, processing and splitting the WAV, and then encoding the processed tracks can be the most effective technique for creating MP3s from your tape and vinyl collection. Not that this process is not without its problems either.

The first difficulty you're likely to encounter is the fact that the audio-editing facilities which come standard with your system may not be adequate for the job. Sound Recorder (the bundled Windows recorder), for instance, can only handle around two minutes of music, which won't get you very far (the audio recorder bundled with BeOS, however, is extremely well suited for tasks like this, and can output directly to MP3; there are a number of third-party utilities for MacOS and Linux that will do the job, though I wasn't aware of any that could save directly to MP3). For most purposes, you'll need a program to record audio from Line-in. If you have a Creative Labs sound card, you should have a program on your system called Wave Studio, which goes far beyond the capabilities of Windows' Sound Recorder. Other sound cards often come bundled with similar audio editors.

To really get all the power features, you'll have to lay out the cash for a pro or pseudo-pro sound editing program. Unfortunately, the leading contenders, such as Sound Forge and CoolEdit Pro are not cheap, at several hundred dollars each. Fortunately, cheaper contenders are starting to emerge. One attractive option is CoolEdit 2000 (www.syntrillium.com/cooledit/) which currently costs a more reasonable \$69. An optional Audio Cleanup plug-in for CoolEdit 2000 is also available, which includes the features you need for reducing clicks, crackles, and hums from analog audio. However, the plug-in is extremely slow, and doesn't offer a visual preview. A popular and even cheaper alternative is GoldWave (www.goldwave.com), a \$40 shareware audio editor. GoldWave includes noise reduction, but click/pop/crackle removal still has to be performed by an external program.



Remember that these giant WAV files are going to consume large amounts of disk space. One 30-minute album side will take up around 300 MB at the standard 44.1kHz, 16 bit (and you probably don't want to go lower than that for your quality MP3 collection). A reasonably fast hard drive can also be useful when dealing with lots of uncompressed audio.

Notes on noise reduction

Assuming the recording process goes well, you're left with a 300 MB, crackly, hissy uncompressed audio file. The next step is to lose the crackles, hisses, etc. In CoolEdit 2000 (or other, more expensive editors like CoolEdit Pro or Sound Forge), you can probably pass the file through the included Audio Cleanup process. If your editor doesn't include such a feature, a wide variety of programs are available to do this. Some of the better ones are:

Sound Laundry (www.algorithmix.com)

The Compact Edition offers limited functionality but has the easiest interface of any of these programs, while the full version has a confusing interface but is very flexible and powerful.

Groove Mechanic (www3.bc.sympatico.ca/badgerbytes/groove/)

Very good output with the default values, but not as flexible as most.

Dart Pro 32 (www.tracertek.com)

Highly configurable but slightly confusing user interface and somewhat sensitive to source file format.

Regardless of the program you choose, play around with the options for reducing click/crackle and noise/hiss until you're getting the cleanest sound possible without removing any of the actual audio. Keep in mind that it's very easy to throw the baby out with the bathwater here—remove too much signal and you'll affect the ambience of the whole piece. Then pass the file through the filter or filters and save the clean output as a new file.



A good way to make sure you're removing only the noise and not the signal (if your program supports it) is to listen to the sound being removed, rather than to the audio that remains.

Splitting large files

The next step is to chop up the clean file into its constituent tracks by selecting the in-point and out-point of a track, copying the selected area to the clipboard, and pasting the results into a new track. Most editors offer a visual image of the file, so it's easy to see the song gaps as "troughs" on the display. Jump to the gap points, zoom in, and listen carefully to make sure you're selecting exactly the right in- and out-points for each track.

Once you've got a set of clean, uncompressed tracks, all that's left is to encode them to MP3 by passing the files through your favorite encoder. Most encoders will not offer the option to tag the resulting MP3s, as they can't interface raw audio with the CDDb and therefore do not have the information they need. You'll have to insert your ID3 tags manually with a separate ID3 tagger or through your favorite MP3 player, if it includes its own tag editor.

If you named the files sensibly, try a tool such as `id3ren` (see Chapter 4), which can extract track title, artist, album, and other data from the filename and use that to create the tags automatically.



If splitting and encoding files manually sounds like a lot of work, a tool that will help to do this in one step is LP Ripper (www.cfbssoftware.com.au). Tell the program how many tracks are on the recording and it will split the file for you by checking for perceived gaps. You can adjust the start and end times of each track to a sensitivity of 0.1 second, and you can enter track titles beforehand, which LP Ripper will use to name the files. Some basic manual editing (such as removing leading and trailing noise) is advisable before passing the file to LP Ripper, but this tool can save a good deal of time and effort when dealing with giant WAV files. LP Ripper's greatest weakness is that it won't let you jump to the end of a track to adjust the end time. If you want to edit the end time of each track precisely, it's probably quicker to do it manually.

Roll Your Own Compact Discs

When compact discs first appeared, they were "black boxes"; unlike LPs or cassettes, few people understood how they worked. In the mid-'90s, writeable CDs appeared for the first time, enabling anyone to create their own audio or data CDs at home... as long as they were sufficiently wealthy. By the end of the '90s, however, the cost of CD-R (the "R" is for "recordable") devices had fallen to well within the casual user's reach, and the cost of blank CDs dropped to the point where CD-R had a lower price-per-megabyte than any other storage medium.

MP3 collections typify the kind of problem CD-R was meant to solve: What happens when the amount of data you want to keep around exceeds the amount of storage space available on your system's hard drives? True, the price of hard drives has plummeted as well, but no matter how big a hard drive you install, it won't be long before your MP3 collection is even bigger.

However, relying on compact discs as a primary storage medium probably isn't the way you want to go either. With the price of hard drives continuing to go down, some people are surprised to discover that storing large MP3 collections on hard drives is actually *cheaper* than storing them on compact disc. Let's do the math. At this writing, 28 GB hard drives could be had for around \$8/GB. Assuming you encode at 128 kbps, you'll need around one megabyte per minute of music. Now let's assume a baseline price of a CD-R drive at \$200 and \$1 for each disc. One 28 GB disk will cost around \$224 and hold around 470 hours of music. Storing the same number of MP3s on CD-R will require around 44 discs, at a cost (including the drive) of \$244. Storing the same number of tracks in CD audio format will require around 380 discs, at a cost (including the drive) of \$580. So to store approximately 500 hours of music, hard disk is the best cost option. On the other hand, if you double this (i.e., 1,000 hours of music), then storing MP3s on CD-R looks like the best option. But by then, the inconvenience of having to search through 90 CDs to find the track you want will probably outweigh the marginal cost benefits.

If cost is all that counts, then the balance point (where CD-R overtakes hard-disks in terms of value) is currently at just over 500 hours of music. But unless you absolutely require portability, having all your tracks on CD-R as opposed to hard disk is going to be extremely inconvenient. And remember that CD-R is not editable. Keep these considerations in mind before devoting a ton of energy and money into building a giant collection of hand-rolled compact discs.

CDA vs. Data

There are two ways to approach MP3 storage on compact disc:

Keep your files in MP3 format

Advantages

Assuming 128 kbps, you can store 650 MB of MP3s, or about 10 hours of music on a single CD.

Disadvantages

You can only listen to the stored music through an MP3 player, which usually means on your computer. However, increasing numbers of dedicated MP3 hardware devices are appearing on the market, and some of them will let you play data CD-ROMs full of MP3s.

*Transform your MP3s into CD audio format**Advantages*

Can be played on any CD player (and most DVD players) in the known universe (and possibly in alternate universes as well, excepting those comprised of antimatter).

Disadvantages

Doing so means uncompressing your MP3s, so they take up 10 times more space. You're back to the 74-minute storage length of regular audio CDs. And doing that often means more manual labor to burn each CD.

How you approach this dilemma depends completely on your needs, and which end of the convenience spectrum you want to land on. If you do most of your MP3 listening at the helm of your machine and just want to free up some disk space, you'll want to burn normal data CDs. If you want to create standard audio CDs you can play in the car or on your home stereo, you may want to go the distance and start burning audio CDs.

Hardware/Software Requirements

In order to create CDs of any type, you'll need a CD-Recording device of some kind. Inexpensive models, as of late 1999, could be had for as little as \$200, while professional models (and CD duplicators) could set you back more than \$1,000. Most people should find the base models more than adequate for their needs, however. If you want to do some detailed research on CD recorders, take a look at CD Media World (www.cdmediaworld.com).

You'll also need to decide whether to get a straight CD-R device or go for CD-RW. The latter type is capable of writing to a blank CD multiple times, just like you would with a hard drive. However, blank media for CD-RW is considerably more expensive than blank CD-Rs, and may not provide as much added convenience as you might imagine, primarily because CD-RW discs have a lower reflectance than standard CDs and thus can't be played back in many standard audio CD players. This makes them more suitable for data storage than for audio storage (although many late-model audio CD players are CD-RW-compatible). At a buck or two a pop, CD-Rs are cheap enough to make occasional mistakes with, even if you can't go back and correct them later on.

Burning Notes

The actual process of burning CD-ROMs is completely dependent on the burning software used. While almost all CD-R and CD-RW burners come with some kind of burning software, it's often possible to use software other than what comes with your unit, in case you don't find it up to snuff. Some users with exacting demands,

for instance, find that the Adaptec software that comes with many CD-R drives is not configurable enough, and has trouble with some scratched CDs. Many “pro” users turn to Goldenhawk’s CD-recording software instead (www.goldenhawk.com). In any case, the process is usually pretty intuitive, and most CD burner applications come with complete documentation and wizards to guide you through, should you need assistance. In general, the process consists of nothing more than telling the software whether you intend to burn an audio or data CD, then dragging your tracks or files into a window.



If you want the audio CDs you burn to be playable both in computers and in stereos, you must remember to “close” the disc when the burn is complete. You should find an option for this somewhere in your burning software.

When shopping for a CD burner, don’t get overly excited about speed without asking around online about burning quality first. While not universally true, some people find that the faster the burner, the higher the chance of bad burns, or of pops and glitches in the final CDs. With a high-quality drive such as a Plextor or Yamaha, however, fast burns can still be error-free (the author hasn’t toasted a CD yet with his 8x SCSI Plextor burner). The better drives from Panasonic and Pioneer also get good marks among users for bit-perfect DAE. Also, make sure you get a drive capable of “Disk at Once,” or DAO technology. This lets you burn disks without a two-second delay between tracks (not generally an issue for mixed collections, but possibly annoying for live shows and for creating segues).

When shopping for blank disks, don’t lunge immediately for the 100-pack at the super-mongo discount price. The chances of bad burns (which wastes both time and money) is much higher with cruddy CDs than with good ones. Unfortunately, there are no blanket rules for determining which of the dozens of available brands is going to work best for you. Buy your CDs in small quantities until you find a brand with near-perfect, reliable results. Again, check around online for recommendations, and see www.cdmediaworld.com to learn all about the various dye colors and manufacturing plants.



If your CD-ROM drive is working but you’re experiencing pops, glitches, or drop-outs regardless of which burning software you use, look for an option in the software that will let you decrease the burn speed or increase the buffer size. You may get more accurate transfers at slower speeds.

Making space

Depending on the CD burning software you're using and the kind of disk you wish to burn, you may need to set aside a big chunk of disk space for the process. If you're burning exact duplicates of existing CDs, this won't be an issue as you'll be copying data directly from one CD to another. If you're burning custom CDs consisting of various tracks from multiple sources, however, you'll need to make sure you've got enough temporary storage space to hold everything that's going to go on the CD. Since the capacity of a CD is 650 MB,* one of the best solutions is to create a 650 MB disk partition. Fill it to capacity with WAV or AIFF files (or MP3s, if you're burning a data CD) and you'll know exactly how much space you've got to work with. You can then just drag the entire folder into the burn window of your CD creation software, or create a "disk image" of the entire partition and burn the image to CD (images or "Disk At Once" options are especially useful if you have albums with songs that flow into each other smoothly, or anytime you want to eliminate gaps between the tracks). If you plan on burning images, however, note that you'll need twice the space, or 1,280 MB, since the image file will be exactly the same size as the total amount of data you intend to burn.

Since creating a new partition isn't a convenient option for all users, you may just have to keep an eye on the size of the folder you're storing source material in. You can learn this information in Windows Explorer by right-clicking a folder name and choosing Properties from the context menu, in the MacOS Finder or BeOS Tracker by selecting a folder and tapping Opt+I to Get Info, and in Unix/Linux by typing `du <pathname>`.

In addition, most graphical burning software will provide some kind of interface widget or status area reporting the total amount of space occupied and remaining on the pending CD by examining the contents of the burn list. Thanks to this feature, you can drag in files from anywhere on your system without needing to set aside a specific folder or partition. Your burning software shouldn't let you attempt to burn a CD with a burn list larger than the capacity of a CD.

Format Conversion

If you only want to burn data CDs, you can skip this section and just use the documentation that came with your CD-R unit. If you're burning audio CDs, read up. While some CD burning software, such as MP3 Maker for Windows, Toast for MacOS, and CDBurner for BeOS will let you drag MP3 files directly into the burner application and have everything decoded to PCM audio automatically, most CD

* Audio CDs have actually pushed this limit up to 750 MB.

Custom CDs vs. Portable MP3 Players

Some people wonder whether it's worthwhile buying a portable MP3 player after they start burning their own audio CDs. Why not just buy a portable CD player and be done with it? After all, most first- and second-generation portable MP3 players won't give you 72 minutes of music encoded at a reasonable bitrate without additional storage cards, and audio CDs don't suffer quality loss due to compression. A portable CD player can play your existing CD collection along with your custom CDs. CDs are universally compatible, and can be played anywhere you find a CD player. Offloading MP3s to CDs will save you tons of disk space. Finally, investing in CD-Rs gives you more flexibility for your money because you can use the same drive as a normal CD-ROM in your computer, and can burn CDs containing any type of computer data, not just audio.

If you're wondering by now why the recording industry isn't more freaked out about recordable CDs than it is about the Rio and similar players, remember that there's a critical difference here: You have to make a physical copy of the disc for every person you want to share your music with, and that's a serious barrier to piracy in comparison to the popular-but-usually-illegal "Upload once, distribute to the entire web population" model. In addition, portable MP3 players are much smaller than portable CD players, and have no moving parts. Portable MP3 players never skip, no matter how hard you exercise. And you don't have to burn a new CD just to get a new playlist!

burner applications will only accept uncompressed audio as input. That means you'll need to find a way to convert your MP3 files to WAV or AIFF before you can make an audio CD. There are many ways to do this, depending on your operating system and the tools you use. Here are a few of the most common methods; see your application's documentation if you use other tools.

Windows: WinAmp

Right-click on WinAmp's interface and select Options → Preferences from the context menu (or just tap Ctrl+P). Navigate to Plugins → Output and select Nullsoft DiskWriter. When you click the Close button, you'll be asked to specify an output directory. Now just play your MP3s normally. You won't hear anything, as all output will be dumped to WAV files rather than sent to the sound card. To decode batches of files, just set up a playlist and let it go to town. Don't forget to return to the Preferences panel and reset the output to WaveOut or DirectSound when you're done!

MacOS

Most MacOS users don't do format conversion directly. Instead, they use CD burning applications like Adaptec's Toast that convert MP3 to raw audio as files are dragged into the burning application itself. Creating audio CDs from MP3s with Toast is no different from creating audio CDs from any other source; the process is elegant and totally transparent.

If you do want to convert from MP3 to AIFF, use the Converter built into SoundJam MP. Open the Converter window, drag in your tracks, and change the MP3 option in the "Convert using..." picklist to AIFF. This may be especially useful if you have an older version of Toast that doesn't convert from MP3 to PCM audio automatically.

Linux: *amp*

Many command-line MP3 players for Linux include the ability to decode MP3 files to PCM audio; you'll need to check your player's documentation for this ability. Here's an example using the encoder `mpg123`—a simple shell script that creates a WAV file for each MP3 file in a given directory; you may need to change the initial command and flags on the second line to work with a different tool:

```
for i in *.mp3 ; do
    mpg123 -w $i ${i%.mp3}.wav
```

To burn CDs, most Linux users use `cdrecord` (www.fokus.gmd.de/research/cc/gclone/employees/joerg.schilling/private/cdrecord.html/), a command-line burner that works with the majority of CD recorders out there, and which has a ton of optional features. `cdrecord`'s documentation is somewhat dense; be sure to look through some of the FAQs and HOW-TOs available on the `cdrecord` site. For a nice graphical wrapper to `cdrecord`, download a copy of `xcdroast` from your favorite Linux software library.

BeOS: *SoundPlay*

BeOS comes with a drag-and-drop CD burner as a standard component of the operating system—just drag MP3 files from the Tracker into the CDBurner window and click burn—an audio CD will be generated automatically. Third-party burning software is available as well, as is a BeOS port of `cdrecord`. If you just want to decode MP3 files to WAV without burning them first, drop the files you want to decode onto SoundPlay's interface, then click on the title of the playing file. From the Special menu, select "Decode to file." Tell SoundPlay whether you want to decode to RAW or WAV, single or all files in the current playlist, and specify the destination folder. Click OK, sit back, and you'll end up with a folder full of RAW or WAV files.

CDs vs. MiniDiscs

Some users prefer to store their music on the Sony MiniDisc format rather than recordable audio CDs, for a variety of reasons. First, and most obviously, MiniDiscs are much smaller than CDs, while storing the same amount of music (ten MiniDiscs require less physical volume than four conventional CDs). This also means MiniDiscs are great for joggers and other people who use a portable player heavily. By default, all recording on the MiniDisc is done with a proprietary, non-MP3 Sony format called ATRAC, which compresses audio at a ratio of 5:1.

It's possible to record to MiniDiscs 100% digitally, just by connecting a TOSLINK digital signal converter to your sound card's digital output (actually, this is a technique which should work with any kind of consumer-oriented digital audio device that comes with TOSLINK or S/PDIF output jacks). This means you can just play MP3 tracks through your favorite decoder and dump them straight to MiniDisc. Finally, MiniDiscs are re-writeable, like a CD-RW device. If you're thinking of buying a RAM-based portable MP3 player, you might want to consider a MiniDisc instead—you'll get a lot more playing time and flexibility.

On the other hand, CD-R blanks are quite a bit cheaper than MiniDisc blanks, so you can afford to just throw out the ones you don't like and not worry about it. Perhaps more troubling, you lose compatibility with the zillions of standard CD audio players out there (this problem is somewhat less substantial in Europe and Japan, where MiniDiscs are more popular than they are in the U.S.). Finally, recording to MiniDisc is quite slow—a 74-minute disc will take 74 minutes to record, which is far longer than the speeds you'll get with 2x, 4x, 8x, and even faster CD-R drives.

In this chapter:

- *Playing MP3 Through Your Home Stereo*
- *Portable Players*
- *Hand-Held Computers and Other Devices*
- *Home Stereo MP3 Players*
- *Car Players*
- *Kit Players*

6

Hardware, Portables, Home Stereos, and Kits

Until recently, MP3 audio was a software-only proposition for most people. MP3 files were created on the computer and listened to on the computer. But that was before the appearance of the now-legendary Diamond Rio, which won the landmark court battle against the RIAA and, in so doing, legitimized a tidal wave of hardware-based MP3 playback devices. At this writing, there are at least a dozen portable MP3 players, and a whole genre of MP3-oriented home stereo devices is threatening to enter the consumer channel. Dozens of sites are available on the Web designed to help you build an MP3 player from scratch, and many alternatives for playing MP3 in your car are available.

This chapter covers the aspects of MP3 playback that escape the limited confines of the personal computer. Since many of the devices covered here involve connecting computers, home stereos, and custom devices, we'll start with an overview of analog and digital connection issues, before examining the field of MP3 portables and the quickly changing question of how best to store files for use with portables (including the advent of Lilliputian hard drives and removable memory cards). In a similar vein, you'll see how you can use a hand-held computer to play back MP3 files, assuming you've got a suitable PDA. We'll survey the landscape of MP3 equipment designed for use in home and car stereos, then move on to the tricky stuff: Do-it-yourself schemes for building your own MP3 hardware from pre-fab kits and plans, or, for the truly hardcore among you, from scratch.

Note that in most cases, the hardware mentioned here will not help you *create* MP3 files outside your computer—only play them back. The fact that commercially available hardware-based players are classified exclusively as storage devices—and thus do not contribute to piracy—is the essence of the landmark Rio suit and has completely shaped the MP3 hardware landscape.

Playing MP3 Through Your Home Stereo

There are basically two ways to listen to MP3 audio through your home stereo. Both methods have their advantages and disadvantages.

- Connect your computer's sound card to your stereo, thereby keeping MP3 decoding functions and playback control inside the computer.
- Purchase or build dedicated MP3 player hardware to occupy a permanent place on your stereo component rack.

Either way, you'll be hooking up your hardware either via analog ports, or through digital outputs in the playback equipment. The difference between these two types of connections is in the conversion process. As described in Chapter 3, *Getting and Playing MP3 Files*, digital bits need to be converted into analog signal at some point in the chain. The question is whether this job is done by a (usually) low-quality, built-in DAC (digital-to-analog converter) or by a higher-quality DAC in a digitally equipped amplifier, DAT, or outboard DAC.

Analog Connections

Since most people have neither amplifiers with digital inputs nor outboard DACs, connections are usually a simple matter of obtaining the right analog cables and connecting the right jacks.

Connecting your computer to your stereo

To make the direct connection, you'll need a special adapter and a length of RCA cable (also known as a "patch cord"). The adapter should have a 1/8" male jack on one side to plug into your sound card's line out jack and two female RCA jacks on the other, as shown in Figure 6-1.* Connect your RCA cable into the adapter's female jacks, and run them into the Auxiliary input on your home stereo's amp or pre-amp (actually, any input on your stereo besides Phono will suffice, but Auxiliary is often the only input available). Turn your amp or pre-amp's input selector to Aux, and you're all set.

The necessary patch cord and adapter can be obtained inexpensively at most electronics parts stores, such as Radio Shack. You can also find prebuilt cables designed for this purpose at www.musicmatch.com/get_gear/.

* To relieve a little stress on your cables, you can get a 1/8" stereo plug attached to a soft cable, rather than the hard adapter pictured here.

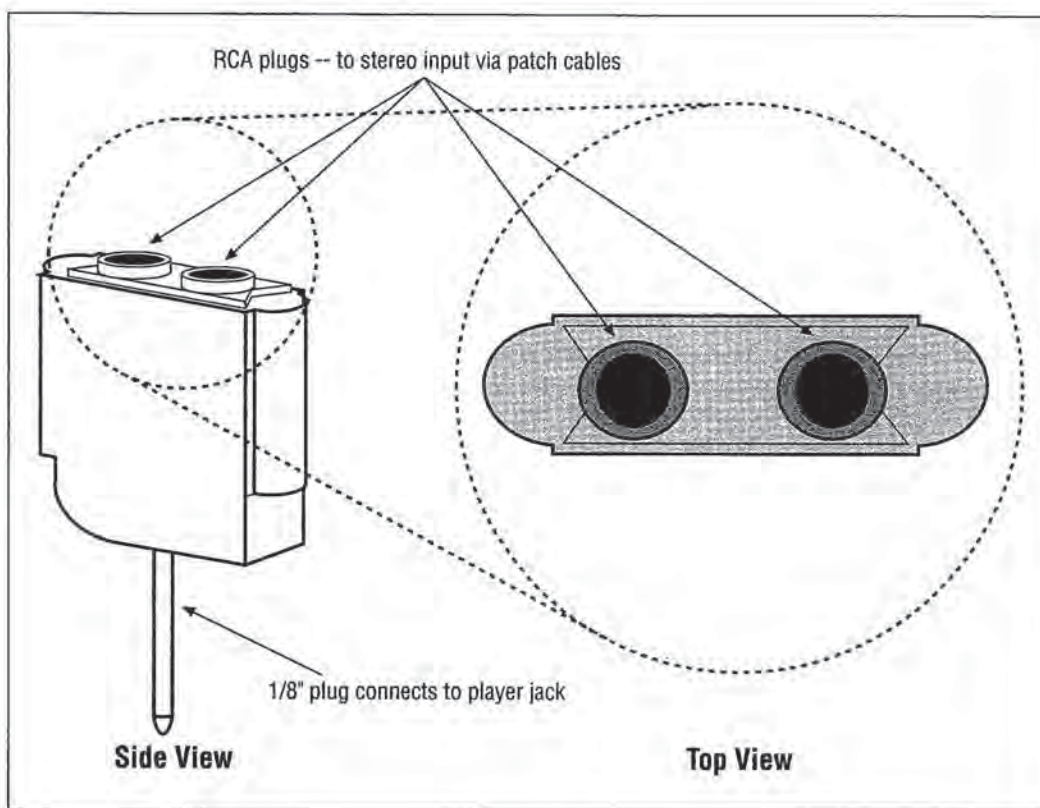


Figure 6-1. Obtain an adapter similar to this one to connect an external MP3 player or computer sound card to your home stereo



Do not purchase more cable than you reasonably need. In the ideal environment, your computer would be very near your stereo, so you can use 3' or 6' patch cords. Of course, this is seldom feasible in real living situations, where the computer is in a completely different room. But keep in mind that longer cables will pick up noise and increase impedance, which will cause high frequencies to roll off.

Connecting external devices

Virtually any external MP3 playback device (some of which are covered later in this chapter) can be used through your home stereo, in addition to functioning as a stand-alone unit. While connection options vary from one device to the next, the principle is generally the same. Home stereo MP3 components will have RCA plugs on the back which can be jacked into the Aux or any other free input on your amplifier (with the exception of Phono). Some car stereo MP3 units, such as the empeg, have a pair of RCA jacks as well, to make it easy to slide them out of the car and use them as a home stereo component. Future high-end MP3 players may include digital outputs as well, though we weren't aware of any at this writing.

Portable players such as the Creative NOMAD typically do not have RCA outputs, but rather a 1/8" headphone output jack. To play a portable device with these outputs through your stereo, you'll again need to obtain an appropriate adapter and cables. Some portable units come bundled with these. If not, you should be able to find one at your local Radio Shack or other electronics/stereo store.

Digital Connections

Digital input jacks have become increasingly common on amplifiers sold in the late nineties, and are the norm on amps built for use with home theaters. The advantage, again, is the ability to transfer the job of digital-to-analog conversion from the sound card itself to a device specialized for the task, such as a digital receiver, home theater system, or outboard DAC. The low cost and limited space available on a typical computer sound card means that built-in DACs are functional, but not optimal. Audiophiles often spend from \$300 to \$10,000 for a dedicated "outboard" DAC in order to handle all incoming digital sound optimally.

In order to use a digital connection, you'll need support for it on either end of the chain—in addition to support on the amplification side, you must also own a sound card or dedicated MP3 playback device with digital outputs. The Creative Labs SoundBlaster Live! is an exceptionally popular card with S/PDIF jacks, though the digital output comes on a "daughter card," not on the main sound card body. If you purchase the SoundBlaster Live! Value Edition, you won't get this daughter card.



The SoundBlaster Live! is used as an example of a popular, relatively high-quality card here, not necessarily as a recommendation. Audiophiles will note that the Live! resamples signal up to 48kHz, even in the case of digital output, and thus introduces some quantization noise. This is not the case when using Turtle Beach or Guillemot ISIS soundcards, for example. The ISIS is even sold with an additional external DAC.

Connection types

Digital connections come in four flavors, though you're likely to see only one or two of them on the vast majority of consumer hardware. Remember—you must have the same connector type on both your sound card and on your amplifier or DAC. Adapters between some of these formats can be found at specialty retailers, but it will take some hunting.



While some people are able to hear the difference between quality cable and consumer cable (analog or digital), others are not. In the grand scheme of things, the differences between these connection types are small enough to be considered negligible. However, if you have very good stereo equipment and listen carefully, the differences *are* there. Like fine wine, it's often the subtle differences that take the most training and experience to detect, but that mean the most in the long run.

If you're purchasing cable for digital connections rather than analog ("digital coax"), be sure to get cable rated at 75 Ohms for S/PDIF connections, or 110 Ohms for AES/EBU connections. Standard analog interconnects will do the job, but won't deliver optimal results. You can also use 75 Ohm video cable with RCA plugs for the job, and probably save some money in the process.

S/PDIF

Short for Sony/Phillips Digital InterFace, this connector looks like a standard RCA jack,* but one jack handles both left and right channels (if you see a pair of them on your equipment, they're for input and output, not left and right). Better home theater systems, outboard DACs, and digital-output-capable sound cards work with S/PDIF, while consumer-grade equipment is more likely to use Toslink (see the following entry). While you can use any standard RCA-type patch cable with S/PDIF connectors, most audiophiles purchase dedicated digital cables for the job. This cable type is typically referred to as *digital coax* in stereo stores.

Toslink

The most commonly available type of fiber optic connection is made over the comparatively inexpensive Toslink cable. Toslink jacks consist of a square-ish shield surrounding a small-diameter metal housing for the optical cable itself. Toslink does an adequate job of transferring digital signal, but does not tend to win accolades from audiophiles due to the fact that the fiber inside is plastic, not glass, and is therefore prone to internal reflections. Because the tip of the fiber must be kept perfectly clean, always replace the plastic shields that came with your Toslink cables when not in use.

AES/EBU

Found almost exclusively on high-end and dedicated equipment, this digital transfer spec moves through balanced cable—like a microphone—and usually terminates in XLR (3-pin canon-style) connectors. You'll most likely see this only on specialized equipment where quality considerations are paramount and long runs are necessary, as is the case in recording studios.

* Higher-quality equipment will use BNC connectors rather than RCA plugs for S/PDIF.

AT&T Glass

Expensive and uncommon, AT&T Glass is a fiber optic connection type which does not suffer the quality stigma of Toslink because the glass fiber is not subject to the internal reflection problems that affect plastic fiber. Only the most expensive, specialized equipment sports AT&T Glass connectors.

As with any cable type, make sure your digital connections are firm and secure, and keep cable lengths short whenever possible (though length is less important with optical connectors than with traditional cables).

Portable Players

MP3 took its first big step outside the realm of the computer when it entered the portable player market. In fact, the market is heating up so quickly that research firm DataQuest expects to see 155 million portable MP3 units sold to consumers in the year 2000. The devices are quickly becoming as ubiquitous as the portable cassette or CD player.

All of these units are based on the same basic concept as a portable CD or cassette player, but enjoy the advantage of having no moving parts, and therefore not being susceptible to skips caused by bumping or jogging. On the other hand, they have the disadvantage that they must be loaded up with music prior to use, which takes more time and effort than just throwing a few CDs into your backpack.



There are many more portable devices on the market than are listed here, with more coming all the time. As MP3 becomes less of a geek toy and more of a killer app for the general consumer, expect portable players to become as popular as—or more popular than—portable CD and cassette players are now. What you see here is a brief survey of the field at this writing, focusing on features and flexibility. Think of this as a guide to the kind of functionality you should be looking for, not a complete menu of portable choices. Take a look at the Hardware section of MP3.com at www.mp3.com/hardware/ for more options.

You'll also want to give some thought to the type of computer connections offered by the device. Various portable brands and products connect to your PC over parallel, serial, or USB connections. I don't know about you, but I haven't had a machine with a spare parallel or serial port for some years now, and despite what anyone says about parallel pass-through arrangements, these have often caused me grief. The parallel port just wasn't designed for sharing (it *is* faster than serial, however), and increasing numbers of low-cost consumer computers ship with only one serial port, rather than two or four, as they did once upon a time. The real

solution is USB, which is designed for sharing, and is much faster than either parallel or serial. Unfortunately, USB-compatible portables were somewhat rare at this writing (this situation should rectify itself in the near future).



To quickly compare the kinds of features you should be looking for in a portable player, see Table 6-1.

Speaking of connections, make sure you get a unit with its own desktop “cradle,” as shown in Figure 6-2. Without a cradle, you’ll have to manually connect your portable to your PC every time you want to transfer tracks. This becomes less important with USB connections, since you can just leave a USB cable dangling from a USB hub. Since a cradle stays permanently connected, all you have to do is drop the unit into its bay, fire up the software, drag a few tracks into the player’s desktop software, and be on your way.

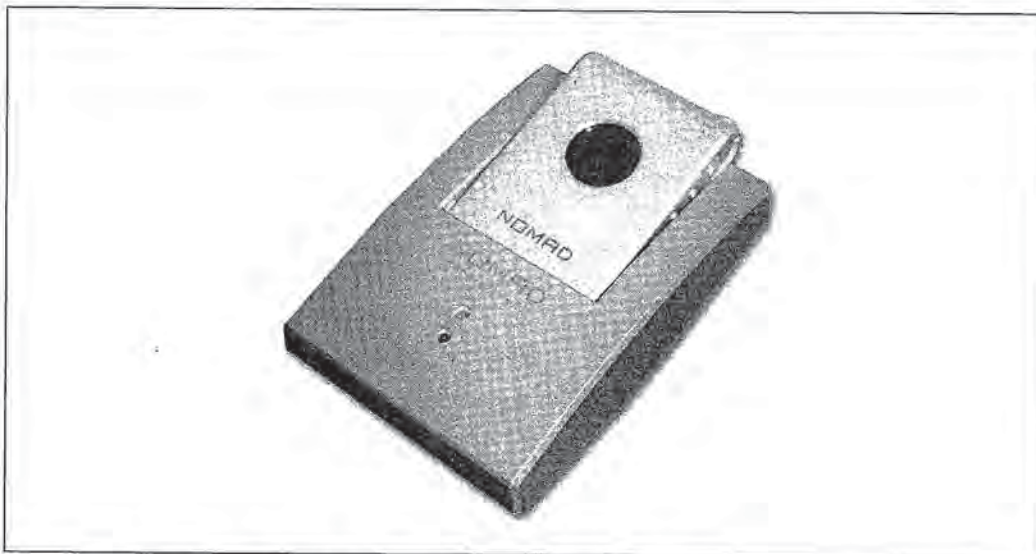


Figure 6-2. The Creative Labs' Nomad portable player in its cradle

Memory and Storage Issues

Most first and second generation portable units suffer from limited storage capacities, which means you get to listen to the same music (typically an hour or less) all day long. If you want to cram more music into a portable player, you have to encode at lower bitrates, which means decreased quality. The problem is being mitigated by the advent of next-generation, hot-swappable memory modules, tiny hard drives, and portable CD players that play MP3 data CDs. Just prior to press time, a new player called Personal Jukebox (www.pjbox.com) had announced an

MP3 player packing a 4.86 GB hard drive capable of storing around 100 CDs worth of music, in a form factor only slightly larger than a traditional MP3 portable.



By Christmas 1999, most portable MP3 hardware was slated to start being SDMI-compatible (see Chapter 7, *The Not-So-Fine-Print: Legal Bits and Pieces*, for more on SDMI). If you want no part of this scheme, you may do well to scan the online auction sites for an older portable.

Many of these units are also capable of operating as simple storage devices for data other than MP3 and other audio files. If the provided software allows (and in many cases it does), you can drag Word docs, images, spreadsheets, or whatever you like into the unit. If you purchase an additional cradle so you have one attached to both your work and home machines, you can take advantage of your fast Internet connection at work to download the latest huge browser upgrade, take it home stored in the portable's memory bank, then transfer it back to your home machine's hard drive for installation. Whoever said floppies were obsolete was right.

Memory to go

When shopping for a portable, *do not* underestimate the importance of getting a unit with as much memory as you can afford. Limited amounts of memory are the single biggest stumbling block to portables being truly viable. Vendors frequently overestimate the amount of playback time you'll get with 32, 64, or 96 MB of RAM in their marketing materials. If you don't mind swishy, warbly sound, sure you can fill up your portable with 96 kbps files and get a lot more playing time. But is it really worth it? If you care enough about your music to want to bring it wherever you go, then you care enough to have it sound decent.



Be cautious when a vendor tells you you'll get an hour of "digital quality audio" from your unit. What does the term "digital quality" really mean? Pretty much anything the vendor wants it to mean. In reality, such statements are often based not on 128 kbps audio, the quality of which is already questionable, but on 96 kbps or even 64 kbps encodings. They might be talking about mono, rather than stereo encodings. Or they might be talking about 22kHz, rather than 44kHz audio. At those rates, the quality is noticeably worse than AM radio or cassette tape—noisy and swishy. Who wants to disrespect the music they love that way? On the other hand, many of these units are capable of taking voice memos at very low bitrates (low enough to discourage you from trying to record live music concerts), and you *can* store several hours of acceptable-quality class lectures in 64 MB without difficulty.

One of the most viable near-term solutions to the storage problem is the portable, hot-swappable memory module. These are beginning to appear from a number of vendors, but the two leaders in the field are SanDisk (www.sandisk.com) with their Multi Media Card (MMC) (Figure 6-3), SmartMedia, and CompactFlash cards, and Sony, who are currently marketing their "Memory Sticks." Sony has a leg up because they also manufacture so many popular consumer electronics devices, including photo printers, car navigation systems, digital telephones, digital cameras, and personal computers compatible with the Memory Stick technology. Toshiba, Matsushita, and SanDisk have announced a partnership to eventually develop postage-stamp sized memory cards capable of storing up to 256 MB, and capable of up to 10 MB/sec of throughput. Most portable units that accept memory cards at this writing were using MMC cards of one flavor or another, though Sony was preparing to jump into the market with a portable player of their own, which would of course use Sony Memory Sticks.

Regardless of vendor, the principle behind these modules is the same—purchase as many memory modules as you like, load them up beforehand, and take them with you. This solves the storage problem quite nicely... if you can afford to buy a bunch of them.



Figure 6-3. The SanDisk Multi Media Card, commonly referred to as an "MMC" is barely the size of a quarter, and can be used as everything from a floppy replacement to an MP3 swappable storage solution



An industry specification called SSFDC (for Solid State Floppy Disk Card) represents an attempt for flash memory cards to adopt common mechanisms and interfaces to ensure interoperability across devices. For example, an SSFDC memory card will work as easily in your portable MP3 player as it does in your digital camera or even your cell phone. If this is important to you, check the manufacturer's web site for mention of SSFDC compliance.

Lilliputian hard drives

Meanwhile, IBM has developed its MicroDrive, a hard drive barely larger than a Susan B. Anthony, weighing less than an AA battery, and capable of storing 340 MB of data (Figure 6-4). Of course, the flipside of small drives is a return to moving parts, and with them a return to skippability and mechanical failure, so this may be one for those who don't exercise rigorously. Interestingly, physics dictates that inertia won't affect tiny hard drives as much as it does larger drives; it should take quite a blow to dislodge the read arm of a microdrive. And, of course, first-generation technologies are always expensive until they reach mainstream adoption.

Shortly after IBM introduced the drive, Diamond Multimedia, Clarion, Samsung, Sanyo, and Casio announced that they already begun receiving shipments of the \$499 device for use in upcoming products. Diamond, in fact was the first to announce that their upcoming Rio II MP3 portable would include the MicroDrive. Meanwhile, i2Go (www.i2go.com) had just released a device called the eGo near press time, which was capable of working with two MicroDrives, for a total storage capacity of 680 MB. Things are about to get a whole lot better.

In my mind, that kind of storage capacity is the key to real usability for these units, and will obviate many of the hassles of carrying around additional flash memory cards and/or constantly updating the contents of MP3 portables. If you're an early adopter and just can't wait, get one of the available units now—they're loads of fun. But if this is a device you intend to use for a long time to come, hold off for better storage capacities; the scene is going to get a lot better than it was in early 2000.

Portable MP3 data CDs

Also new on the scene at this writing are portable devices that can read MP3 tracks from data CDs, which means you get a meaty ten hours of playback time. Unfortunately, it also means a potential return to the skipping problems that occur whenever mechanical devices and physical exercise come together. These skipping problems can, however, be mitigated by a good read-ahead caching scheme.

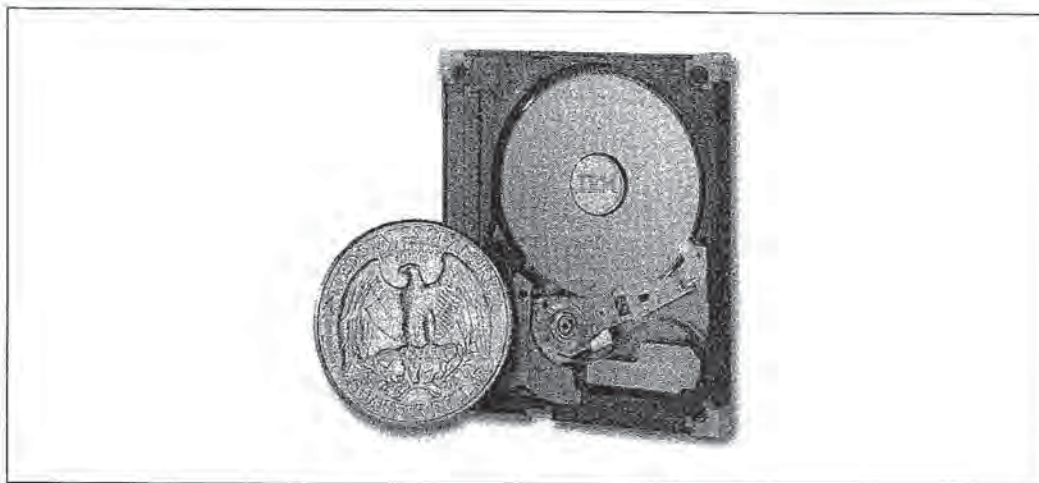


Figure 6-4. IBM's MicroDrive stores 340 MB of data and is small enough to be used in portable MP3 devices

If a unit can read a few minutes of music off the CD in one "gulp," it can then play that music from memory until the buffer expires, keeping the time spent reading data from disc to a minimum. See the section on the Pine player, later in this chapter.

Diamond Rio

www.rioport.com Diamond's Rio changed everything. With the advent of the Rio portable unit, MP3 files were freed from people's computers and suddenly became playable at the gym, on the subway, in the schoolyard, and while skydiving. The suit against Diamond by the RIAA (as covered in Chapter 7, *The Not-So-Fine-Print: Legal Bits and Pieces*) set a critical precedent and cleared the way for the future of similar devices and burgeoning competition.

Since the Rio was a first-generation device, it lacked many of the features now commonplace on portable units. However, subsequent editions of the Rio are much improved, with more memory, USB connections, a better-looking case, and a ~~built-in equalizer so you can optimize the output for any given genre~~, as shown in Figure 6-5. While Diamond has big plans for the future, they're shipping both their original Rio 300 and the newer Rio 500 simultaneously through the first part of 2000. The price difference of \$100 gets you a USB connection, more storage, a cooler case, a backlit display, and MacOS support.

Diamond has created an integrated software/web site solution called the RioPort, which is the name for both the companion software you get with the Rio and the site at www.rioport.com. While any MP3 user can head over to RioPort and download tracks from their growing library of signed (often "name brand" artists) for



Figure 6-5. The Diamond Rio PMP500, in its cool translucent case

free or for a small fee, Rio owners have the advantage of tight integration between the site, the RioPort software, and your Rio player. The software, which is a combination music library, ripper/encoder, and sync utility, will automatically detect MP3 files on your system, regardless whether they come from the RioPort site or another site. The software also includes a built-in web browser that connects directly to RioPort.com and other popular MP3 sites, so you can find the files you're looking for quickly and easily. Once downloaded, songs become part of your music library, and can be synced with a connected Rio (or, in fact, any portable player) or played directly through the computer.



The Rio 500 has a cool bookmarking feature that lets you mark specific points in your files. This is particularly useful for saved news broadcasts, lecture notes, etc. To mark a point in a file, just press the Bookmark button. To access your saved bookmarks, use the thumb-wheel on the unit to navigate through your bookmark collection.

Rio use on other platforms

While older versions of the Rio do not provide software or the proper connection type for MacOS users,* BeOS and Linux users are in luck, thanks to the presence of an Open Source version of the Rio filesystem protocol made available by The Snowblind Alliance. Linux users can download source for the Rio utility from www.world.co.uk/sba/rio.htm. After compiling and installing, usage is straightforward. You'll need to be running as root, since it accesses hardware directly. You'll also need to know the address of your parallel port (if your PC doesn't use the default) and specify it with the `-p` flag. Common memory addresses for parallel ports are 0x378, 0x3BC and 0x278. Then use:

```
rio -d
```

to display a directory of the files currently loaded on the unit. To upload a new file, use:

```
rio -g ~/data/mp3/somesong.mp3
```

to delete a song, use the `-z` flag instead. The utility is capable of many additional functions; see the included documentation for details.

BeOS users have it even easier. Rather than using the Rio utility directly, Be provides a version of the Rio filesystem as a dynamically loadable add-on, as shown in Figure 6-6. See Be's documentation on setting up the Rio.† Once mounted, the Rio appears in the Tracker just like any other mounted file system, and you can simply drag MP3 files into the Rio folder from elsewhere in the Tracker. In fact, you can even FTP files directly from the Internet into the Rio.

If you're a Mac user, you'll find that SoundJam (covered in Chapter 3) includes native support for later versions of the Rio over USB. You'll need to have USB 1.2 support installed—this comes with MacOS 8.6 and later. If you're using an earlier version of MacOS or a PCI USB adapter card, you'll need to download USB 1.2 or later from Apple's support site. You'll find an application in the SoundJam folder called RioPort SoundJam MP. Launch this and you'll be able to drag files into the player from the Finder, create folders within the Rio, delete files, reformat the Rio's memory module, and sort your on-board playlists exactly as you would a standard SoundJam playlist.

Rio: The next generation

While an exact release date was not available at press time, we were able to get a few projections on features to be included in a version of Rio to be released sometime in the first quarter of 2000. Diamond is looking closely at giving users a

* The Rio gained native MacOS connectivity with the advent of the Rio 500.

† BeOS support for the Rio was in "experimental" mode at this writing.

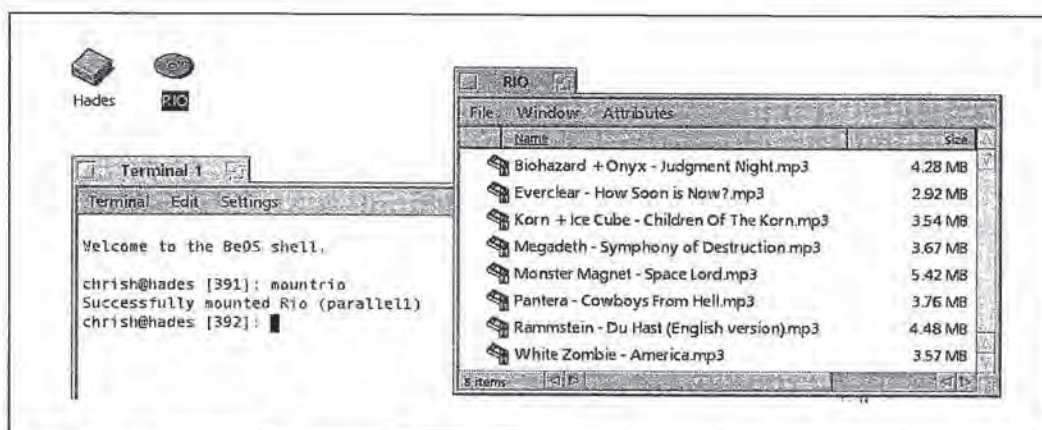


Figure 6-6. In BeOS, users mount the Rio as a normal filesystem

choice in the type of removable storage to be used. Users should be able to buy a base unit capable of playback, and expansion modules similar to those available for the Handspring Visor (covered in this chapter), so users can snap on an MMC module, Memory Stick module, or even an IBM Microdrive module. As the number of mini-storage options increases, this kind of expandability will become increasingly important, and users won't be locked into storage technologies if they should become obsolete.

The new Rio is scheduled to include support for Windows Media files, and Diamond is working with InterTrust to work out the details of a solid security model.

The company is working with outside design firms to find an optimum form factor; Diamond wants users to be able to operate the unit comfortably without even looking at it, which is important for use when driving or biking, for example. And of course Diamond wants the units to get even smaller, so users can stuff them into smaller pockets. They're also looking at ways to expand the size and quality of the display. Ultimately, it should be possible for users to view song lyrics stored in ID3v2 tags, or even to potentially display album cover art. Better playlist support is also on the table.

Meanwhile, Diamond is working on plans to launch a home stereo MP3 component, possibly similar to the Rio in ways, but in a home stereo component form factor and integrated into the home network. Diamond is beginning to see the writing on the wall here: many consumers have seen home networks as a geek toy without any real advantage for normal user. But everybody loves music, and once people see what they can do with a networked MP3 home, they'll be compelled to get those miniLANs up and running. No further details on the device were available at press time.

Creative Labs' NOMAD

www.nomadworld.com Creative Labs, progenitors of the widely popular SoundBlaster and SoundBlaster Live! sound cards, has their own answer to the ubiquitous Rio: the NOMAD (Figure 6-7). The NOMAD I is available in both 32 MB and 64 MB models, both of which can take additional 32 MB flash ROM cards. The NOMAD II, on the other hand, has no built-in memory and relies instead on add-in memory modules, for greater expandability. The NOMAD II can accept any type or size of SmartMedia card.



An important point to keep in mind is that any unit that utilizes memory modules rather than built-in storage circumvents issues surrounding the ability to get files back out of the device (see Chapter 7).

Both versions of the NOMAD feature a built-in FM tuner and, more interestingly, built-in voice memo capabilities. If you have a sudden brainstorm, are fresh out of paper, or forgot your PalmPilot, you can just switch to Memo mode, hit the Record button, and speak directly into the device. When you're done speaking, your memo is encoded to MP3 (at an extremely low bitrate, judging by the quality of the results) and stored in the device's memory. When you later fire up the NOMAD I Manager software, you can save your voice memos to hard disk for posterity (note that, like all portable MP3 players, you *cannot* transfer music files out of the NOMAD I and back to disk; this would allow piracy and put the vendor afoul of laws outlined in Chapter 7).

Because the NOMAD II is fully programmable, the unit is designed to be "future-proof." By this, the company means that users can run Creative-provided software to update the built-in flash ROM, which will theoretically enable it to handle additional file formats or copyright protection mechanisms. This means you won't be left out in the cold if the industry decides to move in another direction. Like a computer, you'll just upgrade the software, rather than being left with an obsolete device.



You *must* set your parallel port to ECP to establish a connection with the NOMAD I. There's a good chance your BIOS is already set up this way. If you're not sure, enter your computer's BIOS screen at boot time and look for the option. Again, this problem could be easily circumvented by providing additional transfer ports.



Figure 6-7. The author in the great urban outdoors with the Creative Labs' NOMAD I

RCA Lyra

www.lyrazone.com While the Lyra doesn't break any real new ground, it does offer a few features that weren't available on other players at this writing. Perhaps most noteworthy, the unit is software-upgradeable, meaning you won't be locked out if one of the other excellent file formats discussed in Chapter 9, *Competing Codecs and Other File Formats*, comes into greater popularity. As shipped, the unit supports RealAudio G2 in addition to MP3, which could be especially beneficial for those wanting to listen to Internet radio shows on the train or bike ride to work. G2 files are uploaded to the unit through the same drag-and-drop interface provided for uploading MP3 files.

The Lyra also sports a six-line LCD readout—displaying track and artist names, folder name, track number (as stored in the unit—not the track number on the original CD), and the bitrate and elapsed time. The display is backlit, so it can be easily read in dimly-lit situations. It also supports a certain amount of on-board file management, so you can actually arrange your music in directory structures similar to those on your hard drive*—a capability I'd like to see in more portable players. To manage folders on the Lyra, just click the New Folder button and give the new directory a name. Files can be dragged into folders from within the Lyra management software, which is a modified version of RealJukebox (see Chapter 5, *Ripping and Encoding: Creating MP3 Files*).

* Up to 99 directories are supported, each of which can have up to 99 subdirectories.

Living with the NOMAD

I lived with a NOMAD I for several weeks, taking it virtually everywhere, from long bike rides over mountainous terrain to shopping trips in the car. The unit performed flawlessly, though I did have some qualms with certain aspects of both the hardware and software design. I found the unit's buttons well-located and easy to access, even when the its leather slipcover was in place. However, the buttons are not easy to distinguish from one another without looking at the unit, which made it very difficult to operate when biking.

With a pair of my own headphones, I found the audio quality of well-encoded MP3 files excellent.

I also really appreciated the magnesium case; many players out there come in plastic cases, lending them an air of, as Frank Zappa would say, "cheepnis." This unit, even though it weighs only a few ounces, felt rock solid in the hand, like a quality instrument. The NOMAD II is scheduled to move to a plastic case, which allow for more color options. This is unfortunate from an aesthetic perspective but not surprising—the company pointed to manufacturing issues with the magnesium cases.

So what are my complaints? For one thing, the NOMAD I is for some reason beholden to Microsoft's antiquated 8.3 filenames restrictions. This doesn't affect usage of the player, but if you drag a file called "Bob Dylan—One More Weekend with You.mp3" into the NOMAD Manager, it will appear in the panel as "BOBDYL~1.MP3." If you have multiple files all starting with the same artist name, it becomes very difficult to tell which file is which. Yuck. Note, however, that this does not affect the unit's own display window, which properly shows ID3 tags. Why this couldn't be done in the Manager interface as well is beyond me. This issue should be cleared up with the NOMAD II, though it was too early to tell at press time.

Second, I found my patience worn a little thin by waiting around for transfers to happen. In my perfect world, the NOMAD cradle would have USB, FireWire, and Ethernet interfaces so I could plug in any of the high-speed connectors available to my system. The version of the NOMAD I tested had none of these (a problem the NOMAD shares with most of the first- and second-generation portable units). Oddly enough, even deleting files takes much more time than one would expect. If you want to delete a lot of files, your quickest bet is to click the "Format" button at the top of the NOMAD Manager window, rather than deleting files singly or in groups. Fortunately, the NOMAD II is scheduled to include USB support, so this should be a non-issue. Interestingly, the convenience of having a cradle becomes less important with units like the NOMAD II, since all you really need is a free USB cable to plug in—no more messing with parallel ports on the back of the machine.

I-Jam's I-Jam

www.ijamworld.com If small is your bag, the I-Jam is one of the smallest portable MP3 players of the bunch, and comes in a rainbow of case colors. Supporting both USB and parallel, MacOS and Windows, the I-Jam boasts a heckuva lot more power than other players, pumping out 60mW, rather than 5, as most players do. The I-Jam stores music only on SanDisk memory cards (Figure 6-3), and includes no built-in storage of its own. The unit also comes equipped with two output jacks—one headphone and one stereo output—which is a nice touch. Also included is a neck strap for those who don't wear belts (though a belt clip is provided as well).

Unlike other portables, the I-Jam comes with no dedicated upload or file management software. Instead, I-Jam provides a .DLL which allows the I-Jam JamStation (a sort of dock for your memory cards) to appear in Windows Explorer as a normal drive. Once mounted, all you have to do is drag files into or out of the drive letter, just as you would with any other drive. You'll know your unit is full when you get an "Out of disk space" error message. Not only does this capability free you from having to use dedicated software, but it also gives you the ability to organize your files in directories and subdirectories, and to store any type of data you like on the memory cards, rather than being limited to audio files—essentially treating the unit like an overgrown floppy drive.

Table 6-1 compares a variety of MP3 players. Features and specs for portable players vary from one to the next. While it may be impossible to find a single player that does everything, think carefully about your needs before purchasing. This table represents a brief snapshot of the market as of October 1999—options may be radically different by the time you read this.

Table 6-1. Portable MP3 Players Compared

Storage type	Diamond Rio 300	Diamond Rio 500	Creative Labs' NOMAD I	Creative Labs' NOMAD II	RCA Lyra RDD204	PONTIS MPlayer3 ^a	I-Jam's I-Jam
Built-in storage capacity	SmartMedia	SmartMedia	Flash memory	SmartMedia	32 MB or 64 MB flash cards only	MultiMedia cards	San Disk MultiMedia Card (16 MB, larger ones expected in the future)
Weight (in ounces, w/o batteries)	Built-in 32 MB plus optional 32 MB add-in cards	Built-in 64 MB plus optional 32 MB add-in cards	32 MB plus optional 32 MB cards	None plus optional 64 MB cards	None plus optional memory modules	None plus two slots	In future models
Dimensions (in inches)	2.4	2.75	2.5	Unknown	4.0	3.5	2.5
Can record voice memos	3.5 × 2.5 × 0.625	3.59 × 2.46 × 0.74	2.3 × 3.35 × 0.67	Unknown	4.5 × 2.5 × 0.875	4.36 × 2.75 × 0.8	3.375 × 2.0 × 0.875
FM radio tuner	No	No	Yes	Yes	No	No	No
Backlit display	No	No	Yes	Yes	No	Planned	Yes
Output connector types	Headphone	Headphone	Headphone plus stereo connecting cable	Headphone plus stereo connecting cable	Headphone	Headphone	Headphone and Stereo
Upload connection types	Parallel	USB	Parallel	USB	Parallel	Serial, parallel, or USB	Parallel or USB
Output signal strength?	5mW	5mW	5mW	Unknown	15–20mW	5mW	50mW

Table 6-1. Portable MP3 Players Compared (continued)

	Diamond Rio 300	Diamond Rio 500	Creative Labs' NOMAD I	Creative Labs' NOMAD II	RCA Lyra RDD204	PONTIS MPlayer3 ^a	I-Jam's I-Jam
Case type	Opaque black plastic (teal or camoflage in some editions)	Opaque metallic gray or translucent teal or purple	Magnesium	Plastic	Metal	Plastic	Plastic (clear or opaque)
Cost (as of December, 1999, in U.S. dollars)	169.95	269.95	249.99	Under 400.00	249.99 for 64 MB with car kit, 199.99 for 32 MB	190.00	219.95
Software upgradeable	No	Yes; audible codec	No	Yes	Yes	Planned	Yes
EQ controls	4 presets	4 presets plus bass and treble	Yes, environments	Yes, environments	5 modes: Flat, bass, rock, pop, jazz	Bass and treble	Bass and treble
On-board file management	No. Displays song numbers only, not filenames	Yes; Filenames displayed, play lists possible	No	Unknown	Yes	Planned	Complete (shows up in Explorer)
Supports other audio file formats	MP3 only	MP2, Audible, com content	Windows Media (WMA)	Any formats are possible with software upgrade	RealAudio G2	Planned	Will support all audio formats
Can store normal data in addition to audio files	No	No	No	Yes (Smart-Media)	No	Planned	All data types

Table 6-1. Portable MP3 Players Compared (continued)

	Diamond Rio 300	Diamond Rio 500	Creative Labs' NOMAD I	Creative Labs' NOMAD II	RCA Lyra RDD204	PONTIS MPlayer3 ^a	I-Jam's I-Jam
Copy protection mechanism	None	MetaTrust infrastructure present, not implemented w/o upgrade	No	Planning SDMI	Will have SDMI in future	Planning SDMI	Will be SDMI
MacOS support	No	Yes	No	No	No	Yes	Yes
Linux/BeOS support	3rd party	3rd party	No	No	No	Yes via open SDK	No
Comes with cradle	No	No	Optional	Memory dock	No	Optional	JamStation

^a Not covered in this chapter; see www.mplayer3.com.

Hand-Held Computers and Other Devices

Necessity is the mother, and MP3 is a necessity for many people. But not everybody can afford to buy a dedicated portable, since they already spent this year's geektoy budget on a PDA or other hand-held device. Fortunately, the popularity of MP3 is driving the extension of some existing devices to support the format. The list here is short, but expect MP3 support to find its way into nearly anything that has a CPU and can conceivably play audio in the near future. In fact, Ericsson had just announced a cell phone with MP3 playback capability via an add-in cartridge at press time.

Utopiasoft's Hum

www.utopiasoft.com This one isn't actually a portable MP3 player, but, a skinnable software system for WinCE-based handheld computers that lets you play MP3 files directly through your PDA (Figure 6-8). Of course, most PDAs ship with a very limited amount of memory, and much of what's there is already occupied by the operating system, applications, and your data. That means Hum probably won't be useful unless you're using add-in CompactFlash memory cards, which are currently available in capacities up to 96 MB.

The Hum software utilizes a technology called AdaptivePlay. If your device has a slow CPU, or is busy handling tasks for other applications, or does not have high-quality media capabilities, audio processing is bumped down to 22kHz automatically. When more computing resources become available to the device, quality jumps back up to the normal 44kHz automatically.

Handspring Visor/InnoGear MiniJam

www.handspring.com / www.innogear.com As this book was going to press, lovers of the original PalmPilot and Palm OS were reveling in the news of the release of the new Handspring Visor, created by several of the original Palm team members and executives. While the Visor runs PalmOS (making it backwards-compatible with the thousands of Palm applications out there), it added the dimension of expandability, thanks to a new class of "Springboard Expansion Modules," one of which is InnoGear's MiniJam MP3 Player, a tiny hardware-based player that snaps into the back of the Visor. The MiniJam can use either power supplied by the Visor's internal batteries or be hooked up to an external power converter or to a cigarette lighter adapter. The unit ships with your choice of 0 or 32 MB of internal memory, and uses standard MMC (Multi Media Cards) for additional expansion. The company also plans to sell car cradles for the unit.

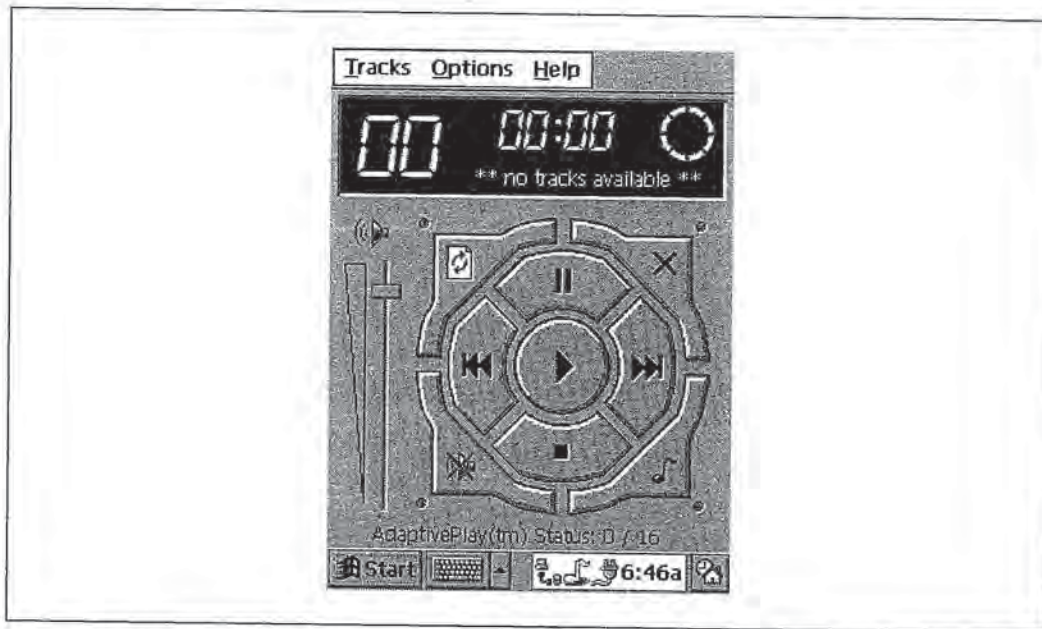


Figure 6-8. Utopiasoft's Hum lets you play MP3 files on your Windows CE-based hand-held computer



In practice, the MiniJam has two MMC slots. One is on the back of the unit, and is inaccessible when the module is snapped into place; the other is on the front, so you can swap memory cards without removing the expansion module. By early 2000, the industry projects that MMC cards may be available in capacities up to 128 MB, which means you may be able to stuff 256 MB of music into your Visor. And because the Visor shares memory space with its expansion modules, you can store data from your PDA on the same MMC cards that hold your music.

Controls and display on the unit are fairly standard: Stop, Play, Rewind, Fast Forward buttons appear on the top of the unit. Display screens are provided for playlist, ID3 tag info, and equalization controls (four EQ presets are available, though users can modify and save additional EQ settings). Users can even store voice memos on their Visors via the MiniJam.

Internally, the MiniJam uses DSPs from Micronis, which some believe offer sound quality superior to that of the DSPs used in most units. Early testers were allegedly claiming that the MiniJam sounded better, and had better bass response than the Diamond Rio, though I was not able to confirm this. The MiniJam outputs a standard 5mW through its headphone jack.

Of PDAs and MP3s

As you can see here, there are both Win CE- and PalmOS-based Personal Digital Assistants (PDAs), or hand-held computers capable of playing MP3 files. If you've been thinking about getting a PDA anyway, should you consider going with an MP3-capable PDA instead? There are pluses and minuses here. Obviously, storage is the biggest limitation you'll be facing. But while PDAs traditionally have shipped with a paltry 2–8 MB of storage, the genre is beefing up as it matures, and hand-held computers are increasingly becoming more full-featured, and coming with larger complements of built-in RAM and hot-swappable storage options. The trend shows no sign of abating.

One advantage to using a single device for both purposes (besides the money you'll save) is the fact that you'll be able to use a single desktop cradle, minimizing the number of ports you'll need to take up on the back of your computer (not an issue for USB-connected devices, thankfully), and saving a little space on your desk. And of course you'll have one less device to cart around or clip to your belt.

On the other hand, a PDA isn't going to have the full complement of physical buttons on its case (unless you get an add-in hardware unit such as the InnoGear expansion module for the Handspring Visor, as shown in Figure 6-9). You'll probably need to operate everything via software running on the PDA itself, just like you do on your desktop computer. However, some MP3 software for PDAs may be capable of mapping simple stop/start/next controls to the normal operational buttons on the PDA itself—you'll need to do a little research beforehand to see whether this will be possible. Perhaps more importantly, your PDA may not have the horsepower to run an operating system and its applications while simultaneously decoding MP3 in the background. Custom DSPs in dedicated MP3 players take care of this job with no risk of skipping or drop-outs. The audio chipset in a PDA also isn't as likely to have the same level of quality you'll find in a dedicated player.

If MP3 is your life, there's probably no question you'll want to go with a dedicated MP3 unit. If you just want to listen to the occasional MP3 file in the background, give this option some serious consideration. The possibilities of having a well-integrated unit are only going to get better as time goes on.

The companion software shipped with the MiniJam is Real Networks' RealJukebox Gold (covered in Chapter 3), which is capable of searching your system for attached portable units. Behind the scenes, RealJukebox ships with Windows .DLLs containing connectivity code for various players. The MiniJam is one of them, so the unit shows up as a choice in RealJukebox's portables menu. Of course, the MiniJam does not require a separate cradle: it uses the Visor's.

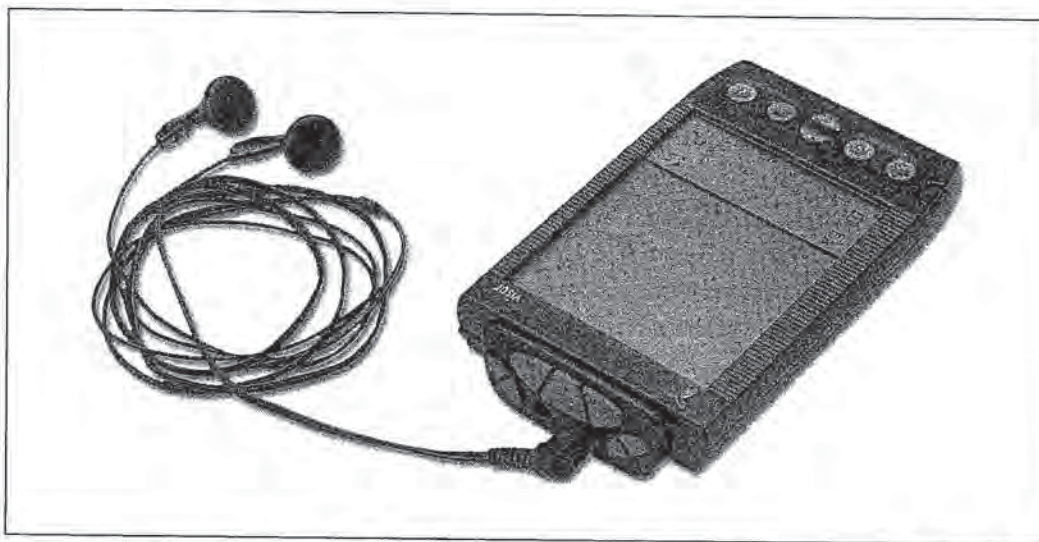


Figure 6-9. The InnoGear MiniJam is an expansion module for the HandSpring Visor PDA

Mac support should be available sometime in 2000, while Linux and BeOS users will be happy to learn that InnoGear is planning to distribute a developer's package (SDK), which will include all the source code needed to deliver MiniJam support for those platforms. The SDK should become available sometime in the first quarter of 2000.



One of the more unusual MP3 players scheduled to make the scene by the time you read this is Sony's VAIO Music Clip Personal Network Player, which quite literally comes in the form of a fountain pen (though it doesn't write). It's truly amazing to behold how many goodies Sony has packed into this little device: An LCD readout, a headphone jack, the standard array of control buttons, and—get this—a USB port. The device stores up to five hours of music and runs on a couple of AA batteries. Advantages: Slips easily into your shirt pocket so you don't have to deal with belt clips or long cords. Disadvantages: Non-expandable storage, limited battery life. Think of the Music Clip not as the most far-out product you've seen, but rather as a harbinger of devices to come: Unlimited by traditional constraints of shape or size, and capable of being streamlined into your lifestyle with near transparency.

Home Stereo MP3 Players

At this point, it practically goes without saying that MP3 playback hardware is bound to become a fixture of the consumer home stereo setup. Because most stereo component MP3 players will be linked to a computer somewhere in the home,

these devices fit perfectly into the “convergence” scenario you’ve been hearing about for so long; the boundaries between computers, networks, and home entertainment systems eroding one piece at a time.

At bottom, these dedicated players aren’t all that much different from portable devices, though they’re packaged in standard home-stereo-size component cases and typically offer more flexibility and playback options than you get with portable players. Not to mention storage space up the yin-yang.

There are essentially four types of home stereo MP3 players:

- Devices similar in concept to a portable: Load it up from your PC via parallel, serial, Ethernet, or USB and treat it much as you would a home stereo CD player.
- Full-on miniature computers in a home stereo form factor and modified with LCD or LED readouts and remote control units.
- Modified CD players designed to handle MP3 data CDs as well as standard audio CDs.
- Noncomputer-based home stereo components dedicated to handling MP3 and other audio file formats.

... and of course we’ll be seeing myriad variations and hybrids of these concepts.

Choosing the Best External Player

While the market for external MP3 players is still young, this is clearly going to become a huge field before long, and competition will undoubtedly create a bewildering array of choices for consumers looking to jump in. So what should you look for in an external player? Here are some criteria to keep in mind as you shop, which you will of course have to balance against your budget.

Storage

How much can it hold? If you have to load it up with new music every few days, you’ll probably end up regarding the device as a hassle. If you’ve got a huge collection, you’ll want to cram as much of it into the device as possible. If you’re already in the habit of burning your own CDs, look for a unit that can play directly from MP3 data CDs.

Good connectivity

If you’re looking for a unit that interfaces with your PC for file transfer, don’t settle for archaic parallel or serial connections.* They’re slow and they hog

* Actually, transfer speeds over parallel can be pretty good if ECP is enabled in the BIOS and the transfer software support is. But that doesn’t get around the inconvenience of using the parallel port rather than USB.

ports that could be put to better use with other devices. If you have a home network, look for a machine running a real operating system, with a real Ethernet port. This way you'll be able to transfer files to it from any room in the house via FTP or Windows Networking, or, if it's a Unix/Linux- or BeOS-based machine, telnet in and manage the machine remotely. Failing that option, don't settle for anything but USB or FireWire. At this writing, none of the commercially available boxes allowed for Ethernet connectivity, though several of them promised it for the future. However, potential legal consequences make it difficult to allow this kind of transfer commercially; you may have to build your own machine if this is what you're looking for. The advent of SDMI should make MP3 networking commercially possible without stepping on the toes of the legal suits.

Display

Look for a machine with a clear, multi-line LCD readout viewable from a distance. Make sure the unit is capable of reading ID3 tags, so you're not left grappling with meaningless track numbers. Check to see whether the display output can be redirected to a TV (most cannot, though it could be a nice feature, depending on your needs).

Software

The unit will probably come with some kind of companion software that can be used for uploading tracks, creating playlists, and possibly even encoding albums. Find out from existing users of the product what they think of the software. Does it impose unnecessary or arbitrary limitations? Can you work with tracks of any bitrate? Good companion software may be an important aspect of your enjoyment of the device.

Playlists

Some of the early devices on the market were limited to playlists of a certain size, or could only manage a single playlist. Music is largely about mood, and you're going to want to be able to switch atmospheres with the press of a button. Check to see if the device or its companion software will let you create query-based playlists. If you're going to have gobs of storage on the machine, you better be able to organize it however you like, either on the machine itself or from playlists created on your PC and transferred to the device.

Audio output

Does the device use a standard sound card for output to your stereo, a custom DSP, or a proprietary chipset? If it uses a sound card, find out what kind. Overall quality may be heavily affected by audio chipset quality. If the device is a sealed unit, you may void your warranty by trying to upgrade it yourself. Ask around online to find out other people's impressions of unit before plunking down your cash. If you have a digital amplifier or DAC, look for a unit with optional digital outputs.

Ambient noise

If the machine is based on standard computer hardware, it may also contain a fan or fans, just like your PC. Do you really want to hear the hum of a fan in the background when you're trying to listen to music? This may be unavoidable with some devices, but ask around to find out just how much noise is being generated by the fan before committing your money. If you have a choice in the matter (i.e., you can open up the case and replace parts without voiding the warranty, you might consider replacing the power supply and CPU fans with some of the ultra-quiet models available from a specialized retailer such as www.pcpowercooling.com. Also see the Silent Systems section of www.molex.com. If you're really hardcore about it, consider coating the inside of the case with acoustical foam, though this will almost certainly void your warranty and could introduce air circulation issues if your existing fans aren't moving enough air.

CD types

If the device has a built-in CD player, will it handle both audio and MP3 data CDs? And with the steady rise of recordable DVD, the possibility arises of storing gobs of MP3s on a DVD disk. You might want to make sure your unit is capable of handling these as well. Of course, you may be able to stick a DVD drive in the machine yourself at a later date, but will the unit's software be upgradeable as well? Most units based on computer hardware are software upgradeable, but make sure before spending your money.

Hackability

This one depends on your personality type. If you're a real do-it-yourselfer, you may want to make sure the machine is running an operating system you understand well, so you can make modifications if necessary. And again, you may want to give the device a larger hard drive, or monkey around with the controls. Will doing so void your warranty? Probably. That's a decision only you can make. If you're more the standard consumer type, you'll probably want to go with a slick, sealed unit much like your existing stereo components.

ReQuest's AudioRequest

www.audiorequest.com One of the most interesting and complete home stereo MP3 solutions out there is ReQuest's AudioRequest—a complex and unusual beast capable of just about everything your PC is capable of doing, MP3-wise. So why not just stick a computer next to your stereo and be done with it? First of all, it wouldn't look too hot, and to make it really useful you'd either have to have a monitor attached or install an LCD readout and control buttons. You'd want to rig up a remote control unit, and figure out how to connect your PC to your television

(AudioRequest lets you manage playlists and play synchronized animated video images via TV hookup). And you'd have to deal with the ambient noise created by the CPU and power-supply fans. Ambitious geeks could pull all of this off, sure, but by the time you were done you probably would have invested a lot of time and money, while the AudioRequest is ready to plug and play, and looks slick to boot.

AudioRequest includes hard drive options large enough to store 150 or 300 hours of MP3 audio—that means you can play it 24 hours a day for 2 weeks before hearing the same song again. And because the device is virtually a complete computer,* it won't be difficult to upgrade its software to allow support for other file formats. Windows Media files were about to become playable on the AudioRequest at this writing.

But it gets even better: In addition to hard drive storage, the AudioRequest (Figure 6-10) also includes a CD-ROM drive capable of playing either audio CDs or MP3 data CDs. In fact, you can even slide an audio CD into the tray and encode it's tracks directly to MP3, directly within the unit.

Perhaps the biggest difference between AudioRequest and a standard PC is that ReQuest takes great pains to enforce compliance with existing copyright laws. Unlike the Rio, AudioRequest most definitely *is* a recording device as well as a playback unit, and thus has to disallow the transfer of unprotected digital audio files back to your PC. Details on that arrangement are covered in Chapter 7.

AudioRequest's ECP parallel connection can be used both for transferring files from your computer to the device or for transferring from the device to portable players. Some form of USB networking is scheduled to be included on the unit in a future release. Analog inputs are also provided, so you can easily encode from cassette or LP.

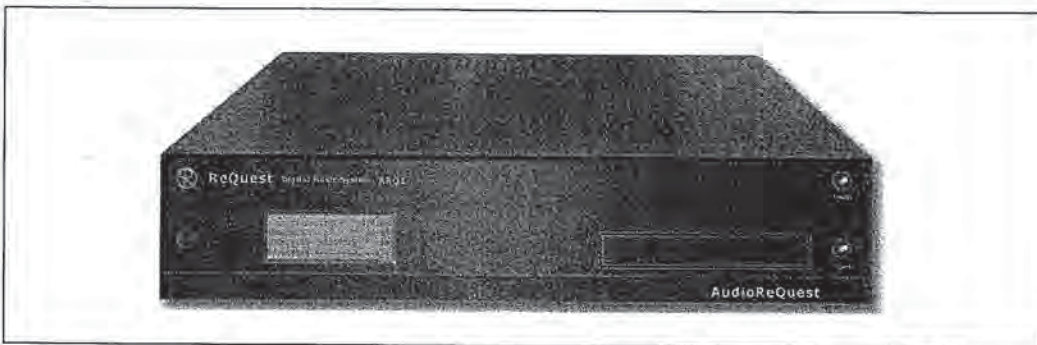


Figure 6-10. The AudioRequest prototype; the appearance of shipping units may differ

* At press time, ReQuest would not disclose what operating system the device runs on.

Lydstrom's SongBank

www.lydstrom.com Occupying a rather unique niche in the file-based home audio scene is Lydstrom's* SongBank (Figure 6-11). The SongBank is not specifically an MP3 unit, though it does handle MP3 files, alongside many other digital audio compression formats including Windows Media. Future versions of the SongBank will support the RealNetworks G2 and AAC formats as well—Lydstrom is format-agnostic. Internally, the unit uses Lucent's EPAC format, for an optimum quality-compression ratio. To understand what the SongBank is and does, you have to first stop thinking in terms of computer-based devices. The SongBank is not a computer, nor does it interface with your computer. It's a home stereo component with an eye toward the convergence of digital audio, the Internet, and even some non-audio home-control functions (you can power up your coffee pot in the other room with this thing).

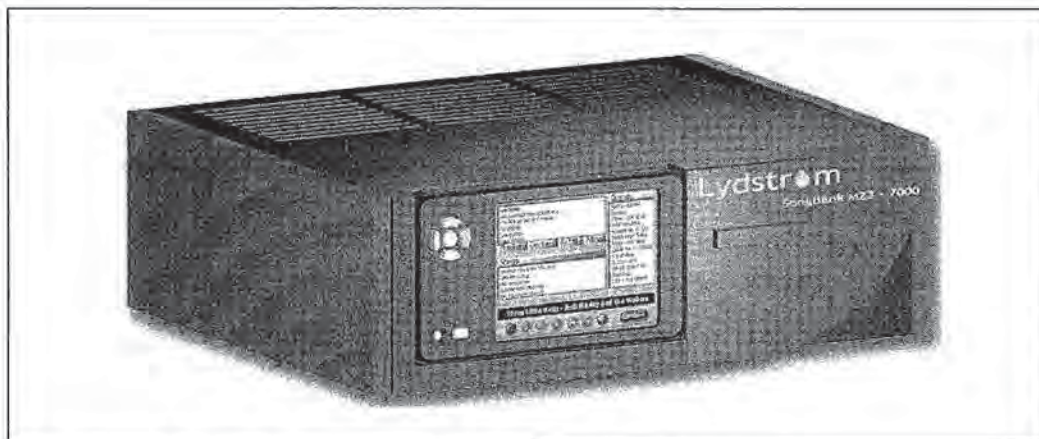


Figure 6-11. An artist's rendering of the futuristic Lydstrom SongBank

The unit itself features very sleek Euro styling and a large and detailed LCD read-out/touchscreen remote control unit (the remote control "docks" into the face of the SongBank and doubles as the onboard control interface). Rather than a single output to your home stereo, the SongBank offers three separate analog output ports and three digital ports (Toslink, S/PDIF, and AES/BEU), connected to three separate software playback modules. Each playback module can be directed toward a separate output, so you can play different music in different rooms of the house, all from the same unit. Want to play blues in the kitchen and jazz in the den? Just select a room and a genre from the remote control, which is RF-based and can work in a house of up to 30,000 square feet in size. While the base unit already stores 6,000 songs, additional "slave" (external storage) units can be connected via FireWire for nearly infinite archivability. Current slave units are standard

* Pronounced "Ludstrum," where "Lud" is like "hood." It's a Swedish name.

external hard drives, but Lydstørn will eventually support various types of removable cartridges, flash cards, PCMCIA units, etc. Other models available at this writing can store up to 14,000 songs without slave units.

How it works

So, how does this thing work again? You pop a standard audio CD into the player and it gets cross-referenced against an internally stored database (not the CDDb; Lydstørn says their database is more flexible than the CDDb) and all of its songs are encoded to EPAC format on the internal storage. The contents of that CD are then made available for playback on the unit via the remote control/faceplate, even after you remove the CD itself. And if you insert a data CD full of MP3 files, the unit handles everything transparently—same deal. Because the internal database is not on the Internet, the device doesn't require a network connection of any kind. Every month, you receive a CD in the mail that updates the SongBank's internal database and upgrades the onboard firmware to make improvements or to enable new features and capabilities.

However, the unit *is* Internet-capable, and Lydstørn plans to have a complete backend site online by the time you read this. Once that happens, you'll be able to optionally connect the unit through a modem or Ethernet connection (via DSL or cable modem) and have the database and firmware upgraded automatically. More importantly, you'll be able to browse Lydstørn's site for music and purchase it for download, all through the remote control unit, which Lydstørn claims is dirt-simple to use. Lydstørn won't be alone selling music online, though. They're partnering with other vendors so you'll simultaneously be able to browse and purchase from collections put online by other online music retailers. The company couldn't disclose their partners at press time, but they're thinking big here: imagining being able to connect directly to a music label, or to an online CD retailer like CD Now! or Amazon.com. Or to more established existing MP3 distribution sites. Meanwhile, the online database will be intelligent. It will know where you live, and will be able to tell you that concert tickets for the same musician are available. You'll be able to buy tickets and sample songs simultaneously, through multiple vendors, all through a single account you maintain with Lydstørn.



Lydstørn claims to have a patent pending on a technology that allows modem users to double their digital audio download speeds, though they were very tight-lipped about the specifics of this technique at the time of this writing. File this under "sounds too good to be true."

What happens when you want to get music out of the SongBank and into your computer? Well, you can't do that, for the usual legal reasons discussed elsewhere in this book, but Lydstørn is working with Rio and other vendors to support export to portable units. When you do that, music will be moved to the device, rather than copied, to prevent unauthorized proliferation of digital music. And Lydstørn is doing all of this with the computer completely removed from the loop.

New paradigms in distribution

The internal database used by the SongBank lets you create and manage playlists, of course, and also does some fancy tricks with the music itself. For example, the SongBank can keep track of the beat count and adjust the tempo of the next song (without changing the pitch), so guests dancing at your parties can keep right on jamming. You'll probably want to keep this option disabled most of the time—changing the tempo is pretty disrespectful to the artist, in my opinion, though DJs do it all the time for precisely the same reason.

Meanwhile, each SongBank owner gets a unique ID, and each unit has a serial number. This information is stored and transmitted through all online music purchases, so if your unit should be stolen or damaged, you can always retrieve the music you legally own without having to pay for it again.



Lydstørn is careful to point out that the SongBank is not a geek toy—they're not going after the technophile audience, but after the general consumer. They're selling a convergence device masked as a fancy CD player. Amazingly, the SongBank is not priced out of the ballpark. The base unit retails for \$399, and the higher-end models still come in at under a grand. The day of the sophisticated consumer convergence device is here.

MP3 CD-ROM Players

While only a few MP3 CD players are mentioned here, this field is growing fast. Be sure and search the Internet for more options.

NetDrives' Brujo

www.netdrives.com/brujo.asp This one is basically a modified CD player designed to handle MP3 data CDs as well as standard audio CDs. Since there's no on-board storage, there's no need to interface this unit with your computer. On the other hand, you must have access to a CD burner, so the total cost of ownership is higher than advertised by NetDrives. If you've already got a CD burner and have been busy socking away massive archives of MP3 data CDs, you'll find this unit a dream come true, with one CD being capable of storing 10–12 hours at 128 kbps.

The unit includes both standard RCA and headphone outputs, and comes with a 31-key remote control usable at up to 8 meters away. The unit can handle up to 220 MP3 files, but oddly enough, playlists are limited to 63 tracks.

MacPower Peripherals' Mozart's Music Box

www.macpower.com.tw/ Despite the name of the manufacturer, this unit has nothing to do with the Macintosh, nor does it require that you own one. The unit is a very simple combo player that handles both audio and MP3 data CDs and jacks into your home stereo via standard RCA plugs. Unfortunately, the unit does not read ID3 tags, so all you'll see on the LCD readout are track numbers. While the version available at this writing required users to hook up different cables to play either audio or MP3 CDs, an updated version without this limitation should be available by the time you read this.

Mozart's Music Box is equally at home in car or home installations. See pictures of the Mozart in car environments at www.mp3car.com.

Pine's D'Music

www.pineusa.com Shipping with no on-board storage of its own, D'Music is a portable CD player capable of playing both audio and MP3 data CDs. If you don't already own a CD burner, this unit probably isn't for you, unless you buy a lot of data CDs from MP3.com or other sources. If you've been storing up a multi-gigabyte collection and burning the overflow to data CDs all along, this unit may be just the ticket. The D'Music comes with a ten-second anti-shock buffer, a remote control (useful for hooking the unit into your home stereo), and a built-in graphic equalizer. Although designed for use as a portable, there's no reason the unit won't work well in home stereo configurations as well.

Car Players

What better place to get deep into your MP3 collection than on a long car trip, or to switch off the endless commercials on the radio and drink in a few new downloads to pass the time while you wait for the gridlock to clear? Because the basics of MP3 hardware are compact, they can be crammed into a car stereo form factor and treated like any other in-dash or pull-out car stereo unit. They can be hooked up to high-quality car amplifiers and speakers, or to those ridiculous monster-bass kettledrum speakers that make even the best music sound like it's been blown out with a gas station air pump.

After reading this section, learn more on MP3 audio and cars at www.mp3car.com.

Remote Control

While most of the non-portable hardware discussed in this chapter comes with its own remote control system, some users will be happy to discover that they can save a ton of money and “virtually integrate” their existing computers into their home stereos by adding a remote control to the PC itself. A company known as Irman (www.evation.com/irman/) makes a small black box about the size of a cigarette pack that connects to an available serial port. One face of the box includes an infrared receiver capable of interfacing with virtually every remote control ever made.

Once installed, users can use any of a number of MP3 playback applications running Irman plug-ins, or otherwise modified to accept incoming signals from the Irman box. Hook up your sound card to your home stereo, attach an Irman box, and run WinAmp or X11Amp from a distance. If your computer isn't situated within line-of-sight of common areas in the house, you can place the Irman box anywhere—even in another room. Serial cables up to 15 meters in length have been tested and allegedly work perfectly.

Then again, you can approach the situation the other way around, spend a bit more, and have a transceiver on your computer broadcast MP3 audio right through the air to a receiver on your home stereo. Essentially, you end up with the same situation you would get by hooking your sound card to your stereo, but without the need for long cables. X10's MP3Anywhere includes transceiver and receiver units that operate at 2.4GHz, and can transmit right through walls (unlike standard IR signals, which are blocked by obstacles). X10 also provides a remote control designed to operate with a wide variety of common MP3 playback software, giving you capabilities similar to those of Irman. See www.x10.com/mp3_x10/mp3_anywhere.htm for details.

Car MP3 on a Budget

Before you get too excited, be aware that the first few generations of any new technology are always going to be expensive, and this genre is no exception. At this writing, dedicated car MP3 players were running \$750–\$1000. Some intrepid souls were creating their own solutions with soldering irons and home-drawn schematics, but that's just not something your Average Joe is likely to do.

Fortunately, there's a cheaper, and possibly more flexible way to go about it. Get yourself a standard portable MP3 player as discussed earlier in this chapter and jack into your car's audio circuitry. There are a number of ways to go about this. Check the documentation or web site of your portable's manufacturer—there's a good chance they sell inexpensive accessories to make auto hookup easy.



To keep things organized inside the car, you might want to attach a bit of velcro tape to your dashboard and to the back of your portable unit. Alternatively, go to a cell phone store and look at the car phone caddies. Some portable units and remote control devices fit neatly into some cell phone caddies.

Cassette adapter

These odd little units are shaped exactly like a cassette tape, and fit into the cassette slot on your existing car stereo. Plug the headphone output of the portable into the jack on the front of the “cassette” and you’re in business. These units are a clever, inexpensive hack, but are a bit of a hassle—you’ve always got to keep the “cassette” handy and swap things around every time you want to play MP3s. But they do get the job done, and let you get more mileage out of your existing equipment. Note, however, that this technique is also the lowest-quality car MP3 solution you’re going to find.

Auxiliary in

Some car stereos come with an auxiliary input jack right on the face plate, which makes hookup of portable devices a snap. If you’re shopping for a normal car stereo anyway, be sure you get one with an Aux input, just so you’ve got the flexibility to work easily with a portable device sometime down the road.

Cigarette lighter adapters

If you want to save on batteries, you can get a cigarette-lighter adapter at nearly any electronics, hardware or department store. These usually come with a variety of connectors and/or a voltage switch. Portable players commonly run on 3V of DC, but check your portable’s documentation carefully before running current through it. You almost certainly *do not* want to run your car’s full 12V through your portable!



It is dangerous (and illegal in most regions) to wear headphones while driving.

Direct wiring

Unless you have Aux In jacks on the back of your receiver, or an equalizer or other external preamp (which also implies the presence of a separate amp), you’ll need to jack your portable directly into the car’s audio wiring via an “RF

modulator,” which accepts line-level input and outputs VHF or FM signal. However, this is not a job for the squeamish. If you’re not already well-versed in car stereo wiring principles, talk to your local car stereo installation shop.

If you’re not using a dedicated portable unit and want to hook up a laptop or dedicated computer in the trunk or under the seat, you’ll probably want to get an inverter to transform the car’s 12V DC into 120V AC. However, unless you purchase a very high quality inverter, you’ll find that these generate additional noise. In addition, using a laptop raises other issues—how will you control the unit? Will it be sufficiently shock-proof? While many users initially get excited about the prospect of using a laptop computer as a car stereo, only serious hackers or those willing to spend time and money putting together a complete system should contemplate this seriously. In the end, it’s going to be a lot easier and cheaper to just jack in a standard portable unit. If you do want to go this route, see the In-Trunk and Do-It-Yourself sections later in this chapter for additional tips, and start doing research at www.mp3car.com.

If you don’t know what you’re doing, let your fingers do the walking and head over to your local car stereo installation shop and let them take it on. In most cases, the job should be simple and inexpensive, but keep in mind that some cars make behind-dash access more difficult than it needs to be, and this fact may mean added labor charges. Still, you’ll end up with a fairly elegant solution for less than you would pay for a dedicated car MP3 player.

In-Dash and In-Trunk Players

If you know you’ll never be satisfied with the paltry 32 or 64 MB of RAM in your portable and are looking for a better solution, you’ll want to go with a unit specifically designed for in-car use. Like standard car stereos, these come in two flavors: Pull-out units you slide into the dash, and trunk-mounted units operated from the driver’s seat via remote control. Regardless, you’ll still need a way to get MP3 files into the unit. Chances are, you’ll have to take the unit into the house to load it up from your PC, unless your garage is very close to your office and you have a very long USB or Ethernet cable. The ideal solution, it seems, would be to use wireless networking technologies so you could FTP files into the unit from a computer in the house, right through the air. I haven’t seen anything like this on the market, but it’s only a matter of time.

empeg

www.empeg.com The amount of attention directed at the remarkable empeg unit in the year leading up to its release was nothing short of incredible. Audio buffs were excited. MP3 buffs were excited. And the Open Source software crowd over at Slashdot.org were positively jubilant, thanks to the fact that the empeg unit runs a superbly hacked version of Linux.



Developers and geeks may want to check out www.empeg.mars.org, a site dedicated to exploring the internals of the empeg unit at the source code level.

Make no mistake: This is not your mother's car stereo. Imagine the components of a laptop computer stuffed into a car stereo form factor, given a custom LED read-out instead of a monitor, and some neatly placed buttons instead of a keyboard or mouse. And rather than risking the "blue screen of death" right in the middle of your favorite track, make sure it's running one of the world's most stable operating systems.

Because the bootstrap code is stored in flash ROM, the empeg unit boots in a few seconds. The empeg includes a 220Mhz Digital/Intel StrongARM processor, 8 MB of memory, and a 6 GB hard drive (in the base model; a 10 GB model is also available, and even larger drives may be available in the empeg by the time you read this), and can be used as easily in the home as in the car, thanks to the presence of a pair of standard RCA plugs in addition to the car audio wiring. Digital to analog conversion is provided by custom DSPs built into the machine, which also handle bass, treble, and other audio output controls. And because the slimmed-down operating system takes up less than a megabyte of system memory, there's RAM left over to do read-ahead cacheing. The hard drive can grab around three minutes of music and store it in memory, so the read heads don't need to be in constant contact with the platter, saving both power and heat while further minimizing the possibility of pothole-induced playback errors.

The decoder software, based on XAudio, can handle MP2 and MP3 files, and since all decoding happens in software, it should be easy to upgrade the unit to handle other compression standards when/if they take off. Shock and vibration protection is provided by hard drives rated at 150G while running and 400G when not running.

Since 6 GB is a lot to transfer, the empeg mercifully comes with a USB port in addition to serial. You can load the hard drive to capacity in around an hour over USB. Oh, and it comes with a credit-card sized remote control, so your friends can operate the unit from the back seat (the driver uses buttons built into the front panel), as shown in Figure 6-12. The default faceplate is blue, though amber and green plates can be had for an additional surcharge.

The unit is loaded by connecting to your PC's serial (230,400bps max), or USB (12Mbit) ports, then firing up the bundled Emplode software (even though the empeg itself runs Linux, the desktop software is aimed at Windows 98/2000 users). Create new playlists by pulling down Edit → New Playlist, then drag tracks from

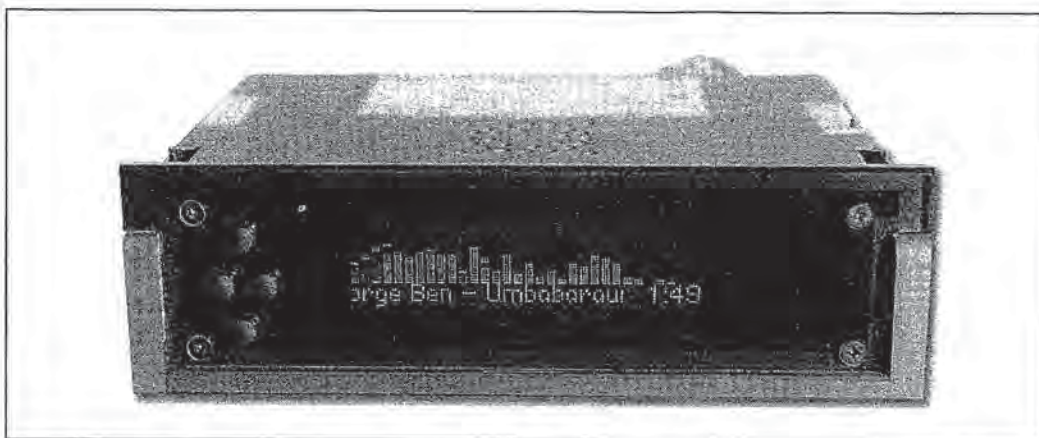


Figure 6-12. The empeg unit running through the author's home stereo

Explorer into that playlist. Space available on the unit is reported in the status bar at the lower right of the Emplode window (Figure 6-13).



As noted in the instructions, be sure to launch the Emplode software before hooking up the USB cable. Otherwise Windows will get confused looking for driver support for the attached unit.

Playlists are created via drag and drop, then synchronized with the contents of the unit. When you've got everything arranged the way you like it, click the Synchronize button on the toolbar and the transfer begins. By the time you shower, eat breakfast, and drink your coffee, the unit will be ready to grab on the way out the door.



Perhaps the only big minus for general consumers interested in going with an empeg unit is that since its case is tightly packed with computer parts, there's no place for an on-board amplifier. You *must* have an external amplifier in your car if you want to run the empeg.* In addition, demand was still exceeding supply at this writing, and potential buyers were queued up for months waiting for units to become available. It's likely that empeg will have ramped up production to a commercial level by the time you read this.

* Unless you're keeping your old cassette deck and or radio, in which case a cassette adapter or RF modulator should work. However, you would lose some quality in the conversion processes.

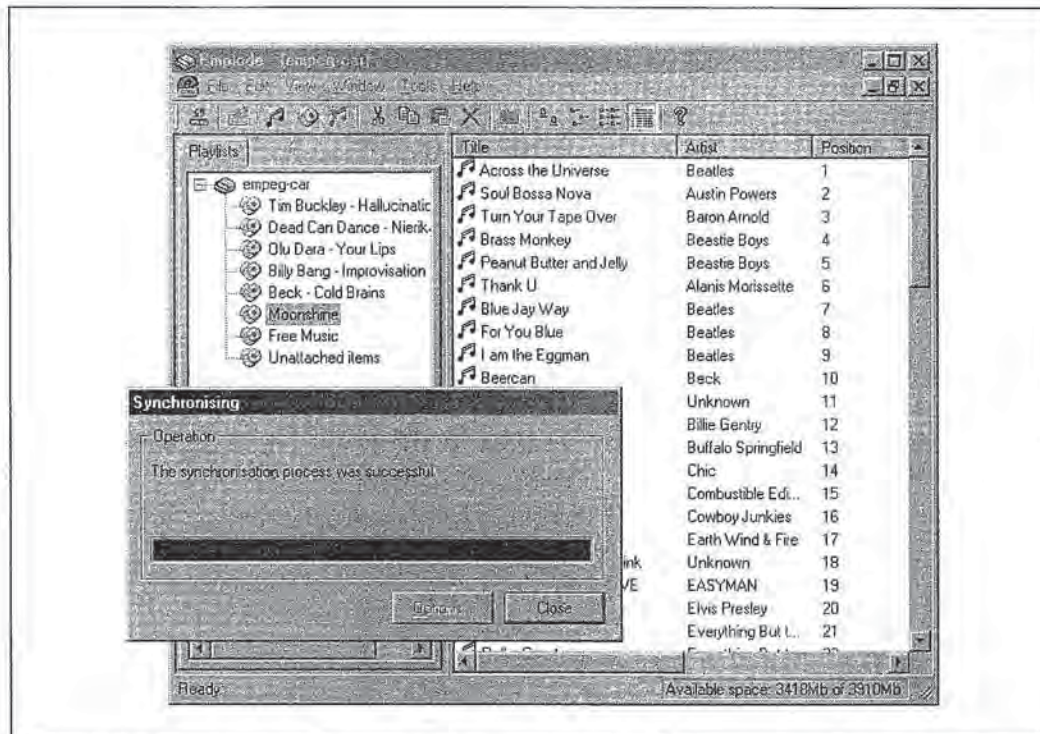


Figure 6-13. The empeg unit comes with its own transfer management software, called *Emplode*

Using the on-board controls

Navigating the empeg controls (see Figure 6-11) may take a bit of getting used to at first, especially as the buttons on the faceplate are unmarked. Pressing the bottom button on the faceplate will bring up a horizontal menu in the display, consisting of:

Playlists | Source | Sound | Shuffle | Information | Visuals | Power Off | About

Use the left and right buttons to navigate this main menu, then press the bottom button again to select that entry; you'll get a submenu relating to that choice. For example, pressing the bottom button while Shuffle is selected will let you turn shuffle mode on or off, while pressing the same button while Source is selected will let you toggle between empeg (the unit itself), auxiliary input, and FM radio. Selecting the Sound entry lets you control the volume, balance, fader (front to back balance), and the built-in equalizer. Be sure and check out the Visuals menu—there are a ton of built-in visualizers from which you can select, from simple ID3 tag display to rather sophisticated histograms, dancing pixels, and warping lines in time to the music. Pressing the top button will always let you escape from any given menu or submenu.



To get the full effect from the built-in visuals, navigate to the Information menu and turn Information off. All available pixels will be used for visualization display.

Upgrading the empeg

Both the firmware on the unit and the Emplode desktop software are software-upgradable, and users (especially early adopters) should check empeg's Upgrades page frequently at www2.empeg.com/upgrades/. This will not only get you bug fixes to the firmware and/or the synchronization software, but important new features as well. If there are upgrades available for both the Emplode software and the player itself, install the Emplode upgrade first. With Emplode open, pull down Tools → Upgrade empeg-car software, then click the Browse button and navigate to the location on your system where the upgrade binary was stored.



Always unplug the power cable from the unit before starting the upgrade. You'll be prompted to turn the power on later during the upgrade process. If you typically use the USB connection to synchronize with the unit, note that you *must* use a serial connection to apply the firmware upgrade (an appropriate serial cable is provided with the unit).

A progress indicator will inform you of the update's progress as the kernel and RAM disk are erased and reprogrammed. Finally, the MP3 data partition on the unit will be "pumped," which prepares the unit's database to work with its playback software.



It's worth remembering, whether you're talking about the empeg or any other unit that's essentially a re-fashioned computer, that just because you build it specially to play MP3 files doesn't mean that the rest of the powers of a general-purpose computer suddenly disappear. Give your machine some connectivity, and you've got web access from your MP3 player. Give it a satellite uplink, and you can have a GPS system in your dashboard, giving you directions to anywhere you need to go, or even playing music appropriate to the scenery outside your window. Need some extra storage space to cart big image or video files around? There's no reason your car stereo, with its massive hard drive, can't also function as a gargantuan floppy disk, letting you cart media files between home and work.

Living with the empeg

I had the pleasure of borrowing one of the first units to roll off the empeg assembly line, and lived with it for a month before I sadly had to return it. Because I do not have an external amplifier in my car, I used the unit exclusively as a home stereo component, in a month-long MP3 fest.

The empeg's built-in DAC is excellent, and I found the audio quality to be fully equivalent to CD audio when listening to 192 kbps MP3 files. Still, I would have liked to have seen a digital output on the unit so I'd have the option of using my outboard DAC for its extra warmth. I also found that the remote control, which was clearly designed for another device, was not as intuitive to use as I would have liked, though I quickly learned its controls. My biggest beef with the controls in my evaluation unit was the lack of a radial volume knob; it was just too much work to get to the volume controls from the faceplate buttons. Fortunately, empeg hints at a big change coming in the future: "... a volume knob done one better." They didn't offer more clues, but hopefully you'll be able to get a glimpse of their plans by the time you read this.

With a 6 GB hard drive, space was a non-issue. I only reloaded it a few times, and that was mostly to add new test files. 6 GB gave me enough storage to play music several hours a day for weeks without repeating tracks. The unit's display was clear enough to read from across the room, and I enjoyed switching between playlists with the press of a button, and toggling between the many built-in visualization modes. I found the built-in FM radio perfectly functional. In a perfect world, this unit would optionally come in a case designed to be integrated into home stereos, and would have an Ethernet interface so I could transfer files via FTP from any machine in the house. Fortunately, empeg hints on their site that they intend to offer a version that allows remote logins in the future. I'll know the future has arrived the first day I can telnet into my home stereo.

Xeenon MP Shuttle II

www.xeenon.com/xeenonweb/product/product.htm/ A rather different approach is taken by the Xeenon shuttle products. Rather than sticking a complete computer in the dash, Xeenon puts a unit in the trunk, with the exception of a removable hard drive which slides in and out of the dash for quick portability and security. The brains of the machine are out of sight and out of mind. All potential thieves will see is an empty hard drive bay in the car—hardly tempting.

Rather than controlling the unit from buttons on the faceplate, the Xeenon utilizes an external keypad, which connects through the car's chassis back to the brains in the trunk, which in turn control the hard drive in the dash. The arrangement

sounds a little inelegant, but has its advantages. The removable hard drives in particular mean that loading the unit up with new music will be faster than with any competing unit, and by locating the drive in the dash rather than in the trunk, everything stays convenient.

In essence, the unit is functionally similar to the trunk-mounted PCs some users build themselves, except the unit is ready to go—you don't have to scrounge for parts and hack things together. Think of this one as a hybrid between a total do-it-yourself tweak and a commercial implementation.



There are *many* more options for playing MP3 in the car, and don't be surprised if the major car stereo vendors start presenting their own models in the near future. Again, keep an eye on www.mp3car.com for the latest dope.

Kit Players

If none of the commercial offerings available by the time you read this quite fit the bill, if you think you can do it yourself cheaper, or if you're the type who has a garage of tools dangerous enough to take off a limb, you're a good candidate for an MP3 hardware kit. If you want something done right, do it yourself, right? Right.



Virtually all of the commercial manufacturers go to pains to remain compliant with SDMI and other security considerations. If you'd rather not be beholden to these laws and would rather be on your own recognizance in terms of legal issues, building your own player will give you the freedom to get tunes into and out of the unit without having to jump over security fences.

There are basically two approaches to building home- and car-based MP3 units:

- Purchase all the parts and plans as part of a bundled unit. Spend an evening with the instructions, get out your razor blade and gluestick, and start sticking Tab A in Slot B until you've got something resembling an MP3 player. This is the easiest approach, designed for those with little-to-no experience working with hardware and software at a low level.
- Purchase, download, or otherwise procure plans—without the parts. Then hunt down the parts yourself at flea markets and electronics warehouses. With any luck, you'll actually be able to find the same parts specified in the plans and everything will just work. In all likelihood, you'll end up doing some improvising and further research to make it all happen.

Of course, if you're really intrepid you can always go it alone, without a plan. However, this option should only be considered by self-styled inventor/programmers ready to get themselves out of any jam they end up creating for themselves. And be warned: Projects like this inevitably run into obstacles.

If you decide to just go for it, be prepared to spend hours in electronics newsgroups or at the library, figuring out how to fool a motherboard into thinking it's got a keyboard attached when all it's really got are a few buttons made out of bottle caps. Clear a space on the workbench, fire up the soldering iron, and go for it. Seriously though, this approach can be both the most challenging and the most satisfying.

If you decide to go the do-it-yourself route, you should definitely subscribe to the mp3stereo mailing list. Send a message to mp3stereo-subscribe@lists.gofast.net with the word "subscribe" as the subject.

Ready-to-Build Kits

While there are a number of ready-to-build MP3 hardware kits floating around out there, you'll want to track one down that best suits your needs and sense of aesthetics. Some kits consist mostly of additional parts needed to hack an existing computer into a standalone MP3 player, while others are based entirely on custom parts. Unfortunately, few of the kits I've seen available on the Internet are things of beauty—most place function above form. If looks matter a lot to you, you might want to either rethink and go with a commercial unit after all, or be prepared to depart from the kit and devise your own case and/or readout.

CarPlayer

One of the best all-around informational sites for MP3 kit builders at this writing was www.carplayer.com, which offers both prebuilt units for the car and the home, and offers both ready-to-build kits and plans without parts. In addition, the site offers a lot of data up front, so if you think you can get by on your own ingenuity, you may be able to simply studying the available diagrams, pictures, and explanations.

TrackZ

If you want to make sure the build process is painless and you're on a budget, check out the TrackZ player at www.mp3kit.com. This unit can be purchased pre-assembled or as a kit. Both are available in a variety of configurations, giving you control over the quality of the display, whether or not you want an external enclosure for it, whether you need cables, etc.



TrackZ uses a proprietary operating system, so don't expect to be able to fiddle much with the unit's basic functionality.

Do-It-Yourself

While the prefab kits make the build process relatively painless, many people opt to go the distance and create their players from scratch, using readily available, cheap parts and either devising the schematics themselves or relying on the shared experience of others who have gone down that road before. While this isn't a book on hardware or operating system hacking, here's a basic outline of the kinds of things you'll need to think about if you go the DIY route. After reading this section, hit the Net and start doing additional research. There are dozens of sites set up to show off the projects other people have built, where you'll find plans, diaries, solutions to common problems, and perhaps most importantly, some of the software you'll need to make it all come together. You'll find a list of good starting points in the "More information" section at the end of this chapter.

Hardware and form factor

The first thing you've got to decide is how your unit will be housed. Most people building their own players from scratch go with a complete computer, using standard parts from the "PC organ bank." This approach avoids the necessity of crafting your own parts to fit into a custom form factor or to play a particular role. Of course, the ease of use is offset by the fact that computer-based players are generally going to be larger than players built to custom specifications (though very small computer cases and motherboards *are* available, if you know where to look). Another approach is to get an old CD player from the swap meet and gut it, saving only the chassis and case. Alternatively, save only the chassis and build your own case from custom-molded sheet metal, plastic, or carefully bent stainless steel mesh.

If you're lucky enough to have an old laptop laying around, you can always use that to cut down on the size, keeping in mind that you won't be able to mount a display in the front. One of the unfortunate things about laptop solutions is that their built-in audio chipsets are generally inferior to dedicated sound cards, and can't be changed out on a whim. If audio quality is paramount, avoid the laptop route. A better solution is to shop around for the smallest case you can possibly find and start from there. Meanwhile, ask around online and find out who's making the quietest power supplies and fans available. You don't want all that humming interfering with your enjoyment of the music. See the notes in the "Choosing the Best External Player" section earlier in this chapter.

Do not shop for computer parts with maximum performance in mind, as you might with your own home computer. Save money, keep things cool, and enjoy the quietude of smaller fans by going with the slowest Pentium machine you can find. For dedicated work like this, where you won't be doing anything but decoding MP3, you can get by with a Pentium 75 or 100 and 8 or 16 MB for RAM. You'll be amazed how cheaply you can purchase a machine like that at online auction sites. Nobody wants them but you.



If you're brave and willing to work with very small parts, you can always rip the guts out of a laptop and place them in a custom-designed case, and possibly even alter the audio path if you're skilled with printed circuit boards and soldering irons.

Another nice thing about building your own box is that you can decide which sound card goes into it. Since you're going to be interfacing with your home stereo, don't skimp on the sound card—get the best one you can afford. If you've got a digital receiver or outboard DAC, be sure to get a sound card with digital outputs to maximize quality. There are also a couple of manufacturers of dedicated sound cards for just this purpose, with MP3 DSPs soldered in place. These may also have RCA outputs, so you won't need an external DAC. See www.interlog.com/~miclee/mp3.html for one such example.



If you've got a slower machine, or one that for some reason will also be busy doing lots of other CPU-intensive tasks, take a look at *lp3*, a dedicated MP3 hardware decoder that hangs out of the parallel port and talks to plugins for WinAmp and other players: lp3music.com. With one of these, you can use a machine much slower than a low-end Pentium or high-end 486—you'll probably get good results even with a 386.

Operating system

If you're building a computer-based system and depending on existing software, your machine will of course need an operating system. While many people use Windows 95 or 98 as a base because that's what they're most familiar with, it may not be the best solution for a standalone MP3 unit, for several reasons:

Long boot times

Granted, you may end up leaving your machine on most of the time, but when you do turn it on, you'll need to wait 30–90 seconds, depending on the configuration. If you do use Windows, make sure you aren't loading anything

unnecessary in the startup group. Windows can, however, be configured to boot straight to DOS, which will save greatly on boot time. In fact, if you know you want to go the Microsoft route, try and get a hold of an old DOS installation, and don't even bother with Windows. Understand, however, that this may present other problems—most (but not all) existing solutions depend on the presence of WinAmp or other GUI Windows players.

Resource consumption

Because you'll probably be going with inexpensive, older, slower hardware, you want as much of your horsepower and memory as possible available to the task of playing MPEG audio, not being soaked up by the operating system. Even when running nothing at all, Windows consumes more resources than the alternatives.

Stability

When was the last time your CD player crashed? When was the last time Windows 98 crashed? Need I say more?

Hackability

Because you're creating a custom solution, it may be beneficial to be able to remove parts of the operating system that are unnecessary, or at least opt not to load them at boot time. If you think you may want to alter the operating system to suit your needs, Linux will be your best solution, as its source code is completely available and open to you. BeOS is less hackable than Linux, but you can get in at a fairly low level and choose not to load unnecessary components. Very few aspects of Windows can be altered by the end user at a low level. If you do intend to build your own Linux kernel or alter the operating system substantially, you will of course need some C or C++ experience.

While thousands of people have created Windows-based standalone MP3 players, you'll probably end up with a more satisfying end-product by going with a Linux- or BeOS-based system. Linux is the most flexible of the lot, and you'll find many tools already out there in the field that work with Linux-based systems. However, working with Linux also requires the greatest degree of technical/programming skills. BeOS has the advantage of being much easier to configure and use than Linux (even easier than Windows in many regards), and is media-optimized from the ground up—it's perfect for this kind of project.* Both Linux and BeOS have the potential advantage that they include built-in telnet servers; if you set up your machine on an Ethernet connection, you'll be able to telnet in from any machine in the world running any operating system and control them from a distance.

* You can find more information on BeOS-based stereo components at www.betips.net/mp3box/.



While it's certainly possible to use a Mac as a standalone MP3 player, this may not be the ideal solution for a variety of reasons. Macs take a long time to boot, MacOS is difficult to program for, cheap parts are not as readily available, and MacOS is not as stable as other platforms thanks to its cooperative multitasking and lack of a memory protection model. Of course, stability isn't usually an issue with units like this, which are just doing a single task for extended periods of time, but you're just not going to find much info out there on how to hack MacOS to output display data to an LCD, or to work via remote control. You could, however, probably create a Perl or other web-based system for controlling an MP3 player on a Mac from browsers on other machines in the house.

Display

Whether your unit ends up in the car or amid the home stereo components, it's highly unlikely you'll want to have a monitor taking up extra space. But you'll still need some way to get a readout of the currently playing song, browse your playlists, and monitor the state of the unit.

This is most often accomplished via an LCD or LED display. Since these don't attach to a video card, you'll need to connect them via another means, typically over a serial connection. You can connect the machine's outbound serial connection to a tethered LCD display in a miniature case, which can be mounted wherever convenient, or you can reroute the connection through the computer case to the front panel and mount the display in a free drive bay.

LCD readouts can be ordered from a number of electronics supply houses. Many people are very happy with the quality of displays purchased from Matrix Orbital, www.matrix-orbital.com, though nearly any display for which you can find a suitable driver is a good candidate. Users looking for an even more polished appearance may want to spend the extra money and get a VFD (Vacuum Fluorescent Display), which enjoy better clarity at a wider viewing angle than do LCDs. Of course, hooking up an LCD isn't as simple as rerouting the normal video display signal to the display unit. You'll need a special driver for your operating system and for your particular LCD. The driver must be capable of receiving data sent to it from a separate software module or from an MP3 player plug-in module. See the URLs listed in the section previously for more information. Precise implementations on various operating system/playback software combinations are very different from one another.

Input methods

Because your unit will probably not have a full keyboard attached, you'll need to find another way to control it. Perhaps the simplest solution is to get a cheapo numeric keypad—the kind with a standard keyboard connector but missing the rest of the keyboard. The nice thing about these is that you can just plug them into the standard keyboard port and be done with it. Unfortunately, they're not exactly elegant looking, and you may not want a separate module dangling from your home stereo.

A much more elegant (and stereo-like) approach is to purchase an infrared remote system, such as the Irman described earlier in this chapter. One of the great things about the Irman is that plug-ins for Windows and Linux already exist (the Linux source code should compile in BeOS without much reconfiguration), so no programming experience is necessary and a big part of the battle has already been fought for you. Setup should be a breeze.

If you want buttons on the front of the unit like normal stereo components, you're definitely going to have some more work to do, and will need some electronics skills. Some users have figured out how to split open a normal keyboard, remove the controller chip and its lead wires, then mount the chip inside the unit and connect the leads to the buttons on the front of the unit from inside the case. You'll need to do some sleuthing to isolate a set of wires, determine the signal they send, and map them to the buttons you install on the front of the unit. The other lead running out of the chip will remain connected to the motherboard where the keyboard normally connects. One thing to keep in mind is that poorly done buttons can easily make a nice case look like a hack job. If you go pure infrared, you can keep the case design very clean.

Software

Finally, you'll need to decide which MP3 player you'll be using in your system. Since the usual graphical interface isn't an issue, you should definitely consider command-line players, since these consume fewer resources and are likely to be easier to control from the keyboard, which is essentially what you're doing with any of the input methods described above. However, many GUI MP3 players can also be launched and controlled from the command line, or have plug-ins to enable keyboard control, so graphical players are not out of the question. As long as you can find a way to create an interface between your chosen input method and the software, you're sitting pretty.

If you're building a Windows machine based on infrared input, definitely check out the Ampapod and Irman plug-ins available at www.winamp.com/plugins/. Just specify the COM port through which your infrared receiver is attached, tell the plug-in which buttons on the remote should map to which WinAmp functions, and

you're set. Linux users can use LIRC (Linux Infrared Remote Control; details at fsinfo.cs.uni-sb.de/~columbus/lirc/) in conjunction with the IRMP3 software (details at www.fasta.fb-dortmund.de/users/andy/irmp3/).



If you're using Linux, you'll need to do some tweaking to get the operating system to boot directly into operational mode without prompting the user for a login. If you don't have much experience hacking Linux, ask around in the community for a custom init pre-configured to bypass the login sequence. If you want to do this yourself, there are a few gotchas to watch out for. You don't want to simply set up your system to boot into runlevel 1, because while that will put you directly into single-user mode, it will also prevent you from using some important services, like networking. The secret is to change a single character in `/etc/inittab` and the system will boot to root access without prompting you for a password. Specifically, you want to change:

```
id:1:initdefault
```

to:

```
id:3:initdefault
```

If your system boots straight to the xdm graphical login manager, change `id:5` rather than `id:1`. Note, however, that there are the usual risks associated with having the box running as root all the time, but this is a stereo component, not a public web server.

If you're working with keypad or button input, you'll probably have to create a custom solution (remember that the kit available from www.carplayer.com includes the software you need to do this, in case you don't want to create it from scratch).

Many projects use rxaudio (www.xaudio.com) for its flexibility and the ease with which it can receive control signals from scripts and custom programs, thanks to a companion utility called rxcontrol. IR support is also possible through rxcontrol. See www.softlab.ece.ntua.gr/~sivann/rxcontrol/ for details. Xaudio is available for nearly all platforms.

Networking alternatives

At several points in this chapter, I've mentioned the advantages of making sure your MP3 player is networkable. While most of the commercial solutions available at this writing didn't have network capabilities, many of the devices being built by hobbyists and hackers do. The prospects of being able to load your MP3 player via FTP, to manage your playlists via telnet, or to select tracks through a web interface from another machine are all irresistible.



If you're building a Linux machine with networking capabilities and want to get really fancy, you can store all of your songs in a database such as MySQL and interface with it through a web server extension such as Perl or PHP. Myriad Perl modules are available so you won't have to create everything from scratch. See igalaxie.com/ltt/mp3/php3/ for more information on PHP solutions, and www.cpan.org for Perl modules. Once configured, you'll be able to bookmark a web page in a browser on another machine in the house, see available playlists, and select songs on your stereo just by clicking on them in your browser. Windows users can of course use Microsoft Access or other Windows database software that can be accessed from the command line or external programs to create custom solutions. If you're building a BeOS-based system, download MP3box from www.betips.net/software/, which does the same thing but uses the Be File System as a native database rather than requiring additional database software.

Even more irresistible is the prospect of *not* having to transfer files between your computer and your home stereo via FTP. If you're crafty and have an always-on home server or "convergence device," you can make this the central repository for all MP3 files in the home. The stereo playback unit or units would still have a sound card and Ethernet capabilities, but would be configured to stream MP3 from the server on-demand and in real time. The playback units wouldn't actually store any MP3 files at all. One could theoretically construct such a system so that multiple playback units in the house could all siphon down separate files or playlists, so each family member could hear the music they want to hear.

However, not everyone has a home network already in place, so these possibilities aren't always viable. If you don't have an existing network and are considering it, take a look at systems from X10 (www.x10.com). This company has long specialized in remote control of lights and alarms, and provides systems for automating many aspects of the home environment, from the simple to the complex. Responding to the desires of MP3 fans, X10 makes available a system called MP3 Anywhere, which lets you beam MP3 audio through the air or right through walls (using 2.4-gigahertz wireless signals), from a computer to the home stereo, without the need to run cables, install a network card, or purchase hubs. Because the system comes with a remote control, you don't need to be sitting at the PC to control the flow of music. The MP3 Anywhere system comes with send/receive units, a "MouseREMOTE" remote control, and software interfaces for talking to the most popular Windows MP3 players. This way, you can still sit at the couch in the living room, but control the playback by pointing the remote at a wall, toward a computer loaded with MP3 files in another room.

The disadvantage to MP3 Anywhere is that you don't have the complete flexibility of possible solutions that you have with a generic TCP/IP connection. On the other hand, you may not need to. Why transfer MP3 files from your computer to another storage unit in your home stereo if you can just beam them through the air as needed?

More information

This is a book on MP3, not on building hardware or hacking operating systems. Creating a custom system for MP3 playback can be very satisfying, but it's not something to be undertaken lightly. You'll definitely need to learn more than what's presented in this book. Here are some good starting points:

Convert your PlayStation into an MP3 stereo component

www.mp3psx.com/info.html/

Information on BeOS-based stereo components

www.betips.net/mp3box/

The InMotion car player

www.jarcom.com/inmotion/

KarPC: Automobile MP3 stereos pre-assembled or in kits, for most operating systems

kpc.phester.org

Kits and prebuilt players for home and car

www.carplayer.com

LC-DAT

www.public.iastate.edu/~jmesterh/lcdat/

Linux MP3 Players Project Page

www.ccs.neu.edu/home/bcbafj/mp3.html/

lp3: Dedicated MP3 hardware decoder unit

lp3music.com/

MP3 Server Box

igalaxie.com/ltr/mp3/

MP3-o-Phono: Plays MP3 CD-ROMs in a standalone Linux-based unit

www.softlab.ece.ntua.gr/~sivanni/mp3ophono/

MP3God: The Car MP3 Player

www.scott.kincaid.net

MP3Hardware

www.mp3hardware.com

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.brrc.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, Real Networks, 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

8

Webcasting and Servers: Internet Distribution

In this chapter:

- *The Fundamentals of Internet Distribution*
- *Offering Files for Download*
- *Webcasting: Real-Time MP3 Broadcasting*
- *An Interview with MP3.com's "High Geek"*

It goes without saying that the Internet made the MP3 revolution possible, and by now you've probably spent a good bit of time hoovering tunes out of the ether and onto your own machine. But what if you want to make your own MP3 files available to others over the Web? What if you want to become a webcaster—set up your own Internet radio station and broadcast tracks in real time out to the music-hungry masses? This chapter covers distribution of MP3 files from web and FTP servers, shows you how to set up an MP3 broadcast system, and covers the bandwidth issues that invariably come into play. Note that this chapter does not cover playing back streamed MP3 audio. That's covered in the section "Listening to MP3 Streams" in Chapter 4, *Playlists, Tags, and Skins: MP3 Options*.

The Fundamentals of Internet Distribution

If you're confident that you have the right to distribute a particular collection of MP3 files, you've got some decisions to make. Do you plan to serve them from an account with an Internet Service Provider, or will you be running your own server? Are you setting up an FTP site for power users, or would you like to build a web site so users can simply click on links to begin their downloads? If you're working with an ISP, are you confident that you're not going to generate too much traffic? Most providers place upper limits on the amount of throughput they'll allow to move through your account without incurring additional fees. We'll look at these issues in the first part of this chapter.

If you're running your own server from home or the office, you have to consider bandwidth issues from two points in the chain: the speed of your personal Internet connection, as well any limits being placed on throughput by your upstream

provider. There are simple formulas you can use to estimate the amount of bandwidth you're going to need, but keep in mind that your estimates are only that. Without knowing how fast your user's connections are going to be, there's no way to accurately determine up front how fat of a pipe you'll require.

If you plan on offering MP3 files on a permanent basis or want to be able to establish a presence quickly, you may want to look into some of the commercial options available. Many high-bandwidth sites exist solely for the purpose of helping artists to host their MP3 material easily, to make your content searchable, and to be listed in important indexes. Going with a commercial option can help make sure your content isn't being overlooked by the public. More on that in the "Commercial Options" section of this chapter.

Finally, you may decide that you want to offer more than simple file downloads to the public. If you'd like to run your own Internet radio station, there are plenty of tools available to help you do that. The simplest method of MP3 streaming is "MP3-on-demand," or "pseudo-streaming," which can be set up from any web server. More sophisticated Internet disc jockeys will want to set up a true broadcast server, in order to be listed in the indexes at www.icecast.org or www.shoutcast.com and to offer bandwidth-optimized audio streams to their listeners. The biggest difference between these two types of MP3 streaming is that true broadcasting is real-time. Like a radio station, users hear whatever is being broadcast at that moment, while pseudo-streaming is really more of a cross between a download and a broadcast. MP3-on-demand is covered in the "Offering Files for Download" section, while true broadcasting occupies the entire second half of this chapter.

Offering Files for Download

When deciding whether to offer your files for download or via real-time streams, remember that if you stream your files, most users won't be able to save the files to their own systems for later enjoyment. Remember also that unless you own the copyright to the files in question, streaming is your only legal option.* Assuming that you do own the copyright to your files, there are a number of ways to make your files available for download to other users.

Serving from an Internet Service Provider

Virtually every dial-up account with an ISP includes access to server space on the ISP's machines. Since ISPs are typically connected to very high-speed Internet backbones, speed is not usually an issue. However, just because the speed is

* And even then, you may have licensing issues to work out.

potentially there doesn't mean you automatically have access to it. Many ISPs place upper limits on outgoing transfer speeds from their servers in order to distribute loads fairly among their many users. In addition, many ISPs place an upper limit on the cumulative amount of data that can be transferred during a given period (usually one month). Exceed this amount, and you're liable to receive a hefty surcharge on next month's bill. You may not even receive notification when the limit has been exceeded, so be careful.

Legal Issues

Of course, there are legal issues to contend with as well. While copyright law is the same whether you're serving from the home, the office, or from an ISP, service providers may function as watchdogs, and revoke your account if they discover that you're distributing MP3 files illegally. While telecommunications laws dating back to the turn of the century have established that providers are not responsible for the content of transmissions flowing through their equipment, lawyers still wave big sticks. Precise interpretation of the law is sometimes made on a state-by-state basis, as well—in at least one case in the mid-90s, a service provider was arrested for allowing smut to travel through their system and onto the computers of juveniles residing in another state.

Rather than face court battles and their associated costs, service providers usually just comply with cease-and-desist letters sent to them by record industry lawyers, and will revoke accounts belonging to users who are in violation of the law. This is why searching the Internet for specific songs results in so many broken links and dead sites.

There's only one law that applies to offering MP3 files for direct download, either over the Web or through an FTP server: If you're the artist of the music being offered and you don't have a contractual obligation with a label or other agency, go for it. If you don't have rights to the music in question, don't do it. You have no legal right to redistribute the works of another artist, period.

Of course, not all MP3 files are music files, so the above might be more accurately and succinctly stated: If you're the copyright holder of the work in question, you can do whatever you like with it. If you're not the copyright holder, you're not legally allowed to do anything with the work without prior permission of the copyright holder.

Standard web downloads

If you choose to serve legitimate MP3 files from an ISP, you can simply transfer them into place on the server with any FTP client and link to them from your web pages like so:

```
<a href="../../audio/mp3/neds_garage.mp3">Ned's Garage</a>
```

No special configuration of the server should be necessary. If the browser window starts to fill with ASCII garbage when users click such a link, you'll know that the server does not have a MIME type established for the MP3 file type. In this case, send a polite message to the server administrator asking them to set up MIME types as described in the following section, "Running Your Own Web Server." It may also be the case that your browser is not configured correctly to handle MP3 MIME headers sent by the server. See the sidebar "Configuring Your Browser" in Chapter 4.

Internet Storage Locker

Realizing that many people would like to be able to access their music as easily from work as from home, a company called myplay (www.myplay.com) has created an "Internet storage locker," which lets users store up to 250 MB of MP3 audio in a private location on the Internet. By using streaming players, users can listen to their own music collections regardless where they are at the moment—if you've got connectivity, you've got access to your music. myplay licenses tunes from major labels, which users can store in their lockers for free. In addition, users can upload music from their own computers, search for MP3 files in the wild and have them transferred to their lockers automatically, or drag MP3 URLs into a special Windows utility called the mplay DropBox, which will automatically upload those tracks to your locker.

Of course, myplay is potentially treading through a number of legal gray areas here, but their legal statement "House Rules" is very clear about the fact that users caught using the system for illegal purposes will be terminated immediately. Nevertheless, it's a system with a lot of potential, especially as broadband access becomes more ubiquitous.

Offering FTP access

Most ISPs do not offer anonymous FTP access to the world at large. Instead, they provide their subscribers only with username/password FTP accounts. While it's technically possible to give the world access to your files via FTP through an ISP by handing out the username and password, doing so would be foolish, as the FTP directory location probably points to the same directory where your HTML files are stored, and anybody would have unfettered access to your site.

If your ISP does allow for anonymous FTP access to a directory location separate from your HTML files, you're in business. Your ISP will provide instructions for using this service. Note that you will probably have to request this service specifically; your ISP may offer the service without advertising its availability. Users will be able to log into the FTP site with the username "anonymous" or "ftp" and a password consisting of their own email address.

If you find that anonymous users are not able to download files from the FTP server, you may have to tweak “permissions” settings for the parent directory containing the files (although in most cases the service provider will have established reasonable default permissions). Many graphical FTP clients include a graphical permissions control panel. Navigate to the directory containing the files and grant permission to the world to read (but not to write or execute) the directory. Alternatively (assuming you’re dealing with a Unix/Linux server), you may be able to telnet in to the server and type:

```
chmod 744 directory_name
```



When uploading files to the FTP server, make sure the FTP client is set in binary mode (you should find this control somewhere in the FTP client’s interface or perhaps buried in the preferences). In most cases, this will be default setting for the client and you won’t need to touch a thing. If you’re using command-line FTP, just type `bin` at the prompt to set binary mode. If you fail to transfer files in binary mode, they’ll be corrupted and unplayable by your users (unless they run the files through a utility like `uncook`, covered in Chapter 4). Make sure your users know they have to download files in binary mode as well!

MP3-on-Demand

As described in the “Listening to MP3 Streams” section of Chapter 4, users are always free to play MP3 files directly from your server, rather than downloading them first. As a webmaster, you can make sure this happens (rather than it just being an option) by setting up an “MP3-on-demand” system. While you’re at it, you can also let users listen to entire playlists from your server by clicking a single link.

Unlike true MP3 broadcasting, as described later in this chapter, MP3-on-demand (or “pseudo-streaming”) doesn’t require much setup on the part of the webmaster. Just create a plain text document containing full URLs to MP3 files on your server, one to a line, and save the file with an `.M3U` extension (again, “.m3u” stands for “MPEG URL”). For example, if your server is located at *www.mpegjam.com* and you store your files in a subdirectory called “tracks,” an M3U file might look like this:

```
http://www.mpegjam.com/tracks/J/johnhardy.mp3
http://www.mpegjam.com/tracks/W/wild_ride.mp3
http://www.mpegjam.com/tracks/B/biloxi_blues.mp3
```

and saved as *Blues_Mix.m3u*. Next, you must make sure your web server's MIME database includes an appropriate association between the *.M3U* extension and the appropriate MIME type. Setting this association is dependent on the exact server you use, but it should be pretty clear. The association you want to set is:

```
audio/x-mpegurl .m3u
```

Once the MIME type has been set and the M3U file placed on the server, a user accessing a link to an M3U file will find that his or her preferred MP3 player is launched and the first track in the list begins playing. Better players will show the entire M3U list in the playlist editor, allowing users to skip around between tracks at will.

Bandwidth considerations for MP3-on-demand

It is unlikely that many of your users will have fast enough connections to work comfortably with pseudo-streamed MP3 files at the standard bitrate of 128 kbps. In addition, your server probably won't be able to handle too many simultaneous users without maxing out its own available bandwidth. Therefore, you'll probably want to do one or more of the following:

- Encode the MP3 files you serve this way at a lower bitrate, or in mono, or at a lower sampling frequency.
- Tell users that your site is for high-bandwidth connections only.
- Limit use of your site to a high-speed LAN in your home, school, or organization.
- Use an option in your web server software to accept only a few simultaneous connections.

Because of these limitations, serious webcasters go with real (not pseudo-) streaming solutions, which are able to downsample or re-encode files at a lower bitrate on the fly, and offer much more control over the structure of the broadcast. However, just because MP3-on-demand configurations don't automatically downsample on the fly, as true broadcast servers can be configured to do, doesn't mean you can't set things up to do so manually. For instance, you might want to write a Perl script which, when invoked, pushes an MP3 file through a command-line downsampling routine and then pipes the output of that routine back to the user. Keep in mind, however, that such a solution is going to be very CPU-intensive. If you have plenty of storage space, you can get around this problem by running a script that will downsample all of your files to various bitrates or frequencies ahead of time. Your web interface can then offer users a choice among multiple connection speeds/quality levels.

If you'd like to go with a ready-made solution, check out Bitcasting's "MP3 Radio Station In A Box," which does just that (and works with RealAudio files as well). See www.bitcasting.com.

Commercial Options

If you don't want to be burdened with some of the more arcane technical responsibilities of running your own server, there are plenty of sites out there more than willing to provide you with free disk space and web pages. All they get in return (typically) are the advertising revenues generated by placing banner ads on your pages. Here are a few of the most popular, but there are dozens (if not hundreds) of similar sites out there, all with slightly different approaches and atmospheres.

MP3.com

www.mp3.com If you're an artist, you'll probably want to both set up your own web or FTP site and make sure your content is listed in one or more of the popular MP3 directory sites. The most popular, of course, is the ubiquitous MP3.com. If you're an unsigned artist or just a musical hobbyist, MP3.com offers the potential to get your music into thousands or millions of new ears, at no cost to you. The site also offers discussion boards where artists can gather and discuss all aspects of the Internet music distribution business. Artists can also keep an eye on MP3.com's dynamically generated download statistics to see how popular your music is in comparison to other MP3.com artists.

In addition, MP3.com will optionally create "DAM" (Digital Audio Music System) CDs of your music (in both MP3 and standard CD audio formats), let you determine the price, and keep a full 50% of the revenues—far, far more than artists can get through even the most generous record contracts. MP3.com users will be able to buy DAM CDs of your music as they surf your site.

As this book was going to press, MP3.com was beginning to position itself as a *Music Service Provider*, or MSP, with the intent to let users get at their music from anywhere on the planet. This potentially means that users will be able to keep a personal music collection on MP3.com servers, and then listen to it from home, work, a friend's house, or wherever they may happen to be. Rather than requiring users to download their files to each location, the system will probably allow users to create online playlists, and stream custom collections of music to themselves from any location. Playback may not be limited to MP3 software, either. Internet-savvy hardware may be capable of interfacing with MSPs such as MP3.com's as well. Of course, users with fast Internet connections will probably stand to gain the most from MSPs.



MP3.com isn't the only site on the Internet offering 100% legitimate, free songs. For another approach to the same basic model, take a look at www.unsigned-music.com. With the great success MP3.com has enjoyed, don't be surprised to find similar sites cropping up all over the place.

EMusic.com (formerly GoodNoise)

www.emusic.com EMusic works similarly to MP3.com, but charges \$.99 (at this writing) per track for music downloads. The site has gained a reputation as being at the forefront of digital commerce for music. Why would users pay for tracks at EMusic when they can download them for free at MP3.com? Because EMusic works only with existing labels and signed artists, they offer music that may already be popularly heard on the radio. Whereas MP3.com is aimed more toward the joys of discovering new music you've never heard of, EMusic caters to fans looking to legitimately download music by their favorite artists. However, the unsigned artist isn't entirely left out of the loop; the site partners with the Internet Underground Music Archive (IUMA) to provide access to lesser-known artists.*



Some of the principal minds behind EMusic.com are, perhaps not coincidentally, also some of the founders of Pretty Good Privacy, the Internet's most popular encryption technology. In an atmosphere where labels would do anything to build excellent copy protection into digital music, don't be surprised if EMusic turns out to be a leader in this field at some point in the future.

Internet Underground Music Archive (IUMA)

www.iuma.com If your experience with the Web stretches back to the beginning, you'll probably have a fond place in your heart for IUMA, which became a home for artists to offer music to their customers digitally long before the MP3 revolution. The site has offered downloadable music in many different formats over the years, and currently focuses on MP3 and RealAudio. IUMA goes one step beyond MP3.com by offering not just server space for audio files, but the ability for artists to create their own complete web sites on IUMA servers, so artists can completely customize the viewing and learning experience for their audiences. IUMA's Artist Uplink feature makes it easy to create online calendars for shows and tour dates, a clean URL (e.g., <http://bandname.iuma.com/>), the addition of images and

* EMusic acquired IUMA in 1999.

Case Study: Bruce Lash Recounts His Experience with MP3 Distribution

Bruce Lash is an independent musician from Chicago, Illinois who has been experimenting with digital music distribution for quite some time. To get a sense for the ways in which the MP3 "revolution" affects artists, I asked Lash to recount some of his experiences promoting his music via MP3. As you'll read below, the MP3 dream doesn't always live up to the hype—while digital music distribution can definitely open new avenues to artists, it's not without its pitfalls, and much of its potential has yet to be realized.

I chose to interview Lash for several reasons: 1) Bruce is (for the most part) a one-man band, he does not perform live—his music is only viable in a recorded context and is therefore well-suited for digital distribution and promotion; 2) Bruce typifies the "do-it-yourself" approach to music making; 3) After listening to the music of hundreds of artists downloaded from MP3.com, I found that it was Bruce's tunes I kept coming back to again and again; his music resonated for me (I'm a big fan of Robert Wyatt, Syd Barrett, Nick Drake, Beck, and the like), and inspired me to buy CDs of every musical project he's been involved with to date. Lash typifies the case of the truly great singer/songwriter struggling independently for greater recognition. Of course, there are thousands of other musicians who walk in similar shoes; we're just using Bruce as an example. See Bruce's site at home.att.net/~bblash, or search MP3.com for "Bruce Lash."

Digital music distribution has been a double-edged sword for me. On the one hand, it's very exciting to be able to be heard around the world (this is especially true for someone (me) who doesn't perform before a live audience). On the other hand, it's easy for one's expectations to be unreasonably high. Wasn't that the way the Internet was sold to independent musicians?

People from all over the world have visited my site—I've sold CDs to people in places as unlikely as Pakistan. I once received a nice note from somebody in rural Costa Rica who liked what he heard on my pages, but didn't have the money to buy the CDs. Would I send him some anyway? You bet I would. I think it was one of the best things I did that year. His address read like bad directions through the Southern part of the country, literally saying: "Go 350 miles south of the main city to the pink house next to the big church and ask for Celeste." And it got there. That, to me, is a good Internet experience. There are people in Russia and Bosnia who wrote to the Virgineers [another of Lash's projects] asking for free CDs because they couldn't get currency out of their countries. Did we send them CDs? Yes. If our free little CD makes someone's life a little better in the Balkans, who are we to ask for money? So in this sense, yes, the Internet has greatly expanded the reach of my music. But I haven't really sold much of my music through MP3.com. In truth, I've had better luck selling through an on-line store (www.cdbaby.com).

—Continued—

I've never really liked MP3s. They take too long to download, and take up too much space on my hard drive. At the defacto bitrate of 128 kbps, they sound squirrely. But if one wants to have a business as an independent artist on the Internet, one had better have MP3s available. I got into MP3.com pretty early. At the time, it seemed like a place that really wanted to "level the playing field," and I liked their attitude. I've had a good number of downloads over the months, and my songs were featured on the site at one point, which led to lots of downloads. I noticed, however, that the song itself was getting more hits than my artist page. People were apparently there more for the free music than to learn about the artist. As much as I wish more people were interested in really learning about artists, I can't fault them for that—everybody loves free stuff. I also became somewhat disillusioned when MP3.com came out with a program wherein artists pay to be a featured artist; it became clear to me that they had moved on from the "level playing field" phase of their operation.

But when it came time for me to manufacture "Prozak for Lovers," I was in email contact with a guy from Oklahoma City who had a friend who worked for a CD manufacturing company, and he was able to get me a good deal on the manufacturing cost. I wouldn't have known anything about the company, let alone get this deal, had I not been active on the Internet.

I've spent nearly two years promoting myself on the Internet. I posted on message boards (I still get hits from many of those posts). I sent CDs to on-line reviewers to get my name out there. I became a part of the independent music community as it exists on the Internet. I've also learned that non-Internet exposure is valuable. In fact, I've become more interested in non-Internet promotion as of late. Getting my music on the radio, or getting a review in a newspaper or magazine is very exciting. When you do the math, you find that people with access to the Internet are still a small lot compared to the number of people who have access to radios and newspapers and books. Despite the hype, much of the potential of digital music distribution has yet to be realized for artists like me.

As much as I've benefited from all of this experience and exposure, I've also noticed that my priorities have become skewed. I'm checking my stats too often rather than making music. I'm spending too much time thinking that something big might happen. Rock stardom is a very, very tall mountain to climb, and I haven't even gotten to the foothills yet. But I have no need for the journey—I'm able to produce quality recordings that I really like. If I didn't sell any CDs, I would still be able to make those records and it wouldn't matter whether anyone buys them.

—Continued—

So, I've made a pact with myself to spend more time making music than promoting it. I'll keep my web pages up and respond to people who listen to my work and want to comment on it, but I will not take my eyes off of the prize: the chance to realize the sounds I hear in my head and massage them to the point where they sound good to my ears. My prize is my own musical freedom and happiness. I'd be foolish to want anything more.

biographical text, and more, all manageable through a web-based interface. Despite the massive popularity of MP3.com, IUMA is the original, folks. The real deal. Don't let anyone tell you different.

GarageBand.com

www.garageband.com Occupying a position somewhere between promotion of unsigned artists (like MP3.com) and offering major label music for sale (like EMusic), GarageBand.com aims to work with the “bubbling” method alluded to in Chapter 1, *The Nuts and Bolts of MP3*. By allowing users to vote for and review the music of struggling musicians, GarageBand.com hopes to determine which of their artists shows commercial potential statistically, and then to offer those artists actual recording contracts. According to the site's founder, only about 600 of the 30,000 CDs published each year are actually profitable;* basically, just the discs that make it into the Top 40. By gauging audience reactions before contracts are signed and CDs pressed, GarageBand.com hopes to publish a higher proportion of profitable music. The Talking Heads' Jerry Harrison is one of the site's co-founders.

Bandwidth: How Much Do You Need?

MP3 files may be small compared to uncompressed audio sources, but when you start tossing a bunch of them over the wires, their cumulative impact on available throughput can be huge. There are many factors that come into play when deciding how much throughput you need to keep your users happy. How many users will you be serving simultaneously? If you're broadcasting, you'll almost certainly be *downsampling*—lowering the quality of the audio to make it more Internet-friendly. But by how much? If you're offering files for download, the equation is purely a function of how many users you allow to connect simultaneously. If you limit the number of connections, you may end up turning some users away, but the ones who do get to connect will be guaranteed good throughput. If you don't, you can offer more simultaneous downloads, but all downloads will be slow. In

* Ironically, this is just the kind of statistic labels use to justify their existence: That with such a low success rate, we need someone to pour tons of money into distribution and promotion on lots of new music so that a few artists can at least hope to succeed.

essence, it's your decision whether to place the bottleneck at their end of the line, or at yours. No one likes an unresponsive server.

So, exactly how much bandwidth do you need to serve up MP3 files from a web or FTP server you run yourself? You'll need to do some ballpark calculations, then do some real-world testing to make sure your calculations hold up okay. Trouble is, you don't have advance knowledge of the speed at which your users will be connecting, so the best you can do is to estimate. Having ten simultaneous modem users is very different from having ten simultaneous DSL or cable users:

U = Number of simultaneous users
S = Max connect speed of users (estimated)
O = Overhead. General Internet congestion. Estimated constant of .9
B = Bandwidth required
 $U \times S \times O = B$

So let's assume you're going to allow 10 users to connect to your site simultaneously and you figure they're probably all using 33.6 K modems:

$$10 \times 33.6 \times .9 = 302.4$$

So your upstream connection must be capable of serving 302.4 K per second. In other words, your upstream path should be able to bear nearly ten times what a standard modem can download. If any of your users are on faster connections, your line will be saturated, and some of the users will get slower transfer speeds than their equipment is capable of.



It's important to remember that transfer speeds are often measured in bits per second, not bytes per second. There are eight bits in a byte, so 302.4 kbps is roughly equal to 38 kilobytes per second (38 K). If you've got a symmetric 384 kbps DSL connection, you're probably used to seeing your download speeds top out at around 42K (most browsers report download speeds in kilobytes/sec), so a 38K threshold would serve 10 simultaneous modem downloads pretty nicely.

Because MP3 fans are often download hogs, they're more prone to have fast connections than normal Internet users, so let's assume that half of your users are on 768 kbps DSL connections:

$$\begin{array}{r} 5 \times 768 \times .9 = 3456 \\ 5 \times 33.6 \times .9 = 151.2 \\ \hline \text{Total: } 3607.2, \text{ or } 450\text{K/sec.} \end{array}$$

As you can see, a few fast users can change the picture pretty dramatically—like grocery bills, these things can add up very quickly. Most web and FTP server software worth its salt includes a setting that will let you specify the maximum

number of simultaneous users. More advanced software even offers the ability to let the administrator “throttle,” or limit the maximum bandwidth available to any single user. If you plan to promote your site heavily, either be ready for the onslaught with a big fat pipe, or be prepared to face the wrath of disgruntled users.

Colocation

If you want to run your own server but can't get a fast enough connection to your home for an affordable rate, surf the sites of the ISPs in your area, looking for colocation options. With “colo,” you supply the machine, install and set up the web server software, and maintain the site. But you store the machine on the premises of the service provider, taking advantage of their proximity to the high-speed Internet backbone. Most colo plans will also guarantee that someone will be around to reboot your machine if it hangs or crashes, or provide other QOS (Quality of Service) guarantees.

Free bandwidth

If you don't need a lot of server space, you may want to look into one of the generic “free web space” options such as GeoCities, Tripod, or XOOM. All run on a similar business model: Free web space in exchange for advertising banner eyeballs. The disadvantage to sites such as these is that they have a very low “cool” factor, and you don't get grouped together with other artists, have the opportunity to be featured as an “Artist of the Day” or similar, don't show up in the alphabetized artist indexes or searches, and may be subject to restrictions or limitations that don't fit your style. If free is what you're after, you're better off uploading to MP3.com or a similar site.

Running Your Own Web Server

It's easier than you may think to set up web server software on your own machine and grant access to the entire world. One of the nice things about working directly at the helm of your own server is that you don't have to build your site locally and then FTP it into place; all changes you make to your site are live immediately. Of course, there are issues that make this an unrealistic option for many users. The most prominent obstacle is, of course, bandwidth; this is really only an option for those with DSL or cable connections to the Internet (ISDN will do in a pinch, if you don't expect much traffic). Even then, some providers only offer their subscribers fast *downstream* bandwidth, assuming that most of their users will only want to download large files, not offer them for upload. Upstream bandwidth is sometimes constrained to a fraction of the download speed. This type of connection is known as *asymmetric*. Symmetric connections offer the same speed

upstream and downstream, and are better suited for web server deployment. If you're not sure of your upstream bandwidth, check your provider's web site, contact them directly, or run some upload tests of your own. The other big issue is the ever-changing, dynamically assigned IP address (we'll address that in a minute).

Running your own web server doesn't mean you have to have a domain name associated with your machine; a server can run just fine from an IP address alone, though it won't be easy for your users to remember. If you want a domain name associated with your machine, you'll need to work with your ISP to obtain a static, unchanging IP address, then register a domain and ask your ISP to map that domain to your static IP address. Of course you'll want to make sure your machine is always running and always connected.



Setting up a web server is far easier than setting up a *secure* web server, and the importance of server security should not be understated. Once you've opened your machine's door to the rest of the world, there's no end to ways in which nefarious crackers can exploit vulnerabilities in a setup to alter or delete your data. While you're probably going to be safe with quick experiments, be sure to do additional research if you want to do this full time. A good place to start is Simson Garfinkel's book *Web Security and Commerce*, www.oreilly.com/catalog/websec/.

Server setup

Some operating systems come with built-in web servers. Windows NT comes with Internet Information Services (IIS), BeOS comes bundled with PoorMan, and Linux distributions almost always come with Apache. Even if your operating system doesn't include a web server, there are dozens of them to choose from, regardless of platform.

In all cases, setup involves the same simple steps: Choose a "root" directory on your machine representing the base directory for your server, a default filename to be served for directory accesses (e.g., if a file called *index.html* exists in the "images" subdirectory of your server's root and somebody access www.mysite.com/images/ they'll get served with *index.html* automatically), and enable the server. You'll probably also want to configure the server to be launched automatically when you start your machine.

Testing the server

Test your server by placing an HTML document under the server root, then enter your "loopback" address into a browser window, followed by the filename.

Assuming you created a file called *mp3.html*, you would enter one of the following:

```
http://localhost/mp3.html  
http://127.0.0.1/mp3.html
```

These two are simply references to your own machine, hence the term “loop-back” Once that’s working, find out your real IP address and use that instead.

Discovering this address may depend on whether you’re connecting through a network card or over a dial-up connection. An easy way to learn your IP address in Windows is to click Start → Run and type *winipcfg.exe*. Select the PPP or Network Interface Card from the picklist in the resulting panel to learn your current IP address. In BeOS, right-click the Dial-Up Networking icon in the Deskbar and chose “Statistics,” or use the Network preferences panel if you’re not using a modem. In Linux, launch a terminal window and type *netstat*—your available network interfaces will be reported on the first few lines. In MacOS, you’ll need a utility like IP Net Monitor or “Lucky Numbers”; check your favorite Mac software library. If the site works with that address, hand it out to your friends and go to town.

Establishing MIME types

MIME, or Multipurpose Internet Mail Extension, is a specification for formatting non-ASCII messages so they can be sent over the Internet. Originally developed to enable email attachments, MIME now forms the basis for file-transfer communications between browsers and web servers.

Before releasing your site to the public, make sure that when you click a link to an MP3 file, the browser either initiates a download of that file or launches an MP3 player. If the browser window fills up with ASCII garbage (older browsers only) or asks the user what he wants to do with the file, then you’ll know that either the MP3 MIME type has not been established in the web server software or the user hasn’t configured his browser’s MIME types properly. It’s crucial that the MIME type sent by the server matches one of the types registered in the user’s browser so that the proper action can be taken (i.e., start a download or launch an associated application, rather than trying to display the file’s data in the browser window), and as a webmaster, it’s your responsibility to make sure you’re carrying out your end of this deal.

On most platforms and most web servers, MIME types are registered in a table and consist of a simple association between the MIME type and filename extensions. In this case, you want to tell the server what MIME header to send when a file ending in MP3 is accessed:

```
audio/x-mpeg      .mp3  
audio/mpeg        .mp3
```

Two MIME types are listed here because, unfortunately, the industry has yet to agree on a standard MIME type for MP3 files. Technically speaking, `audio/x-mpeg` is the correct MIME type, although `audio/mpeg` is in wide use as well. Essentially, MIME types preceded with `x-` refer to file types that have not yet been codified by international committee. Once a file type is deemed common enough to become a part of "the canon," the `x-` is removed and the type becomes official. Both of the above MIME types are in common use at this writing, even though the type has not yet been made official. You *may* want to additionally establish:

```
audio/x-mpegurl    .m3u
```

particularly if you plan to do any MP3 streaming from your server. Associate these with the extension `.M3U`.

Every web server has a different method of establishing MIME associations. Consult your server's documentation for details (in most cases, the MIME database can be easily found in the Options or Preferences).



BeOS web servers do not utilize MIME tables, because all filetyping in BeOS is MIME-based to begin with. If the MP3 files on your system have correct filetypes, they'll be served up correctly regardless their filename extensions, or even without extensions (though it's a good idea to keep the `.mp3` extensions in place to make users of other platforms happy).

Dealing with dynamic IP addresses

If you connect to the Internet through a dial-up connection, chances are your IP address is drawn from a pool of available addresses and assigned to you at connect time. This is great for conserving on addresses, but terrible for people who want to run servers of any kind. The address you hand out for your server today will probably be different tomorrow, or even a couple of hours from now.

Fortunately, there are a number of ways to address this situation. None of them are as painless as securing a static address, but they do a fine job of working around a thorny problem. One way to get around it is to keep a page at a permanent location, on your ISP's server, for instance. This page will, in turn, simply provide a link or a redirect back to your machine. Every time you connect, a script or program will be run automatically that will regenerate this page and FTP it into place at the permanent location. Such a page can even tell users when your machine is off-line. Search your favorite software library for "dynamic IP" and you'll find many such solutions (try *hotfiles.com* for Windows, *macdownload.com* for MacOS, *freshmeat.net* for Linux, or *bebits.com* for BeOS).

In addition, there are web-based services that will do this job for you. In fact, these services go one better by allowing you to have an elegant URL and dynamic IP addressing at the same time. A good example of such a service is *www.dynip.com*. DynIP will make your machine appear as *mymachine.dynip.com* any time you're logged in, regardless of your IP address at the moment. The client software runs on all versions of Windows, plus some versions of MacOS and several Unix variants, including Linux.

Running Your Own FTP Server

There are three major advantages to running an FTP server rather than (or in addition to) a web server:

- Most FTP clients these days are capable of batch transfers. If you want your users to be able to download lots of files at once, FTP is the way to go, since users will be able to drag multiple files or even entire directory structures down from your machine and walk away while the transfer proceeds.
- FTP servers can exist somewhat more secretively than web servers. Web crawlers ("spiders") will eventually discover and index the contents of most web sites, which means they'll turn up in most search engines. Public knowledge of FTP sites, on the other hand, will generally only get around by word of mouth, or if you want it to. If you only want to give access to a small handful of select people, the rest of the world may never learn you're running an FTP server.
- FTP servers make it easy to establish users and groups with specific levels of permission, so you can offer one set of files to the general public and another set to your close friends, for example. This can also be done by using password protection on web sites, but it's generally easier to configure and manage the permissions lists on FTP servers; this is what they're designed to do.

On the downside, the Web is of course easier for users to navigate. Everybody knows how to use a web browser, while far fewer are comfortable with FTP. And of course you may *want* your content indexed by the search engines.

An FTP server is part of Windows NT's IIS package, is included in every Linux distribution, and is built into BeOS Networking. Third-party FTP servers are available for MacOS, Windows 95/98, and for virtually every platform in existence. Popular choices for Windows 95/98 include War FTPD and WS_FTP Server. Popular servers for MacOS include FTPd and Rumpus. Again, check your favorite software library for details.

Regardless of your platform, be extremely careful when setting up an FTP server for public access. Make sure users only have access to a limited corner of your hard drive, and if you set up access outside of that directory structure for select,

trusted users, test very carefully to make sure you don't accidentally grant that access to the general public. Pay attention to read-only versus read/write permissions. Give a malicious person read/write permissions to the wrong location and they can wipe out entire directory structures right under your nose.



If you've set up an FTP site with a username/password pair but still want to be able to grant easy access to the general public (who may not be familiar with FTP client usage), you can use the following shortcut to let them log in to a protected FTP site through their web browser:

```
ftp://username:password@ftp.mysite.com/directory/path/
```

You can even use this construct in an HTML link:

```
<a href="ftp://username:password@ftp.mysite.com/directory/  
path/">Click here!</a>
```

Most browsers developed since the mid-90s will properly parse out the username and password and pass it to the FTP server, which will then return the contents of /directory/path/ to the browser. But remember: The username and password will be in plain view in the URL pane—you've just thrown all security out the window!

Webcasting: Real-Time MP3 Broadcasting

Just a few years ago, Internet pundits (like me) were waxing poetic about how one day we'd all be able to run TV and radio stations out of our homes as the new era of convergent media and speedy connections dawned. Well, guess what? It came true, and today you can broadcast audio signal out over the Internet in real time, ready to be heard by anyone who cares to listen. Thousands of users are queuing up their favorite tracks and personal creations, doing live call-in shows, variety shows, live interviews, even depressingly juvenile 24-hour live prank call shows.



In fact, just last week, riots again erupted in downtown Berkeley in protest of a real radio station having its freedom of speech curtailed by its parent company. Within hours, an Internet-based replacement show sprung to life and I was able to hear the voices of the "gagged" station employees, liberated by the openness of the Internet and the magic of streaming MP3.

Head over to www.shoutcast.com, www.icecast.org, www.radiospy.com, www.mycaster.com, www.greenwitch.com, www.live365.com, or any other directory of

live MP3 radio, and you'll be amazed at the sheer number and variety of broadcast types being offered. And unlike your local radio environment, these broadcasts aren't constrained by the distance a signal can travel. All of them are available worldwide.

On the other hand, you don't necessarily get the same guarantee of quality you get with commercial or professional radio. Stations come up and down in the blink of an eye, they all broadcast at varying levels of quality, and the vast majority of them broadcast stuff you aren't really interested in hearing (just like real radio!). But at the same time, there are real gems out there: impromptu DJs broadcasting at decent quality levels, playing stuff you've never heard of but fall instantly in love with. It's a brave new world of "pirate" radio, as yet unfettered by the mandates of the FCC, anarchic, wonderful, and buzzing with a level of excitement the Internet hasn't seen since Mosaic hit the scene. Just as the world went nuts a few years ago when we realized that anyone could become a publisher, we now exhilarate in the fact that anyone can become a DJ and be heard around the world, instantly, and nearly for free.

While the notion of not being able to easily save the files you download may seem unappealing to some, the fact remains that not everybody wants to spend hours managing a huge MP3 collection, and sometimes we just want to let someone else control the horizontal and the vertical in the background while we work on other things. There are zillions of CDs and videocassettes out there, but we don't necessarily want to own all of them. The transience of radio and TV have an appealing place in our cultural landscape, and streaming MP3 fills the same kind of position in many people's lives. Many people are very happy to just listen in as the bits float by in the stream, enjoying them as they pass without feeling they have to "own" them.



Note the legal situation changes somewhat in the case of Internet broadcasting. The legal distinction between offering MP3 downloads and streaming MP3 files is simple: If the ultimate destination of your files is the user's hard drive, you need legal rights to the music in question. If the music will be heard by the user but not stored for later playback, you should be able to obtain a "statutory license" for re-broadcast of the music in question. However, exact details on obtaining a statutory license were still being worked out by the relevant organizations at this writing, and negotiations appeared to have stalled. See the "Webcasting" section of Chapter 7, *The Not-So-Fine-Print: Legal Bits and Pieces* for additional information. You'll also find an excellent summary of the ongoing legal wranglings over webcasting in the Beginner's Guide at www.radiospys.com.

Commercial Internet Radio

SHOUTcast and icecast may be providing the most popular MP3 server software, but they're by no means the only sites offering directory listings. Other popular sites taking advantage of the popularity of MP3 streaming include Spinner.com, Tunes.com, SonicNet, and NetRadio. And how do they intend to make money? Without users clicking through dozens of pages to generate ad revenue, these sites look to other means, such as relationships with e-commerce sites like Amazon.com, and affinities with sites who pay a fee for purchases made via links. More importantly, the sites sell CDs and merchandise, and some of them experiment with ads embedded into streamed broadcasts as well. Small potatoes? Hardly. Both ImagineRadio and SonicNet were purchased by MTV in 1999. Meanwhile, a site called MP3Radio.com is partnering with hundreds of traditional radio stations around the world to help them get their standard transmissions and related MP3 downloads onto the Internet. The Web isn't just enabling a new flavor of radio, it's changing the way traditional radio operates as well.

How MP3 Broadcast Works

Streaming MP3 systems (which basically boil down to SHOUTcast and icecast, discussed later), are constructed modularly. A "source stream" module is used to organize and play the MP3 files themselves, and possibly to allow for live mixing of multiple sources, or to offer microphone input capabilities. The source stream may also interact with an MPEG codec to "downsample" the quality of the source stream to make it less bandwidth-intensive and more broadcast-friendly. The source stream sends the signal it generates on to a server module, which takes responsibility for distributing the source stream out to any currently listening clients. The source stream may or may not be running on the same machine as the server software, or even under the same operating system. The server software, in turn, may be redistributing streams from multiple source streams.

In a nutshell, the source streamer may be responsible for:

- "Down-sampling" audio signal to a reasonable distribution level
- Sending the currently playing track to the server
- Identifying the Internet and port addresses of the server
- Establishing appropriate data to be registered in public databases
- Connecting and disconnecting from the server

The server is responsible for:

- Replicating audio data from the source stream and sending it to listeners
- Logging all transactions
- Limiting or controlling the maximum number of concurrent users
- Disconnecting users whose connections have died
- Specifying the format of “metadata”: titles, URLs, etc.

There are several advantages to keeping the broadcast system modular. First of all, it keeps a lot of complexity out of the source stream software itself (which will be WinAmp or Sonique for most users) that isn't needed by people who use that same software for normal (nonserving) playback. Second, it allows multiple users to send their data to a single server, which may remain under the control of a single administrator on some networks. But perhaps most importantly, it accommodates situations where you don't have much bandwidth on the source machine, but can partake of the services of another machine on a high-bandwidth pipe (possibly a friend's, or a machine at work or school). All you have to do is get a single stream of data out to the server, and the server can parse that data out to many listeners simultaneously. The modular system may provide security benefits in some cases. The source stream can be run on a user-friendly but insecure system like Windows 98, while the intensive network needs are taken by a security-conscious system such as Linux.

To get a handle on the possible ways in which MP3 broadcast systems can be set up, study Figure 8-1. In scenario A, the source streamer (WinAmp) is being run on one machine (possibly Windows 98). Music being pumped out of the source streamer is sent to another machine (possibly Linux), which is running the server software. In scenario B, a single machine (possibly Windows 95) is running *both* the source streaming software and the server software. In either scenario, it is the server software that sends the audio stream out to multiple listeners. In both scenarios, the server software broadcasts the fact of its existence (but not the audio stream itself) to a directory listing server, such as the ones run by *shoutcast.com* and *icecast.org*. To learn about the existence of currently running servers, end users access a directory listing site. When they click on a link in the directory, they're connected directly with a streaming server.



Directory listing servers *do not* send audio data to users—they just keep track of currently running servers and help users make the connection. When a streaming server goes offline, it automatically disappears from the directory listing site.

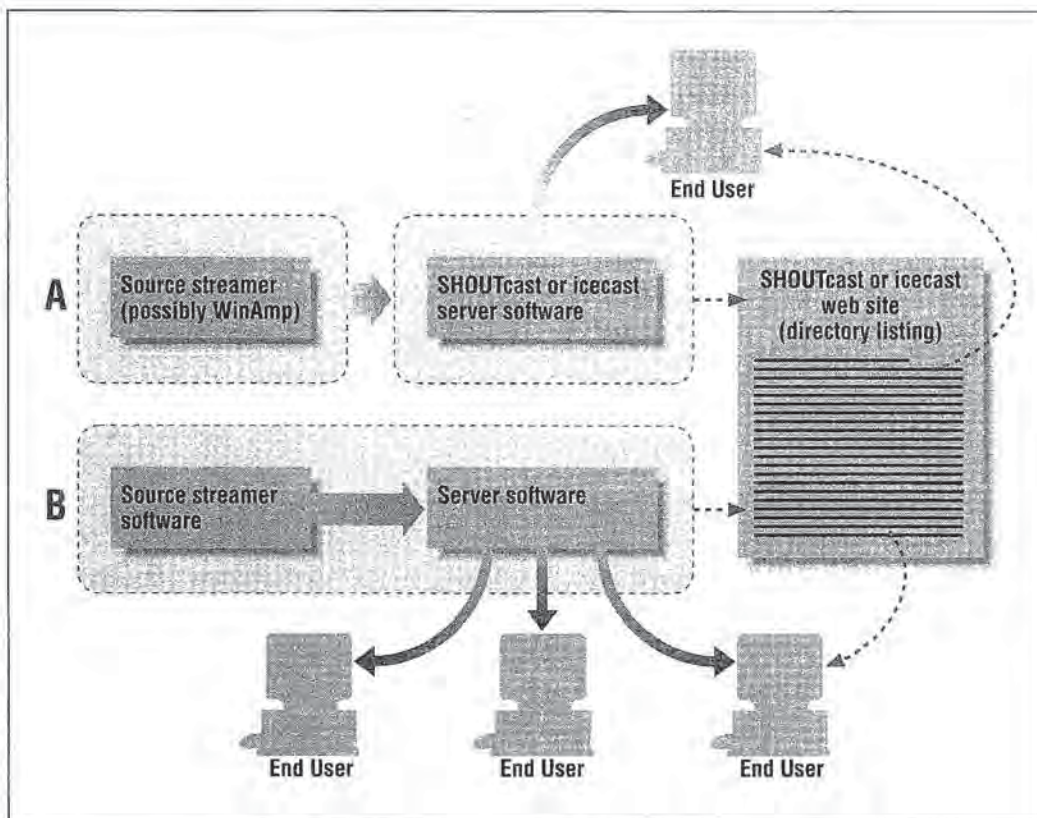


Figure 8-1. MP3 broadcast systems are constructed modularly to allow for maximum flexibility in implementation

The server is, in essence, an “intelligent repeater”; it knows how to listen to audio data coming from a source stream (e.g., the WinAmp DSP plug-in) and efficiently broadcast it to listening clients “out there.” If the person controlling the source stream tells the server that the stream is to be publicly accessible, the server will send updates to the public web directory at *shoutcast.com* or *icecast.org* listing the name and description of the station along with a linked IP address so that listeners can check in with a single click.

If you don’t tell the source streamer that the stream is meant for the public, your broadcasts will only be known to people who are specifically given the URL. Many people use this option to run a server at home that they listen to personally at work—a radio station for them and them alone.

Types of source streamers

There are two types of source streamers: The server may receive audio data from a *static file* source, or from a *re-encoding* source. Static sources simply broadcast MP3 files out over the network at the bitrate at which they’ve been encoded. Static streamers require a lot of bandwidth, but consume very little CPU or memory, as their task is very simple—just pass it on, brother. Re-encoding source streamers,

on the other hand, must decode existing files to raw data and then re-encode them to downsampled, modem-friendly quality levels on the fly. As a result, they consume much less bandwidth, but are much more taxing on the machine's available resources. They also require access to an MPEG codec to do their job, while static streamers do not.

There are two popularly used MP3 broadcast solutions available: SHOUTcast and icecast. SHOUTcast is developed by NullSoft,* the makers of WinAmp, and some of its functions are tightly integrated into WinAmp's. icecast is an open source alternative to SHOUTcast, and emerges from the Unix/Linux world. Fortunately, the two are nearly 100% compatible. You can send streams to an icecast server from WinAmp, or to the SHOUTcast server from shout or liveice, both of which are source streamers that come with icecast.

This willing compatibility between the two products is a refreshing change from the standards wars that usually boil around the introduction of new technologies. At this writing, icecast offers more security options than SHOUTcast, and uses less memory and CPU cycles than SHOUTcast (though this may change in the future). Both products are free (as in free beer), while only icecast is truly free (as in free speech and Open Source software). Both products' server modules run under most Unix-like operating systems and under Windows 95/98/NT. At this writing, only SHOUTcast offers a graphical user interface (though the SHOUTcast server is only quasi-GUI). icecast is completely command-line based, though this may change in the future. SHOUTcast is much more popular than icecast, and will receive the bulk of attention in the remainder of this chapter.

Working with SHOUTcast

In the SHOUTcast system, WinAmp itself is used as the source stream. WinAmp is used to organize and play music, while a WinAmp, Sonique, or other streaming output plug-in is used to send the stream off to an MPEG codec on the same machine for downsampling, and then on to the SHOUTcast server.



Do not confuse the term "SHOUTcast server" or "server" in this section with the web site at www.shoutcast.com. The former refers to the server module running on your own machine or a machine to which you've been granted explicit permission to broadcast from. The SHOUTcast web site merely serves as a public directory of currently broadcasting servers. shoutcast.com does not replicate your stream or offer bandwidth or storage space to users—it just points to stations which are currently "on the air."

* NullSoft was purchased by AOL in mid-1999.

The server receives downsampled information from the WinAmp plug-in and passes audio data out to the Internet. Optionally, the system will also cause your "station" to be displayed in SHOUTcast's public directory of available radio stations.

Configure the plug-in

To get started, head over to *shoutcast.com* and download the DSP plug-in for WinAmp.* Install the included .DLL into WinAmp's plug-ins directory, restart WinAmp, launch the Preferences panel, and you'll find a new entry in the DSP category labeled "SHOUTcast source for WinAmp." Click Configure, and you'll see a panel similar to the one in Figure 8-2.

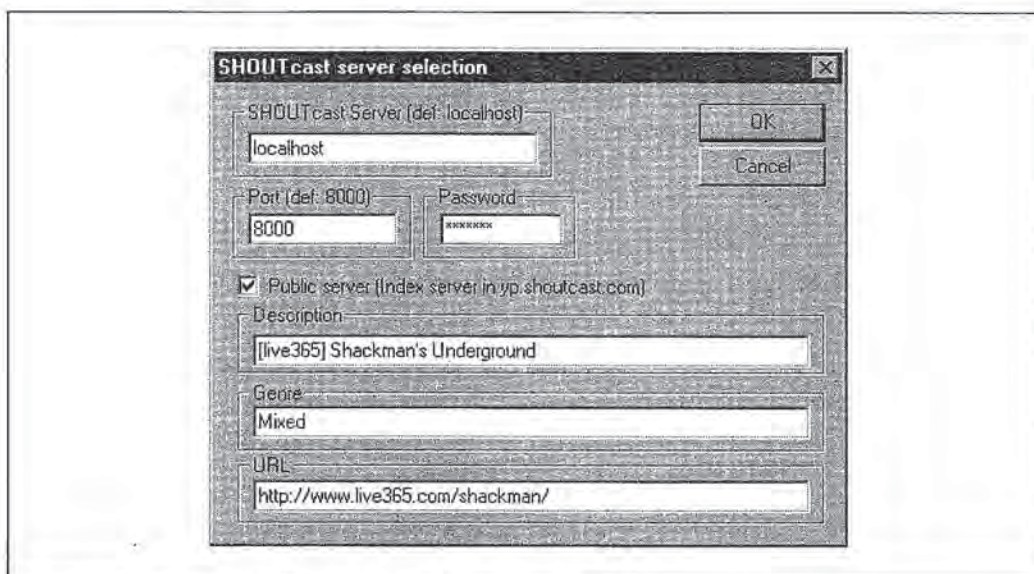


Figure 8-2. The SHOUTcast DSP plug-in for WinAmp

In the vast majority of cases, you'll want to leave the port address set to the default of 8000. You'll only need to change this if you're on a machine that's already using this port for another service (if unsure, ask your network administrator for advice).

The "Format" field represents the most critical choice you'll have to make as a station operator, as this is where you'll determine the balance between audio quality and the number of listeners your station can support. Because very few people have a fast enough connection to the Internet to listen to MP3 audio exactly as it's stored on your system, the SHOUTcast DSP *downsamples*: decreases the sample-rate and bitrate of the audio so it can be distributed effectively over POTS (plain old telephone service).

* WinAmp's DSP plug-in isn't the only source streamer available for use with SHOUTcast, but it's by far the most popular. Sonique and other players are also capable of sending audio streams to broadcast servers. However, we'll use WinAmp as an example in this section due to its broad popularity.

The configuration dialog for the DSP plug-in allows you to specify downsampling parameters, and here you'll have to do some calculations to take into account the speed of your Internet connection, and more importantly, the speed of your user's Internet connections. If you're broadcasting to a few known friends with high-speed connections, you'll probably want to set the ratio high, to give them maximum possible fidelity. If you're broadcasting to the general public, however, you'll probably want to assume that most of your users have a 33.6 kbps connection. Of course, as you probably know from experience, the maximum rated throughput of any given modem rarely represents the actual throughput after Internet overhead is taken into account. The safest bet for general streaming implementations is to select one of the 24 KB options in this dialog. That, however, means you're going to be broadcasting some very low fidelity music out into the world, which isn't a very happy prospect either. You'll need to choose your samplerate based on your perception of your target audience's available bandwidth. If you're reading this a couple of years from the time it was written (early 2000), you may have a very large audience of cable and DSL users, and may be able to afford a much higher samplerate.



It's important to distinguish between your own connection speed and that of your users, and to remember that your upstream bandwidth is going to be affected by Internet congestion. In addition, if you're using a 56 K modem, remember that the "56 K" label applies only to your downstream bandwidth. 56 K modem users rarely see better than 33.6 upstream bandwidth, so unless you've got a DSL or cable connection on the server machine, you'll probably want to limit your samplerate to 24 K even if you think your audience has a faster connection. If you're running the source streamer and the server software on separate machines, you'll need to take into account both the speed of your connection to the server machine and the speed of the server machine's outgoing connection. If you try to dish up more than can fit through the pipe from one end to the other, your users will experience drop-outs and lost signal. Be sure to do plenty of testing before marking your server as "public."

Install an appropriate codec

The DSP plug-in depends on the presence of an MP3 codec on your system running the source streamer. Because NullSoft can't provide this codec without paying huge amounts of money to Fraunhofer, they assume that the user already has an appropriate MPEG codec installed. Fortunately, users can freely download the codec from Microsoft, or purchase a higher quality codec from a third party. To see if you have the codec installed on your system, launch the Windows Control Panel and navigate to Multimedia → Devices. Look under the Audio Compression

Codecs section for an entry labeled "Fraunhofer IIS MPEG Layer-3 Codec." If you're using a recent version of Windows, or if you've installed the latest version of the Windows Media Player, you probably have this already. If you don't, you can install the codec that comes with the Microsoft NetShow tools package. You'll find a direct link to the latest version of this package at shoutcast.com's Downloads page.

Note, however, that the Microsoft-provided Fraunhofer codec is limited to streams of 56 kbps. While this should be plenty of resolution for almost all MP3 broadcast implementations, those with special high-bandwidth needs may install a high-end codec available from www.opticom.de.



If you want to dish up files exactly as they're currently encoded, you won't need to do any downsampling during the serving process. However, this either means that A) your files are encoded at normal bitrates and you don't mind limiting your audience to owners of very fast connections only, or B) you've already encoded the files you intend to broadcast at relatively low bitrates and/or frequencies.

Configure the server

Once you've selected the SHOUTcast DSP plug-in in WinAmp's plug-in preferences panel, everything played in WinAmp will be sent on to the SHOUTcast server as well as to your own speakers. The server, however, will not redistribute music to the masses until it's been launched and enabled. Launch the SHOUTcast server (probably at `c:\program files\shoutcast\sc_serv.exe`) and you'll see a panel with a simple reporting window, which will show you what the server is up to at any given moment, as shown in Figure 8-3. The main window simply reports the server's transactions. All configuration is accomplished by editing a text file.

The default configuration of the server should be fine for starters, and you can now test the system. Make sure something is playing in WinAmp, then go to another machine, launch a broadcast-capable MP3 player, and enter the IP address of the host machine along with the port number. On a common "Class C" network, you might enter something like:

`http://192.168.0.3:8000`

If you have a real IP address, you might enter something like:

`http://202.14.123.33:8000`

After a brief delay, downsampled music being played through WinAmp on the remote machine should start to play on the client.

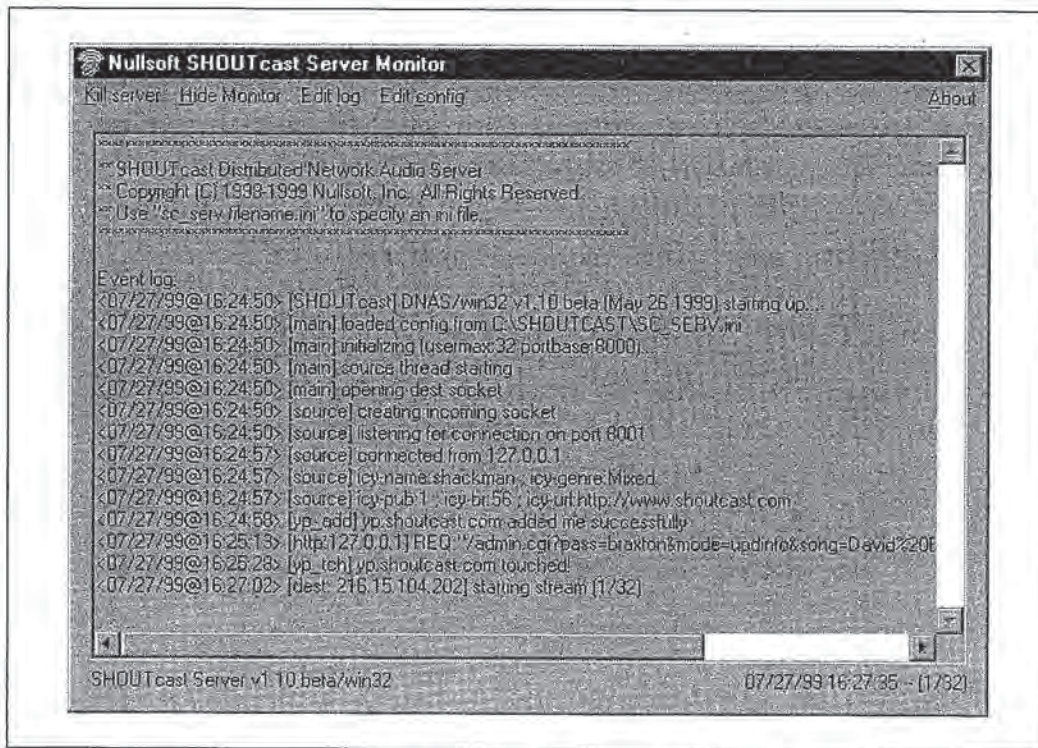


Figure 8-3. The SHOUTcast server offers little in the way of an interface



Most of these instructions assume you're working in Windows, but the SHOUTcast server is available for Linux and other Unix-like platforms as well. Configuration instructions are virtually identical for both operating systems. Linux users will need to change permissions on the server binary with `chmod u+x`, and to make sure the config file is readable by the user who will be running the server. Note that if the user is not root, he will not be able to use port addresses lower than 1024. Note also that only the server component is available to Linux users, not the WinAmp source stream plugin. Under *nix, substitute `sc_serv.conf` wherever it says `sc_serv.ini` in the text below.

Additional server options

Beyond the defaults, the server software offers a ton of additional options. All configuration of the SHOUTcast server is (at this writing) accomplished by editing the text file `sc_serv.ini`, which you'll find in the SHOUTcast installation directory. The file can also be accessed via the server's Edit Config pull-down menu. The config file is very heavily commented (commented lines begin with a semicolon) and should be relatively self-explanatory. Since `sc_serv.ini` is read into the server's

memory when it's launched, you must kill and restart the SHOUTcast server after making any configuration changes. Note that many of the options in the config file are simple on/off switches, where "1" represents "on" and "0" represents "off." Here are brief descriptions of some of the most useful options:

Logging

The four logging options are quite intuitive, and let you specify the location and name of the file to which a traffic log is written, specify whether traffic is written to the screen in addition to the log file, and control the behavior of the "history" log, which can be parsed by other programs to create HTML graphs of ongoing traffic patterns. This can be set to */dev/null* on Linux machines if you want it disabled entirely, or *none* for Windows machines. The number of seconds between writes to the history log file can be specified on the *HistoryLogTime=* line.

Network configuration

Options set here determine the way your station interacts with the network. For example, you can change the port address your server is running on with the *PortBase=* line. The *SrcIP=ANY* line gives you control over who is allowed to broadcast MP3 streams through your server. By default, this is set to *ANY*, so that anybody running the DSP plug-in will be able to broadcast via your server. If you set this to your "loopback" IP of 127.0.0.1, only you will be allowed to use the server. If you want to limit use to people on your own network, you might use something like 192.168.*.*. Remember, however, that source streamers must still connect with the appropriate password, so you aren't completely exposed to the world.

If your server machine is bound to multiple IP addresses, you can specify which address the SHOUTcast server is on by editing the *DestIP=ANY* line. If your machine is running a DNS server, you can see the actual domain and machine names of the clients connecting to your server by editing the *NameLookups=0* line, though this may impact performance if your nameserver is slow.

Relays

You can cause your server to function as a "relay" server by uncommenting and editing the *RelayPort* and *RelayServer* lines (Figure 8-4). This allows you to re-broadcast incoming streams to another server for final distribution to "official" servers. Relay servers are particularly important for situations where you want to serve more listeners than the bandwidth available to a single server can accommodate.

Server configuration

Perhaps most important, this section lets you control specific parameters of your server's behavior. The *MaxUser=32* line lets you raise or lower the

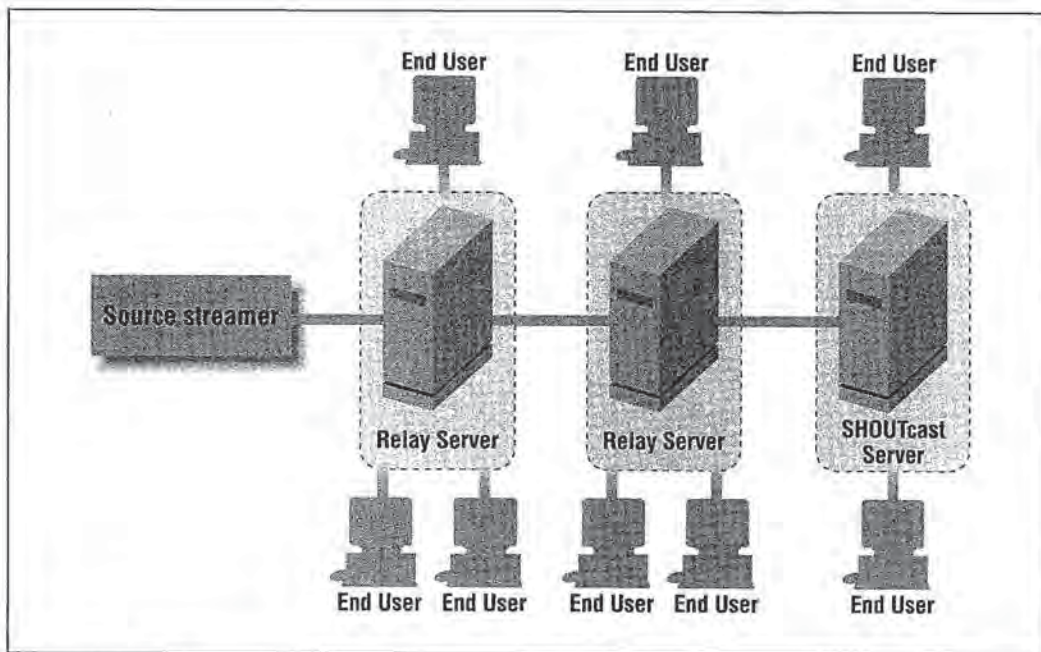


Figure 8-4. SHOUTcast/icecast "Relay" functions let administrators perform load-balancing across multiple machines

maximum number of clients that can be served by your station simultaneously. You'll need to tailor this line carefully to match the actual bandwidth available to your machine. To do this, multiply your available bandwidth by .9 in order to account for Internet congestion and overhead, then divide by the bitrate at which you intend to serve. For example, if you have a 384 kbps upstream DSL connection and you want to serve MP3 audio at 24 kbps: $384 \times .9 / 24 = 14.4$. Thus, you should probably set the *MaxUsers* line to 14, or decrease the quality of the audio you're serving. Since the default is set to 32, it's important to edit this line if you're on anything less than a very fast connection. The version of the server available at this writing was capable of serving a maximum of 1,024 simultaneous users—pretty amazing, considering that 60 simultaneous users can saturate a typical T1 line!



If you're serving up a streaming MP3 server on a shared corporate or organizational network, it's very important that you consult with your sysadmin before beginning. Set a high number of *MaxUsers* without permission and it won't be long before you're hogging all the bandwidth your organization has at its disposal. Since the sysadmin's very existence is predicated on keeping all of his users happy, such a stunt is likely to land you in a Siberian cell block with nothing to eat but gruel and water. Be courteous, think ahead, and get permission before you begin.

The default password built into both the DSP plug-in and the SHOUTcast server is "changeme," and it's probably a very good idea to change this so that rogue users can't piggyback on your server's connection without your permission. Remember that the password in both the server config file and the DSP plug-in must match, so don't forget to change this in both places.

AutoDumpUsers and *AutoDumpSourceTime* can be used to specify whether your users will be disconnected automatically if the source stream becomes disconnected for some reason, and how long the server should wait before disconnecting those users.

If you'd like your users to hear an "intro" file as soon as they connect, before being transferred to the audio stream in progress, encode it at the same sample-rate and number of channels as the rest of your stream, then specify the file's name on the *IntroFile=* line. You can set a bitrate for the file in the filename itself, and then reference it on this line with the *%d* flag. For example, rename your intro file to *intro64.mp3*, then specify *IntroFile=c:\intro%d.mp3* to cause the file to be broadcast at 64 kbps. Similarly, if you haven't enabled *AutoDumpUsers*, you can provide an MP3 file that will be played automatically, over and over again, should the source stream disconnect. The same options for the intro file apply to the backup file.

You can also control the title that appears to users as they listen to your station. By default, this will simply be provided by the DSP plug-in, and will be read out of the currently playing track's ID3 tags. If you'd rather your users see a string like "Shackman's Underground," uncomment the *TitleFormat=* and append your preferred string. For extra credit, you can combine your station name with the title being sent by the source stream by using the *%s* flag; for example, if you use:

```
TitleFormat=Shackman's Underground: %s
```

and you're currently playing Billy Bang's "Valve #10," your users will see:

```
Shackman's Underground: Valve #10
```

Users can also be sent a URL, which makes it easy for them to visit your site if they decide they dig the tracks you're laying down and want to learn more. Because the *URLFormat* line can also accept the *%s* flag as an argument, you can allow users to arrive at a specific page on your web site via a custom CGI script. For example, if you use:

```
URLFormat=http://www.server.com/redirect.cgi?url=%s
```

then the name of the currently playing track will be passed to a CGI script running on the web server, and you can dish up additional information on that track. This presents a fantastic opportunity for artists and bands to encourage users to learn more about them as they listen, though you'll need some pre-existing CGI (probably perl) experience to pull this off.

All of this title, URL, and station ID information is known as “meta-data,” and is constantly being sent to listening clients. Because meta-data will change as the source streamer moves from one track to the next, you can control how often the meta-data is sent as well. This interval is established not in seconds, but in bytes, and you can control how often meta-data is refreshed by tweaking the *MetaInterval*= line, though SHOUTcast recommends sticking with this default. If you're running a special broadcast consisting of zillions of very short tracks, you might want to make this shorter, but there's probably no good reason to make it longer, as the amount of bandwidth it consumes is so small as to be almost unmeasurable.

Extended logging

In addition to the standard logging capabilities described earlier, the SHOUTcast server is also capable of generating some pretty cool HTML-based logs and up-to-date reports, which you can access from anywhere thanks to the fact that SHOUTcast actually includes a built-in miniature HTTP server. So if you're running a SHOUTcast server from your home machine and want to see how popular it's been today from your desk at work, you can type in (substituting the actual IP address of the server, of course):

```
http://217.77.123.3:8000/current.html
```

In order to generate HTML reports, you'll need to create an HTML template file defining the look and feel of the report, then stuff it full of the allowable SHOUTcast logging keywords, in uppercase letters and embedded in HTML comments, as shown in Table 8-1. The server will treat each commented keyword like a variable, substituting in actual data from the history log.

Table 8-1. Special Remote Variables

Keyword	Keyword Function
<!-- LISTENERS -->	Number of active listeners
<!-- MAXLISTENERS -->	Number of maximum listeners configured in the server
<!-- LISTENERTABLE -->	Listeners listed in table format
<!-- GENRE -->	Genre string of currently playing track (from ID3 tag)
<!-- DESCRIPTION -->	The current description string
<!-- URL -->	The current URL string
<!-- HITS -->	Total number of hits since server was started
<!-- CURRENTSONG -->	Name of the current song

Once you've configured your template, you'll need to tell the server where to find it and how to treat it. The template is considered an input file, so edit the *CurrentLogIn*= to point to the template file. The generated HTML document is your output, to edit the *CurrentLogOut*= line to specify a path to the generated

report, e.g., *CurrentLogOut=C:\current.html*. Finally, you can specify how often the report should be updated with the *CurrentLogTime=* line.



If you find that you need to run the server under one configuration one day and another configuration the next, you may want to make multiple copies of *sc_serv.ini*. You can then launch the server with a particular configuration from the DOS shell by typing something like:

```
sc_serv.exe sc_serv2.ini
```

Remote administration

Because the SHOUTcast server has (limited) built-in HTTP services, it's also capable of being administered remotely, i.e., from another machine anywhere on the Internet (Figure 8-5). Open any web browser and enter *http://ip_address:port* (substituting the real IP and port addresses) and a web page will be sent back to the browser listing the name of the station, the musical genre being played (as identified in the DSP plug-in), the bitrate being dished up, and the name of the currently playing track. Enter the correct password, click the Admin button, and you'll be passed to an administration page reporting full details on the current server configuration. While you can't change all server options remotely (or at least this was not possible at this writing), you can disconnect from the stream source with a single click, thereby disconnecting all currently connected users. Unfortunately, it is not possible to reconnect the source remotely, since this requires access to the DSP plug-in interface, not just to the server, so be careful!

In addition, you can view the entire log file that has been generated since the server was started, or just view information on the last few connections, by clicking the "tail logfile" link. Click the Usage Graph link to see an ASCII representation of the number of connections that have been established on a scale of time. You can also change the number of *MaxUsers* if necessary, but remember that doing so does not actually change the *sc_serv.ini* file, so you won't be making a permanent change here.



You can jump directly to the administration page without going through the "front door" by including your password in the URL. For example:

```
http://127.0.0.1:8000/admin.cgi?pass=zoomer23
```

might get you into administrative mode on your own machine. If you're not worried too much about people hijacking your machine, you can bookmark this URL for easy access in the future.

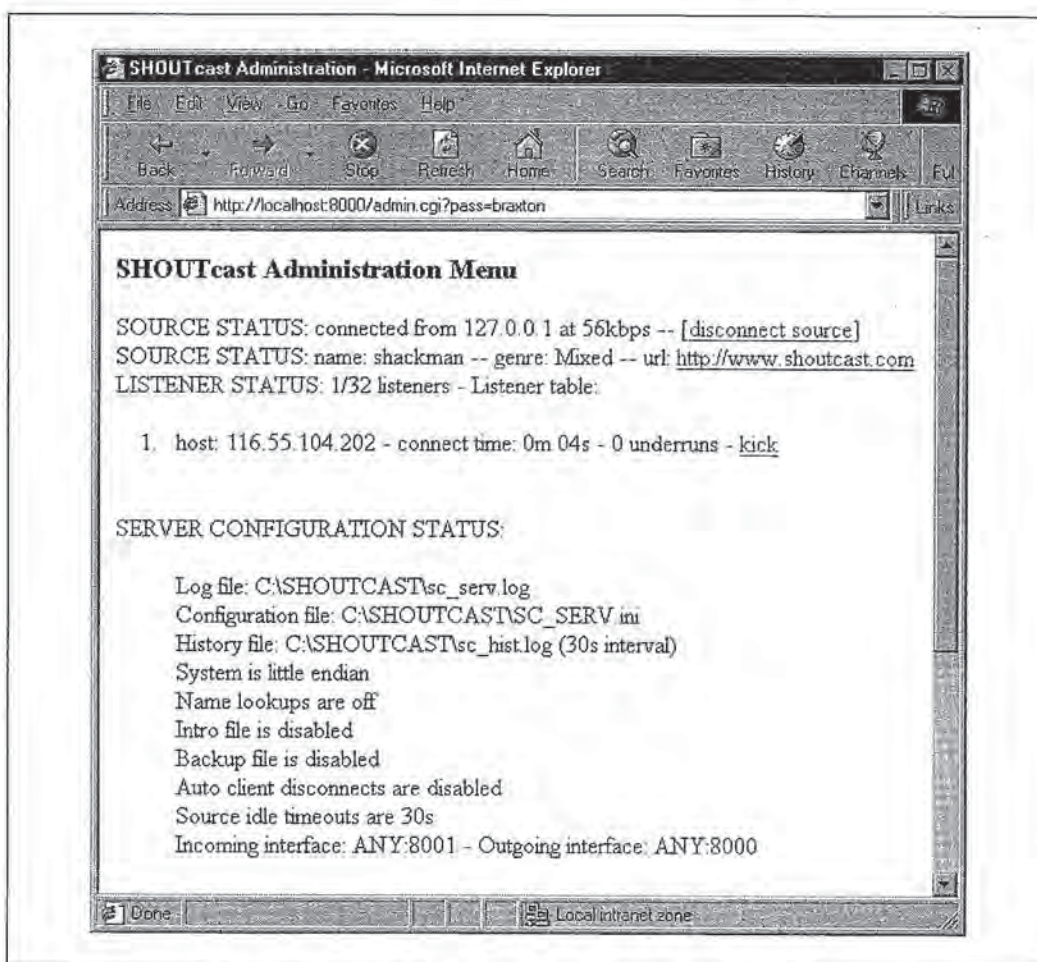


Figure 8-5. The SHOUTcast server can be administered from any web browser on the Internet in a limited fashion, as long as you can supply the correct password

Serving live streams, microphone input, and other formats

The SHOUTcast server is not limited to the distribution of existing MP3 files, but can handle any format that can be processed internally by WinAmp. Thus, streams from other sources that register in WinAmp's Input plug-ins can be encoded to low-samplerate MP3 on the fly, and served up just as you would serve up files. If your band is playing at a local club that has a decent Internet connection, you can take a line feed from the mixing board, run it into the input jack of your sound card, and hand your URL to the world so they can listen in from the comfort of their living rooms. However, there's a catch: The CD audio, line input, and MIDI input plug-ins do not use WinAmp's internal processor; WinAmp simply acts as a pass-through for these formats. To get around this limitation, you can install a third-party plug-in to replace the default plug-in. These alternatives will force WinAmp to handle the audio stream internally; it can pass the signal on to the SHOUTcast server. Look for the cd-da plug-in if you want to stream CD audio.

If you want to add a little vocal commentary to the mix as your station is broadcasting, tap Ctrl+L in WinAmp's playlist window and type *linerec://*. When you select the *linerec://* entry in the playlist, your station will be broadcasting anything coming through the sound card's input jack, including live input from your microphone. When you're done waxing poetic about your favorite artist, click on another song title in the playlist. Be careful though—if you leave the *linerec://* entry in place and the playlist cycles around to its entry again, it's all too easy to inadvertently broadcast what you might rather the world didn't hear.

Resource consumption

Since the SHOUTcast server doesn't have to hit the disk to retrieve MP3 data, or to run the MP3 decoding process, you'll actually find that running the SHOUTcast server consumes fewer machine resources than running an MP3 player locally. However, that's not the whole story—serving MP3 files is very taxing on network resources. Not only will you gobble up large amounts of outgoing bandwidth, but you'll be chugging through a ton of packets on the server machine as well. Linux users can run *tcdump* while the server is in use, and take a look at the output when you quit the utility. At 56 kbps, you'll probably find that you're processing around 20 packets per second, which can cause bottlenecks at around 10 users, depending on your machine and its particular configuration. This is where “repeaters” (see the “Relays” section of the server config file) come in handy—rather than trying to handle everything on one machine, run the server on different machines on the same LAN to distribute the load. This should only be necessary if your station is becoming very popular and you don't want to dedicate a machine expressly to the task of running your station.

Using publicly available servers

So SHOUTcast is all well and good for people with fast connections to the Internet, but what if you just have a dinky modem and no access to a machine on a fat pipe running the SHOUTcast server component? Does that mean you're stuck serving very low quality audio to just one or two people at a time? Fortunately, no. The advent of the WinAmp DSP plug-in signaled the appearance of sites such as *www.live365.com*, which provide server space and beefy bandwidth to those in need, absolutely free. In fact, live365 is so generous with their bandwidth that they'll actually let you stream your content to more people than you can probably reach with a typical DSL connection, without chewing up the bandwidth you might like to use for other purposes, such as running a standard web server. At this writing, live365 let anyone and their sister dish up 100 simultaneous streams of 56K audio, absolutely free, bringing the possibility of running your own station much closer to home for the average Joe. By the time you read this, it's a good bet live365 will have plenty of competitors with similarly generous offers.

Saving MP3 Streams

Since MP3 streams are typically broadcast in low-bitrate, heavily downsampled formats, it's not too likely that users will want to save them to disk for posterity. However, there are those odd times when it can be desirable and/or convenient to trap a broadcast to disk—if you want to save a news broadcast in the middle of the night and listen to it in the morning, for example. That's really no different from videotaping a TV show to watch later, and doesn't necessarily run afoul of legal issues. Since the capacity to do so is conducive to piracy, however, most commercial MP3 players that support streaming make this impossible by design, or at least make the process difficult (if you try to use WinAmp's built-in DiskWriter plug-in for this purpose, you'll get nothing but a copyright violation alert box from WinAmp. Older versions of WinAmp, however, do support saving MP3 streams to disk). There are a few select utilities dedicated to the task of pulling down streamed broadcasts to disk; check out X-FileGet, available at www.2bsys.com or search the Internet for tools like TotalRecorder or AudioJacker.



As tantalizing as the prospect of serving up 100 simultaneous streams may sound, keep in mind that very few stations actually attract this many listeners. Take a quick surf around the SHOUTcast or icecast directory pages and you'll see that only the very rare station attracts more than 10 simultaneous users, and these are stations that have risen to the top of the popularity heap with constant dedication to excellence, not just casual users broadcasting whenever the mood strikes. Trouble is, there are just so many hundreds of thousands of stations to choose from that the competition for listeners is immense.

Working with public servers is virtually the same as working with the SHOUTcast server, as described earlier in this chapter. You'll still need to install the WinAmp DSP plug-in, still need to make sure you've got a codec installed, and still need to create and manage a WinAmp playlist. Instead of connecting the source stream to the SHOUTcast server, you'll send your stream on to a unique set of IP and port addresses supplied to you by the public server. Detailed documentation is available at live365, should you run into any configuration issues.



While the MP3 broadcast tools covered in this chapter are by far the most common, they're not the only ones available. Check out Xing's StreamWorks and MP3Live! server and encoder combination (www.xingtech.com/mp3/streamworks/) for an interesting alternative.

Working with icecast

Those who prefer to stay on the open source side of the fence will be happy to learn that a functional equivalent to SHOUTcast exists, in the form of a server solution called icecast. In fact, there are very few functional differences between SHOUTcast and icecast, and icecast is designed to be completely compatible with SHOUTcast. You can broadcast source streams from WinAmp under Windows to another machine running the icecast server, and vice versa. You can broadcast from an icecast server to the public SHOUTcast directory, and vice versa. And since icecast is truly Open Source, it runs on more platforms than SHOUTcast (at least it did at this writing), including Linux, Windows 95/98/NT, BeOS, Solaris, and likely more by the time you read this. The icecast server offers more configuration options than SHOUTcast, and its development cycle is more rapid than that of its NullSoft cousin.



Unless your operating system comes bundled with a fully licensed MP3 codec, you'll need to locate and install one of your own if you want to stay legal (details can be found in the sections below). Both Microsoft and Be have paid a king's ransom to Fraunhofer and THOMSON for the right to include the MP3 codec in the OS (though Microsoft's is limited to 56 kbps).

By default, icecast will serve files only at the bitrate in which they're encoded (i.e., it's a static streamer). That means that if you want to accommodate listeners who log in via modem, you'll need to encode your files at very low bitrates to begin with. icecast ships with two source streamers: shout, for streaming static files, and liveice, for interacting with third-party codecs and for mixing multiple source streams, including microphone input.

Whether you decide to source your station from shout or liveice, start by firing up the icecast server. You should find it in the `bin` subdirectory of the icecast installation. From that directory, type:

```
./icecast
```

Don't follow the icecast command with an `&` for now, as you'll want to monitor its output while you get everything up and running. Later on you can run it in the background, and have the output dumped to a log file. By default, the log file will be created as *icecast.log* in the *icecast* directory.

Like SHOUTcast, icecast also has a text-based configuration file that you can tweak to establish additional parameters or to set up your station for unique situations. While the defaults will get you up and running quickly, do look through *icecast.conf*, which you'll find in the *etc* subdirectory of the icecast installation (you can

optionally toss this into */etc* along with all of your other *.conf* files). At the very least, you should establish a password for connecting source streams after you verify that everything is running properly with the defaults.

Running shout

With icecast running, you can now send a source stream to it with WinAmp from another machine on the network, or use the *shout* command included with icecast. *shout* takes a ton of optional parameters, and can also be made to run with a configuration file encapsulating your most-used options. For starters, try:

```
shout localhost -P letmein -a -p ~/my_playlist.m3u -l -g funk -n "Mr. Toad's Wild Ride"
```

In this case, *shout* would send its stream to the server "localhost" (i.e., your own machine; you can use any IP address here), would use the password "letmein," would play the playlist stored at */home/my_playlist.m3u*, would loop forever (return to the beginning of the playlist when it reached the end), would list its genre as "funk," and would broadcast its name as "Mr. Toad's Wild Ride." The *-a* flag is used to enable *shout*'s automatic transfer bitrate correction feature. As it plays, it monitors its own transfer speed. If it finds itself going too fast or too slow, it will automatically correct for these variances.

To save yourself some typing, you can embed all of these options into a configuration file and simply launch *shout* with the *-C* flag followed by the path to the file. Take a look at *README.shout* for additional *shout* options and parameters.



You'll notice that *shout* offers a *-b* flag to specify the bitrate. Since *shout* is a static streamer, it cannot actually change the bitrate of the files it sends. This option merely lets you broadcast the bitrate of your files for the benefit of users. This bitrate will appear in the readout window of most MP3 players, but it's up to you to make sure it accurately reflects the actual bitrate of your files!

Running liveice

If you need to downsample your files before sending them out into the ether, you'll want to source your stream through *liveice* rather than *shout*. Since *liveice* is a re-encoding streamer, it requires access to a codec. *liveice* can either use a "raw" codec stored in a central location in the OS as one does with Windows and BeOS, or simply utilize the decoding and encoding capabilities built into one of your existing MP3 players. No encoder is distributed with icecast or *liveice*; you'll need to obtain one elsewhere on the Internet. Use a search engine to track down encoders listed in *liveice.cfg*. By default, *liveice* looks for the existence of *mpg123*

(see Chapter 4) somewhere in your path for its decoding needs. Supported encoders include l3enc, Xing, LAME, etc. See icecast's documentation for details: the options are likely to grow in the future. Utilizing the Xing or LAME encoders may be well worth your while; both are more efficient than mpg123 (Xing in particular). For now, we'll stick with the defaults and assume you're using mpg123.

As you may recall, mpg123 can take certain command-line flags that force it to downsample the stream to a specific frequency (this capability should also be present in LAME by the time you read this). liveice simply calls mpg123 with these parameters and returns the resulting downsampled stream to the icecast server.

Running liveice in its default configuration is just as easy as running shout. Copy your playlist file and *liveice.cfg* (from the icecast distribution) to the liveice installation directory, make sure icecast is running, and type */liveice*. You should now be able to open a streaming MP3 client such as Xmmms on your own machine and open the URL *http://localhost:8000*. Naturally, an actual IP address can be substituted for "localhost" for network usage. Bitrate, samplerate, broadcast name, associated URL, and many other configuration options can be set by editing *liveice.cfg*.

liveice is capable of broadcasting multiple simultaneous streams, so you can, for instance, send high, medium, and low-bandwidth streams to various servers on fatter or thinner pipes all at once. Up to eight streams can be dished up simultaneously (bandwidth on the source streamer machine permitting, of course). After establishing the main settings in the top of the config file, look for:

```
ENCODER_STREAM_SET 0
```

section of the file and use *SAMPLERATE*, *BITRATE*, and other options to tweak the parameters for the first stream being sent. Move on to *SET 1* and do the same, and so on.

At the time of this writing, liveice's mixing capabilities were somewhat primitive (to say the least). The command line really isn't the ideal way to control live audio mixing, and a GUI interface was rumored to be in development. The command-line version of the mixer comes into play when you launch liveice with the *-M* flag. The playlist begins to play as normal, and two additional channels are opened. Audio can then be streamed into these channels from pipes set up elsewhere in the operating system, or from microphone. The keyboard controls in Table 8-2 are then in effect.

Table 8-2. *liveice* Keyboard Shortcuts

Action	Channel 1 key	Channel 2 key
Select next track on channel	1	A
Select prev track on channel	Q	Z
Start/Stop channel	2	S

Table 8-2. *liveice Keyboard Shortcuts (continued)*

Action	Channel 1 key	Channel 2 key
Reset channel	W	X
Increase volume on channel	3	D
Decrease volume on channel	E	C
Increase speed on channel	4	F
Decrease speed on channel	R	V
Sticky mode On/Random/Off	5	G
Preview channel	T	B
Random Track	U	M



If you're handy with bash or Perl scripting, you can automate an Internet broadcast almost entirely, setting up tracks to be mixed in advance. For an excellent Perl script built to handle much of this automation, check out IceDJ at www.remixradio.com/icedj/.

An Interview with MP3.com's "High Geek"

Sander van Zoest is the "High Geek" and Senior Web Engineer at MP3.com, one of the Web's most-trafficked sites. He also happens to have been among MP3.com's first employees. The author interviewed van Zoest via email.

Scot Hacker: How long have you been a part of MP3.com? When did it all start, and what was the original objective? Where is MP3.com based?

Sander van Zoest: MP3.com is based in San Diego, California. It was launched out of The Z Company, which was founded by our CEO, Michael Robertson. The Z Company was a small web shop that ran many web sites, such as Filez, the Net's largest and fastest file search engine, and Websitez, a domain name search engine. Greg Flores, our current Vice Present of Sales, told Michael about MP3, and in November 1997 they opened MP3.com.

The objective was to provide a one-stop site for the MP3 movement. It provided ground-breaking news and editorials by Michael Robertson, which can now be found on our site under the heading "Michael's Minute," and hardware and software sections to get new users started. In the summer of 1998 we started accepting MP3 music from musicians and artists around the world, to provide them with a revolutionary way of distributing their music.

What About Multicasting?

Internet veterans may have heard of a remarkable technology known as "IP Multicasting," which provides for more efficient distribution of high-bandwidth media than do traditional serving mechanisms. With traditional servers, a copy of the requested media is sent over the Internet to each client who requests it, even if they're sitting in side-by-side cubicles. The result is that for most of the journey from server to client, twice as much data is sent than is really necessary. The idea of multicasting is that the network and broadcast applications are given a new layer of intelligence, so they can determine whether data being sent to geographically proximate locations would be redundant. If a server is in France and 14 Americans are requesting an MP3 stream from it, the server only needs to send 1 stream to the U.S. Once the paths to the clients become divergent, they're split off to the individual users.

Clearly, IP multicasting is far more respectful of the Internet's bandwidth than standard serving techniques. In addition, multicasting can take a huge bandwidth burden off the server itself. Potentially, multicasting technology could make it feasible to run an MP3 broadcast server over a standard modem, which would clearly be a huge breakthrough for users who lack access to a high-bandwidth server.

The catch? Multicasting requires that all points in the chain be upgraded to handle the technology. All of the routers between the two points must be multicast aware, as does the TCP/IP stack in use and the broadcast application software. Fortunately, IP multicasting technology is slowly but surely shaping up at many points in the chain, and is becoming a more realistic possibility for broad deployment every day (it can already be used in some scenarios). Take a look at www.live.com and www.ipmulticast.com for more information. To learn more about the region of the Internet currently "roped off" for use in multicasting scenarios (commonly referred to as "the MBone"), see www.live.com/mbone/.

LIVE.COM offers a program called liveCaster, which is geared specifically toward serving up MP3 broadcasts over IP multicast. The site offers detailed instructions both on installing the liveCaster server software and on tuning into existing liveCaster streams with popular MP3 players.

I started working part time with Z Company in June of 1998, around the time of our first annual MP3 Summit, which was an attempt to gather all the influential people of the MP3 movement together under one roof. The event was held at the University of California, San Diego campus, and was when I realized how huge the MP3 movement was going to be. My good friend Josh Beck introduced me to Michael, and I knew I had to find a way to join their quest to revolutionize the music industry. In November of 1998, I joined Z Company as a full time employee.

SH: I can hardly imagine a site with more intense bandwidth requirements than MP3.com; you must be pumping out absolutely massive amounts of data. Tell me a bit about the server and connectivity infrastructure at MP3.com. How do you scale the system as you grow?

SVZ: We are really dedicated to providing every visitor with the best experience and we work some insane hours to accomplish this. Our servers are modified versions of the Apache Web Server and run on a variety of Unix machines ranging from Sun's Solaris, to Linux, to FreeBSD. Most of the systems are Linux machines, since most of our staff is familiar with the operating system and it performs quite well in our environment.

To provide constant uptime, we use some of our own load-balancing techniques and third-party network products.

SH: What kind of bandwidth are we talking about here? Do you have a hole in the back wall with a dozen T3s running out to the world or something?

SVZ: Our bandwidth requirements change by the day. Just to give you an idea, we host our site out of several different data centers to provide 24-hour access. We're working on bringing the content closer to you by sending you to the closest machines, network-wise, that we have. It's a big operation, especially since we do not control the content directly—the artists do, and we have to make sure they have the tools and infrastructure to keep their own sites up to date.

SH: What about the CD recording back-end for creating the DAM CDs? Do you have a whole CD factory on site?

SVZ: Yes, we have a just-in-time production facility that creates the DAM CDs and CD covers on the fly for each order. Each CD comes with the MP3s, Red Book audio tracks (for playing in your stereo system), and a multimedia interface with information about the artist and links to their web site.

SH: As astute users have observed, pointing a browser at mp4.com currently redirects back to mp3.com. Michael has stated several times that there's nothing sacred about MP3 itself, and that when/if the next big format comes along, the company will be ready for it. Do you do much internal testing of other file formats? Do you personally have a favorite digital music codec or file format, or is MP3 "it" for you?

SVZ: We are aware of the advantages and disadvantages of the MP3 format, but we plan to follow whatever standard the consumers want. Ultimately, it is the consumers who decide what format they would like to use. Without consumers there is no market, and without a market the artists will not be heard.

I, personally, do not really have a preferred format, though I do enjoy the freedom that MP3 provides. The fact that the MP3 specification is available and is a public standard, unlike formats such as MS Audio, Liquid Audio, or Real Audio,

really opens the doors to allow anyone to use the format in the way they see fit. The patents held by the Fraunhofer Institute and THOMSON Multimedia do impose some limitations of use, but not nearly as much as the closed, proprietary formats. MP3 allows for a wide range of bitrates, so you can specify the amount of disk space relative to the quality you desire. You can store high-quality audio for use in home theater surround sound systems, or take it all the way down to AM radio quality for listening to music over your cell phone. It's all possible with the MP3 format; it just depends on how much disk space and bandwidth you have available.

SH: We're just now beginning to see the very first "stars" being born out of this new format, rising through the democratic model of MP3.com and hitting it big without necessarily ever having gone through the traditional rigamarole of touring, promoting, signing, and pressing. Do you see this new phenomenon as an exception to the rule, or do we have a new star-making process on our hands?

SVZ: It really depends on what your goals are as an artist and what you would consider "a star." The beauty of the Internet and the MP3 movement is that it provides the artist a way to market to their fans that really are into the music. It provides fans a way to really get in touch with the artists. It provides much more loyal fans and better communication between the artists and fans.

How many times have you bought a CD that you had heard so many times on the radio that you decided that you really liked the artist, only to find after a few listens that they really had just that one good song? It can be really hard to find the music you're really into via the traditional music industry. Via the MP3 movement, you'll be able to really get to know the artist, and have a much wider variety of music to choose from. If you're into African death metal bagpipes, chances are there are artists that can fill your need.

So, getting back to "stars." I personally think there will be a lot more and a much larger variety of stars coming out of this movement. You now have the ability to become "a local hero"—a star for a more specific community. I think this will be much more rewarding for the artist as well. Sure you're going to find artists such as "Red Delicious" that will appeal to the mainstream audience, but this new movement also provides others with the ability to be noticed within their own communities.

SH: What are some of the biggest obstacles you've faced from a technical perspective?

SVZ: Wow, this is a tough question. Probably the biggest challenge we face continually is in making sure that when artists change their sites on the back end, their changes appear live on the site as quickly as possible. While this might seem like a trivial task, the moment you add millions of visitors the balance between

dynamic content and performance becomes an interesting but difficult task. And when you start dealing with more MP3 uploads than you can listen to in a day, you need to move the data around efficiently and as fast as possible.

Different interpretations of the MPEG Layer III Standard (ISO/IEC 11172-3:1993) also have been an obstacle. Because the MP3 spec is vague on precise implementation details, and because few developers actually read the full specification anyway, different encoders and decoders do things in slightly different ways. Since we've got artists uploading tracks created with the full range of available tools, we need to do things like determining whether or not a file is a valid MP3 file, how to handle free-form (no bitrate specified) files, and evaluate how players deal with bit reservoir variable bit rate (VBR) files—different encoders handle VBR MP3s differently. The objective was to have all MP3 files on our site usable by the majority of players, which was quite a task.

Another thing that's haunted us are the different ways various browsers, platforms, and players handled MIME types. We've also done quite a bit of troubleshooting with MP3 files to uncover bugs in encoders/players, and reported our findings to the appropriate authors.

SH: Are these things becoming a bigger problem as time goes on and the tools proliferate, or less of a problem as developers standardize and start to convene around a few standardized implementations?

SVZ: If all we did was fix bugs this would definitely become less of a problem, but with the demand for additional features such as embedding meta-data technologies such as ID3 and integration into every day appliances, I can see how this problem can get out of hand quickly. Open standards organizations and communication between developers and manufacturers will be the key.

SH: I've noticed that MP3.com has done a bit of work behind the scenes to prevent users from downloading files in big batches, with wget scripts for instance. What's the rationale behind that? Do you think you'll ever open up straight FTP access, or are you purposely making sure users stay glued to the site?

SVZ: Our business is about data and content. It is what we do best! We prevent people from doing mass downloads for the protection of both the artists and ourselves. If we allow people straight access to our content without hitting up our web site, someone with enough bandwidth and disk space could possibly bring up a competing site with illegal content (they most likely would not have permission from the artist and/or copyright owners to do so). Of course, we also need the ability to display banner advertising to cover our costs. And we use HTTP cookies to track downloads and user customized content. Not all download utilities fully support HTTP/1.1 or cookies, which can be the cause of the problems.

So there are a lot of reasons why we limit batch downloads and do not open straight access to our content.

SH: What are your impressions of the many home/car/portable MP3 playback devices beginning to appear on the market and in the DIY scene? Do you think store shelves will start to fill up with these in the first decade of the millenium? Do you own any dedicated MP3 hardware yourself?

SVZ: I really think this is a great start towards better ways of managing your music and your access to it. Just imagine being able to carry your entire music collection with you wherever you go, and having it available to you any time of the day no matter where you are. I think there's a lot of potential for digital music devices, and I have heard rumors of gaming systems using MP3 and next-generation phones having built-in MP3 players. It is exciting and has the potential to be very popular with consumers. I myself have a Diamond Rio player and hope to get an MP3 car player in the future.

SH: Do you expect any of the copyright protection mechanisms built into MP3 hardware to adversely affect MP3.com's business, which is built on the premise that a lot of MP3 music is willfully and openly distributed by the artist?

SVZ: I do not think it will affect MP3.com's business. The artists should be able to determine the fate of their works and decide how they want to distribute their music. If there is no content then the hardware becomes useless. At the same time consumers are used to being able to use their music anywhere they see fit. They can play the songs in the car, in their stereo and even make a mixed tape and share it with their friends. It exposes music to others, who may become the artist's next fans. If the copyright protection mechanisms restrict this behavior, I am sure this will upset them as well.

SH: Would you say there's currently a "typical" MP3.com visitor, or does the site appeal to a broad cross-section of people? Do you see your demographic changing in near future?

SVZ: There doesn't seem to be a "typical" MP3 visitor, other than the fact that they most likely have an Internet connection, a computer, and are interested in music. :) Our most popular genre of music is electronic, but other genres such as rock, alternative, and hip-hop are catching up fast. In the early days we had a lot of early adopters visiting our site, but this has changed.

SH: Do you get many "Aunt Margaret" visitors? That is, do you see many visitors who are just beginning to learn about computers, or are most of your customers relatively computer-savvy?

SVZ: Yes, we get a decent amount. We have been working hard on making it easier for the computer novice by making our help section easy to help people get started. People had a lot of trouble setting up their browsers and installing MP3

players with control over the appropriate MIME types and extensions. So we have been working actively with player developers and internally on making this as simple as possible.

SH: You must get some very wild submissions from time to time. Are there any artists that have just been too “out there” for you to host? Do you ever reject submissions based on adult or violent content? Does MP3.com ever make aesthetic judgements? (e.g., if something was just so bad it was unlistenable by any sane human, would you still run it?). Any funny stories relating to MP3.com submissions?

SVZ: We have had just about everything on our site, from the latest Tom Petty tracks to comedian Tom Green, from hardcore noise to stuff that’s just plain bizarre. In the early days, we had Beethoven’s Fifth Symphony done entirely in farts. We have designated genres for adult content, and request that artists put “parental advisory” text around their songs where appropriate. I am not aware of any music we have denied unless it was in violation of copyright. Cover songs are also not allowed on our site for legal reasons. We expect the world to decide what they think is good music; something that one person cannot stand is someone else’s favorite song.

SH: Who are some of your favorite musicians/bands?

SVZ: I personally really enjoy listening to music. I must have a collection of about a thousand CDs, and I seem to buy more every week or so. I mostly listen to a lot of dark electronic music ranging from X marks the pedwalk, Haujobb and Die Form, to Forma Tadre, Dementia Simplex, Violet Arcana, Internal, and Consequence. Some of these artists have music available on MP3.com, check them out if you’re interested.

In this chapter:

- *The Architectures*
- *The Codecs*

9

Competing Codecs and Other File Formats

MP3 isn't the only game in town. There are many other audio file formats out there trying to compete with MP3, with varying degrees of success. Some of the alternative formats are meant to establish security and guard against piracy, some aim to provide better audio quality at smaller file sizes, and most try to accomplish both goals at once. Despite the fact that some of the formats you'll read about in this chapter are technically superior to MP3, don't expect MP3 to go the way of the pterodactyl anytime soon. MP3 is deeply entrenched, and seems to have struck a near-optimal balance between simplicity, openness, and quality for most users. MP3 software and hardware is everywhere, not to mention the zillions of legal and illegal MP3 files themselves. Any contending format faces a massive uphill battle to popular acceptance, and the only way the world is liable to change horses in mid-stream is if the industry forces a new format upon us; a dubious proposition at best. Nevertheless, many of the competing formats represent excellent technology, and deserve consideration.

Because many of the competing formats are actually much more than simple audio codecs, we've broken this chapter up into two major categories: First we'll look at a few of the "architectures"—complete audio/video/multimedia authoring and playback platforms that encompass audio compression as a subset. Then we'll take a closer look at a few specific codecs in use out there, some of which are actually components of the larger architectures.

You'll also find a "What this technology means for MP3" follow-up at the end of each section. A codec comparison chart can be found at the end of the chapter, as well.

The Architectures

It's impossible to discuss the future of the MP3 scene without also touching on video issues. As available bandwidth and storage space increases, it's likely that users will increasingly begin to collect and swap movie clips and interactive presentations as readily as they do audio files today. RealPlayer G2, Windows Media, QuickTime 4.0, and MPEG-4 all represent complex, multifaceted presentation formats of which audio is but one component.

In the near future, users will be able to create and actually "participate" in interactive presentations that combine audio and video in ways we can only dream about today. For example, users might be able to download music videos that feature themselves as lead singer, pick up and carry objects through three-dimensional space, and enjoy it all through multi-channel home theatre systems.



In contrast to MP3, many of the file formats, architectures, and codecs described in this chapter are proprietary technologies. The significance of this fact should not be underestimated. Even if alternative formats can be proven to provide distinct advantages over MP3, one should carefully consider the importance of keeping formats and specifications open. It is precisely MP3's openness and wide availability that has allowed it to flourish on all operating systems, and to give rise to an extensive industry of related software. In general, open standards promote the interests of consumers and most likely artists, while closed standards benefit labels, patent holders, artists, and other intermediaries in the traditional business model. Note that artists are caught in the middle here. While artists can benefit from greater exposure via file-based music distribution, they also stand to lose through piracy. A seemingly ideal compromise for most parties involved would be an open format such as MP3, but which still provided some sort of security or copy protection mechanism. Keep an eye open for news of the ability to protect MP3 files through advanced use of ID3v2 tags; this will afford us the benefits of MP3's nonproprietary nature while still satisfying the most important requirements of artists.

ASF/MS Audio/Windows Media Technologies

www.microsoft.com/windowsmedia/ Advanced Streaming Format, or ASF, is a subset of the Windows Media codec, which is designed to compress both audio and video into Internet-friendly sizes that can be streamed to users over 28.8 and faster modem connections. Windows Media includes a number of pieces that all work together, including compression tools, a playback component (Media Player) and a streaming server. Microsoft positions the codec as a replacement for two of its largest threats in Internet audio: RealAudio's G2 streaming codec, and MP3 itself.

Notes on Windows Media Nomenclature

Confused about the ASF/MS Audio/Windows Media nomenclature? Microsoft and others have used these terms nearly interchangeably, so you're not alone. Technically, "Windows Media Audio (WMA) files are ASF files that contain audio content compressed with the Windows Media Audio codec." According to Microsoft, "These files are named with the file extension 'WMA' so they can be easily identified as audio and distinguished from other ASF files that may have video." But a single codec encompasses both audio and video, and you can name your files with either *.WMA* or *.ASF* extensions—it's your choice, and most users stick to the *.ASF* extension, as do the examples shown in this section.

At its introduction, Microsoft claimed that MS Audio was capable of compressing audio nearly two times smaller than MP3 with comparable sound quality. Tests conducted by a number of organizations and individuals after the release yielded mixed results. However, most agreed that Microsoft's claims were close to true, with caveats. For example, some listeners found that stereo imaging was noticeably degraded in WMA, due to weak stereo compression algorithms used in the codec. Others noticed a persistent "echo" effect and general "thinness" in WMA files, along with a "blurring" or "rushing" at the high end (particularly with percussive or aggressive sounds) which MP3 does not have, even when both the WMA and MP3 files are encoded at 128 kbps. Finally, many listeners note that WMA performs very well with music, but terribly with the spoken word. Nevertheless, many listeners agreed that, with most types of audio signal, Microsoft had done a pretty good job with the format.

However, some critics noted that in its test, Microsoft had compared 64 kbps mono WMA samples to 128 kbps stereo MP3s. As you may recall, 128 kbps stereo MP3s are basically two 64 kbps mono files working in conjunction (though some factors, such as joint stereo, make this a somewhat fuzzy definition). If listening conditions were sub-optimal and stereo effects were not easily perceptible, an unfair advantage could be shown in favor of WMA. And, in fact, many felt that the test's listening conditions were indeed sub-optimal. A fairer test would be to compare 64 kbps mono MP3 to 64 kbps mono WMA in such an environment.

But even though Microsoft makes available a streaming server component with no cost-per-stream fee, Microsoft is (so far) having trouble getting any significant buy-in on the Windows Media concept in general. Part of this is due to the fact that Real has been in this business for years, and had the market sewn up tight long ago. Microsoft is making some headway, but not what you'd expect given the overall quality of their product. Additionally, Real offers some server functions that

are not present in Windows Media, such as better handling of artifacts introduced by lost packets in Internet transmission and their "SureStream" technology, which offers the ability to shift delivery rates to accommodate available bandwidth at any given moment. However, the better quality offered by Windows Media at lower bitrates will probably compensate for these omissions in most cases and for most users.

The best way to determine whether you should deliver your content via Windows Media, MP3, or RealAudio is to do some real-world tests. If you don't want to do all the encoding yourself, both Microsoft and Real have set up their own comparative listening tests, both of course aiming to prove that their own format is optimal. See www.real.com/msaudio/ and/or www.microsoft.com/windows/windowsmedia/en/compare/.

Playing ASF files

The default player for all Windows Media, including ASF files, is the Windows Media Player. If your version of Media Player isn't yet configured to handle ASF files, the codec will be downloaded and installed automatically the first time you attempt to play one. This is a nice feature, but note that the codec isn't downloadable except via MediaPlayer.

However, playback isn't limited to Media Player. Later versions of WinAmp, Sonique, and MusicMatch JukeBox also include ASF support, as does Diamond's RioPort. In addition, increasing numbers of portable and home stereo-based players are gaining ASF support. The portable market is an ideal candidate, because the limited amount of memory available on portable devices often limits total playback time to less than an hour with MP3. With Windows Media's tight compression, users are able to stuff more files of an acceptable quality into the limited amount of available memory. Because Microsoft is, well, Microsoft, expect to see ASF support starting to appear in virtually all major audio software and hardware products.

Does this mean that ASF could eventually topple MP3? That's certainly Microsoft's intention, but don't count on it happening anytime soon. The fact that ASF includes digital copyright protection mechanisms means many users will shun it, opting instead for the protection-free MP3 format already in such heavy use. In addition, people who encode to ASF now will be creating files that are not playable for users of alternative operating systems, or for users of MP3 players that don't support ASF. Since people naturally want their files to be playable by everyone, they'll likely opt for MP3. However, remember that Microsoft has applied to the International Standards Organization to have ASF declared a standard, which should theoretically open the doors to ASF support on all platforms and all players. In all likelihood, many large labels will side with Microsoft on ASF (Sony

already has), which will mean that much legitimate content will become available only in ASF format. Many users will thus be forced to adopt the format, at least in part.



It may become possible to convert ASF files to MP3. If an audio application supports both, and if its security features are not overly paranoid, the conversion may be as simple as pulling down File → Export. However, if an ASF file has security features enabled and the application respects these, export to other formats will likely not be possible. Your only recourse in that case may be to play back the ASF file normally and record the sound stream moving through the system with an audio recording application. Theoretically, in a system full of SDMI-enabled hardware, even this would be difficult, though clever users and hackers will always find a way. To convert MP3 files to ASF, see the next section.

Creating ASF files

To create ASF files, you'll need to download the Windows Media Tools from www.microsoft.com/windowsmedia/. The tools are very clearly documented, and you shouldn't have much difficulty figuring out how to use the Windows Media Encoder, as shown in Figure 9-1. Here we're creating an ASF file from an existing 128 kbps, 5.03 MB MP3 file. The output file was 64 kbps and 2.57 MB. Audio quality differences were negligible through an average-quality sound card and computer speakers, though a quality sound card and speakers uncovered notable weaknesses in the ASF file. Note, however, that you'll want to choose Custom Options when you launch the encoder if you want to re-encode an existing file. The resulting wizard will let you choose an existing WAV or .MP3 file (oddly, the option to choose MP3 as an input format is not available if you don't choose the right Wizard on startup). Because the encoder is capable of re-encoding existing MP3 files to the Windows Media format, you have the opportunity to re-do your entire collection and save space on your hard drive. If you're planning to distribute files, however, remember that the encoded files will not be playable by other users on other machines, due to the copy protection mechanisms built into the Windows Media system.

The Windows Media codec is also included in an increasing number of tools from third-party vendors. At this writing, ASF audio/video files could be created in Adobe Premiere, SonicFoundry SoundForge, and others.

Linking to and streaming ASF files

While you can always link to an ASF file in the standard manner:

```
<a href="windows_blues.asf">Click here to listen</a>
```

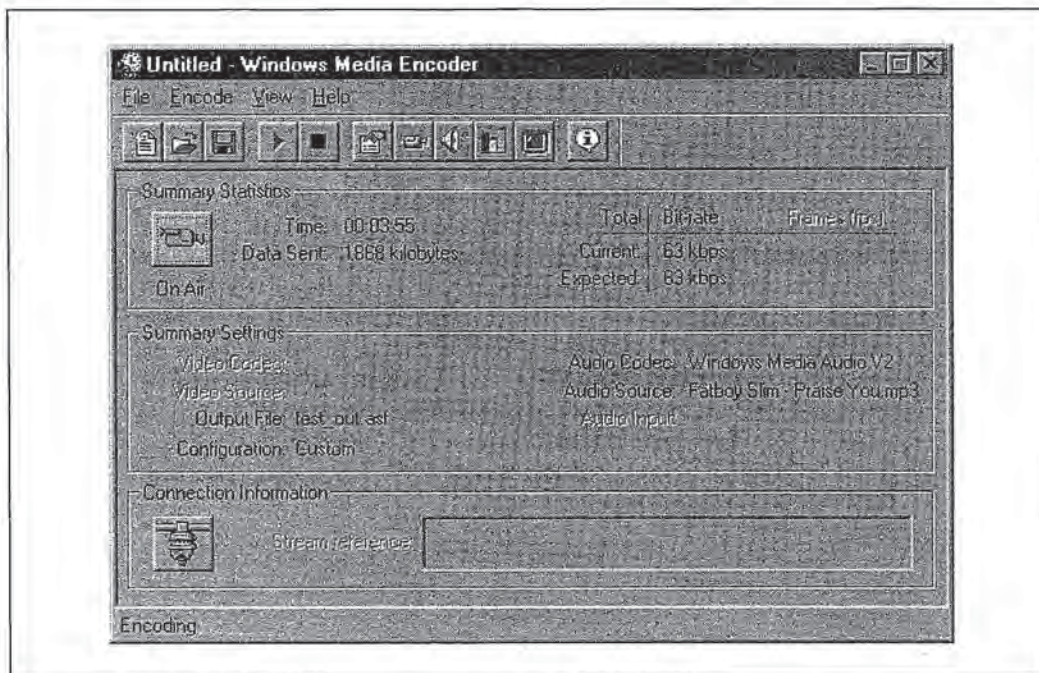


Figure 9-1. Windows Media Encoder gets the job done

doing so will simply initiate a download of *windows_blues.asf*, just as it would with an MP3 file. This may or may not be what you want. In many cases, you may want to let your listeners hear the file, but not save it to their hard drives easily. Because Media Player and most other audio players do not have a simple File → Save As... option, you can make it tougher for users to save your files by enveloping the file's URL in an ASX "locator" file. Doing so will not only give you a degree of protection against unwanted file saving, but it will also cause the file to be pseudo-streamed. In other words, the file will start playing for the user as soon as enough bits have been transferred, rather than forcing the user to wait through the entire download. Pseudo-streaming is managed by the client (in this case, Media Player) and can be done via any web server. We'll get to real streaming in a moment.

The role of the ASX file is to pass control of the download from the browser to the player. ASX is nothing more than a simple text document (in XML format, if you must know) that includes the reference to the actual ASF file (Figure 9-2). If you've worked with RealAudio before, you're already familiar with this process.

Create a new text file that looks like this:

```
<ASX version="3">
<Entry>
  <ref href="path\windows_blues.asf"/>
</Entry>
</ASX>
```

The href portion can be any form of URL: local, relative, or absolute. Save this file as *windows_blues.asx*. Then, in your page code, link to the *windows_blues.asx* rather than to *windows_blues.asf*. Upload the ASX and ASF files to your web server along with your HTML, and you're all set.

The ASX file can optionally include a number of additional options, including references to banners, images, copyright statements, and even playlists. See *msdn.microsoft.com/workshop/imedia/windowsmedia/crcontent/asx.asp/* for details.

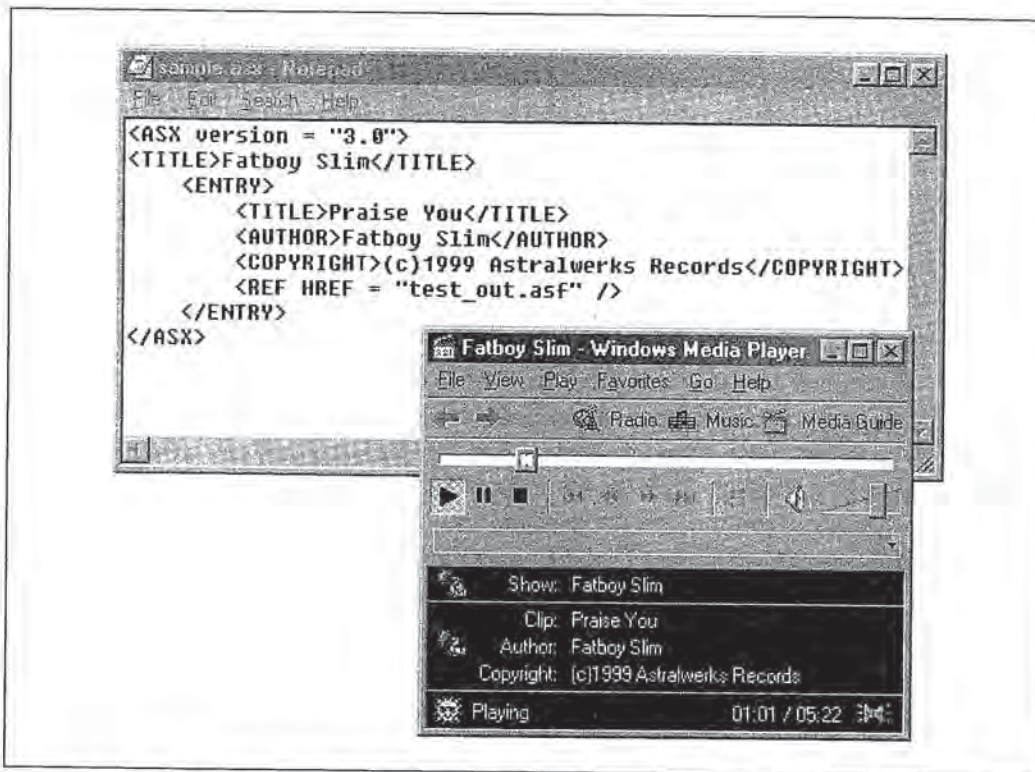


Figure 9-2. The ASX file provides meta-data for Windows Media Player



Devious users can choose to save the ASX file to their hard drives by right-clicking the link in the browser. Once they have the ASX file, they can open it in a text editor, find the actual URL to the ASF file, and download it directly, thus circumventing your intention to prevent users from saving files to disk.

If you expect a lot of traffic on your site, or if you want to dish up live feeds with Windows Media at multiple simultaneous bitrates, you can set up the optional server components to manage and load-balance traffic, parse your available bandwidth, and deliver your audio in true streaming fashion, rather than pseudo-streaming. See *www.microsoft.com/windows/windowsmedia/serve/* for details.

Unfuck

The same day Microsoft officially rolled out MS Audio 4.0, including in their press release the phrase “proven digital rights management,” a hack called “unfuck” surfaced in the MP3 community, capable of disabling the copyright protection in MS Audio files. Essentially, unfuck takes protected .WMA files, outputs them to WAV format, then re-encodes the WAVs in unprotected .WMA format. Testers at MP3.com tried the hack on one of their own Windows Media files and were successful in unlocking the file’s protection mechanism. Microsoft vowed to tackle the situation with a software fix, but a permanent fix is theoretically impossible, since the crack works “outside” of the format itself. Like the Audiojacker and Totalrecorder programs that came before it, unfuck works by siphoning off the audio stream as it leaves the player, on its way to the sound card. Playback interception techniques will remain a staple of the audio hacking world, because there will always be a way to do so outside the control of any security mechanism currently in place (although hardware-based security mechanisms could theoretically make this much more difficult, if not impossible).

But the swiftness of the hack provided further evidence of the uphill battle faced by any vendor attempting to introduce security or digital rights management mechanisms into software and/or hardware: The hacking community is incredibly adroit, and apparently committed to overcoming any obstacles that lay in the path of free music distribution, with or without the approval of the artists and labels in question. The hack/counterhack tug-of-war inevitably follows on the heels of any product calling itself “secure.”

Larger problems potentially loom. If utilities such as unfuck take off and end up representing a substantive threat to the music industry, they retain the right to revoke the device or software’s SDMI-compliance rating (though MS Audio 4.0 is not SDMI-compliant since SDMI had not yet been finalized when it was released). If enough companies depart or are forced to depart from SDMI, the whole plan potentially unravels. Remember, though, that the Digital Millennium Copyright Act (see Chapter 7, *The Not-So-Fine-Print: Legal Bits and Pieces*) makes it illegal to circumvent copyright security mechanisms, so hackers caught behind the wheel of tools such as unfuck could face serious penalties. On the flip side, such hacks may be *necessary* to retain the user’s right to make copies of music they already own. Anyone got an aspirin? The new Cold War is confusing.

What Windows Media means for the future of MP3

If for no other reason than the fact that Microsoft is able to include Windows Media playback tools with every copy of Windows (and, potentially, authoring

tools as well), Windows Media stands a better chance of usurping MP3's domination than any other competing format, which is why I give the format such detailed coverage here. In addition, major labels and the SDMI group will of course work closely with Microsoft to work on and continue to improve file-based security models.

Marketing isn't the only thing Windows Media has going for it. The company has produced a format with excellent compression and fairly high quality, meaning that, in addition to the wide distribution of ASF/WMA files we're liable to see coming from major labels, millions of users will likely adopt the format for their personal use as well.

The biggest battle Windows Media faces is the fact that it arrived late to the game, when other architectures and formats had already formed an established user base. But Microsoft has demonstrated their ability to chart a new course on a dime (e.g., with Internet Explorer), and one should not underestimate their ability to achieve major consumer buy-in in a relatively short period of time.

Without releasing source code to their internal codecs, however, Microsoft will have trouble gaining any kind of favor with the alternative systems crowd, who form a large cross-section of the MP3 enthusiasts universe.

QuickTime

quicktime.apple.com It's a floor wax! It's a dessert topping! Ask a hundred people to define QuickTime, and you'll get a hundred different responses. That's because QuickTime isn't a single codec, but an umbrella format—an entire architecture encompassing dozens of different audio and video compression and multimedia playback techniques. QuickTime is a long-established industry leader used for creating multimedia presentations, delivering Internet video, embedding hypertext references in movies, creating interactive panoramic scenes, delivering streamed media to users over modems, synchronizing audio/video content through time-code sequences, and more. This flexibility has made QuickTime a multimedia leader not just on the Mac platform, but on virtually all platforms. Major browsers for Windows, for example, come bundled with the ability to play back QuickTime movies, even though the product is one of Apple's crown jewels. In fact, QuickTime may represent one of Apple's greatest successes in reaching users of non-Apple platforms.*

* Apple does, however, retain a few bits and pieces just for Mac users. For example, the excellent Sorenson codec can be used to generate very high quality movies with small file sizes. Many users of Windows and other platforms were frustrated when prerelease trailers for *Star Wars: The Phantom Menace* appeared at first only in Sorenson-encoded QuickTime format, playable only on MaM on the Macintosh.



It was QuickTime's extreme flexibility and wide adoption that led to the MPEG working group recommending to ISO that the QuickTime file format be adopted as a basis for the creation of MPEG-4 presentations (see the next section).

While QuickTime itself can be used for creating audio-only bitstreams, its primary role in practice is as a video and multimedia authoring/playback environment, with a reach extending far beyond that which concerns the audio-centric nature of this book. Audio-only QuickTime downloads are comparatively rare on the Internet, so we don't cover them here. QuickTime authors using Version 4.0 or later of the system, along with the QuickTime Streaming Server, can avoid the step of having to convert audio/video presentations to RealAudio or Windows Media formats before placing them online. Inclusion of the QDesign codec (covered later in this chapter) means that people using QuickTime for audio can take advantage of the extreme compression ratios that format provides.

Note that the QuickTime 4.0 player and above is capable of handling existing MP3 files, and QuickTime authors can embed MP3 files into their movies as soundtracks.

What QuickTime means for the future of MP3

Because QuickTime has been around for so long, and because its capabilities are so extensive, it's almost surprising that QuickTime hasn't become more popular as an audio-only distribution format. But search the Internet for downloadable music, or look through Apple's QuickTime Showcase, and you'll be hard-pressed to find much in the way of downloadable music unaccompanied by video. However, its inclusion of the QDesign codec, which threatens to give QuickTime a real edge in audio compression, is relatively new. It may take a while for users to catch on to the compression advantages they can achieve with QDesign. And, of course, the QuickTime Streaming Server means users will have an easy time of setting up Internet broadcasts of their own.

Considered as a complete audio/video compression solution, QuickTime will continue to be a major player and may even achieve greater buy-in as users wait patiently for MPEG-4 to reach maturity. As an audio-only solution, few people are turning to QuickTime over MP3, though there's not much stopping them. Again, only the openness of MP3 can really account for its popularity in the face of potentially superior solutions.

MPEG-4

Not just an audio coding standard, MPEG-4 is a complete, object-oriented scheme devised by the MPEG working group and completed in October 1998. The specification allows for the creation of low-bitrate audio/video presentations, aiming to cover every media possibility in the digital environment.

Much to the confusion of the general public, the terms AAC, VQF, MPEG-4, and MP4 are often used interchangeably. To clarify: MPEG-4 is the next-generation standard for specifying the compression and manipulation of audio and video objects, where an object is defined as “a component of a presentation” (even if it's the only component.) MP4 will most likely be the name of the file format that will encompass MPEG-4, while AAC and VQF are subsections of the MPEG-4 spec. The audio component of MPEG-4 is technically known as “MPEG-4, subpart 4, General Audio coder,” or GA for short. In fact, GA is an umbrella term encompassing MPEG's AAC codec and Yamaha/Sony's VQF codec (both described independently in this chapter);* an MPEG-4 encoder should give you the option of creating either AAC or VQF files. Both AAC and VQF are capable of delivering better audio quality at 96 kbps than MP3 does at 160 kbps; i.e., they deliver the same level of quality as MP3 at lower bitrates.



Nomenclature is further confused by the release of a file format cheekily called “MP4” by Global Music Online, which is essentially a proprietary implementation of AAC (someone needs to explain all of this to Chuck D, who releases his own material in the proprietary “MP4” format).

It would be a mistake to think MPEG-4 is nothing but a more advanced compression format. In fact, the specification allows for some very futuristic features. If you have trouble visualizing the need for some of these features, picture the video-phone in your house running over plain old telephone service, and checking out virtual items in a virtual store prior to buying. You might be able to “log in” to an interactive work of fiction, and manipulate objects in the characters' world. Virtually anything you can imagine that involves media and requires compression (i.e., to be transmitted or stored efficiently) will be able to benefit from the MPEG-4 spec. It's pretty amazing.

* Just because GA encompasses both AAC and VQF, don't be fooled into thinking that AAC and VQF are in fact MPEG-4 files in and of themselves—they're not. Converting between them should be possible given appropriate tools, but we're talking about three separate file formats here.

The important thing to remember is that MPEG-4 projects will not represent a single file format, but rather a variety of compressed formats delivered as independent objects, along with a scheme for “gluing” them together and defining their relationships and interactions through the BIFS description language (see later). This object-oriented model will allow for much more advanced control and customization options; audio and video can be compressed independently so that everything is optimized for both speed and functionality.

Some of the benefits that MPEG-4 will enable include:

Complete interactivity

Thanks to the Binary Format for Scenes (BIFS) description language, users will be able to “manipulate” objects in space—set objects into motion, “pick up and carry” objects in an audio/video environment, and do complete walk-throughs of three-dimensional scenes. These objects can be either synthesized (generated with 3D A/V modeling tools or CAD programs) or originate from “natural sources.” For example, a source video might include footage of a wine glass sitting on a table. In its MPEG-4 implementation, you might be able to pick up the wine glass and “drink” from it.

BIFS will allow for full streaming support

Users will be able to begin partaking in a scene before the whole thing has been downloaded.

Support for “structured audio”

Provides a mechanism for the generation of sounds, rather than simply playing them back. In some ways, structured audio is similar to MIDI, with the critical distinction that structured audio describes *methods* of synthesis. As you may have experienced in the past, General MIDI often sounds “cheesy,” because its palette is limited to the sounds built into your sound card. You have no guarantee that the sound of the MIDI file you’re listening to is the same as the sound the creator heard on her own sound card. With structured audio, the actual sounds of the instruments being used are accurately described, which guarantees that the sound the artist hears is the sound you hear, while still retaining tiny file sizes (since the audio signal is described, rather than stored bit-by-bit).

Support for multiple tracks

From mono to multichannel surround sound.

Audio is represented in the form of objects

In fact, object support in MPEG-4 is actually more sophisticated for audio than for video. What will this get us? Authors will be able to create interactive scenes where audio sources appear to originate from different points in space, depending on the relation of the moving listener object (you) to the moving

audio source. Authors can specify different sound scenes for different listening situations, so they can simultaneously optimize for playback of in-car audio and for home theater situations, for example. In fact, the object-oriented architecture of MPEG-4 borrows heavily from VRML (virtual reality modeling language), so three-dimensional soundscapes are built into the whole concept of MPEG-4. In another possible scenario, users may be able to take an existing recording and manipulate its tracks independently, as long as the musician packages the project in such a way as to make this possible.

The standard holds as a key objective decent audio quality at low bitrates

A number of ground-breaking techniques have been brought to bear on the problem (no single solution is the "official" one, since the objective is only to achieve compliant bitstreams, not to completely define the means of creating these bitstreams). Futuristic techniques such as morphology and fractal-based compression are being worked on. Whereas MP3 is pretty much able to duplicate CD quality at 160 kbps, AAC can achieve CD quality at 96 kbps, and good quality at 64 kbps. Remember: CD audio data is stored at 1411.2 kbps.

The possibility of including multiple "sound fonts" in a single audio file

The user could theoretically choose filters from a list and apply them to the audio in order to simulate the sound of different environments (from clubs to concert halls, for instance).



If some of these features sound familiar, perhaps it's because many of them are also found in Apple's QuickTime format and in the Windows Media system. In fact, ISO officially adopted QuickTime as the basis for MPEG-4 in February, 1998. Microsoft, as you can imagine, was not pleased; the company had been trying to get the organization to use what later became Windows Media in this role. Apple was backed by IBM, Netscape, Oracle, Silicon Graphics, and Sun Microsystems. The recommendation of the format by these companies to the ISO standards body was based on a consensus that the QuickTime format already provided many of the features ISO wanted to see in MPEG-4, such as real-time audio and video streaming, the ability to be streamed across different network protocols, and support for a huge array of pre-existing digital media types. The fact that QuickTime already enjoys broad adoption by media professionals couldn't have hurt either.

Playing and creating MPEG-4

At this writing, there is no *complete* implementation of the MPEG-4 standard. However, bits and pieces of it have been implemented in various products. Most notably, you'll find some AAC and, to a lesser extent, VQF support in many popular

players. See those sections of this chapter for details. One of the early companies working on a complete MPEG-4 authoring suite is PsyTEL Telecom. Keep an eye on www.psytel.com/mpeg4.htm. Once the specification is finalized, expect to see many companies competing to create complete authoring solutions. As MP4 tools do become available, don't expect them to be free or inexpensive, like MP3 tools are. At least in the initial stages, MP4 authoring tools and encoders are likely to be highly complex and therefore expensive. Whether any freeware MP4 tools become available remains to be seen, although one can bet that if sample implementations are released, the Open Source community will be eager to develop Open Source MP4 authoring environments.

MPEG-4 copy protection

The MPEG-4 specification does not include its own means of implementing copyright protection. Instead, it provides a collection of "hooks" which other organizations can use to implement their own security mechanisms. The idea here is to keep MPEG-4 object-oriented and to help the format steer clear of becoming obsoleted by future advances in security technology. It's a safe bet that much thought will go into making sure security mechanisms are indeed secure, and possibly SDMI-compliant. Of course, virtually any security mechanism will be circumvented eventually,* but don't expect to find the same wealth of unprotected MP4 files floating around out there that one finds in the MP3 world. Of course, it may also be possible for people to create MP4 encoders that do not take advantage of security hooks. Coupled with non-SDMI hardware, such tools may make unprotected MP4 files possible. Everything remains to be seen on this front.

What MP4 means for the future of MP3

As you can see from the description above, MPEG-4 really is a completely different animal from MP3, since it encompasses so much more than just audio. As a result, asking what MP4 means for the future of MP3 is sort of the wrong question to ask. At this point in time (early 2000), MP4 is basically a non-issue for MP3 users. Much of it is still up in the air, there are very few tools available for creating MP4 files, and the ones that are available aren't quite mature yet. Licensing issues are still unresolved, and the industry isn't likely to launch another format that will be prone to the same problems MP3 has gotten itself into, with the patent holder scrambling around trying to collect fees *post facto* from developers who have implemented the technology without permission. It's not even clear yet who will be responsible for collecting MP4 licensing fees (although we do know that Dolby is handling licensing fees for AAC, one of the audio components of MP4).

* At this writing, security mechanisms built into DVD had just been cracked by Linux users trying to reverse engineer the format to create Linux DVD playback facilities. While they only stumbled across the security key because an application vendor had left it unencrypted, it is precisely this kind of human oversight that ultimately leads to the downfall of "uncrackable" security mechanisms.

When MP4 does reach maturity and is put into wider circulation, its extensive feature set will enable all kinds of futuristic audio applications, some of them fanciful and some genuinely useful. Tight compression, high quality, and multi-channel audio streams are likely to be the features of most interest to current MP3 users. It's more realistic to expect wider implementations of MP4 sub-components AAC and VQF before we see a complete MP4 implementation.

RealAudio

www.real.com Easily the most well-established corporate player in the Internet audio compression business, RealNetworks specializes in streaming audio and video, where it holds around 75% of the streaming market at this writing. However, Real is now facing competition from Microsoft's Windows Media services, which provide excellent codecs and publishing systems without the costs associated with Real's higher-end programs. Real holds the technological edge in certain key departments, however, and enjoys the mindshare they've earned by making their product a de facto standard half a decade before Microsoft emerged with anything that could be considered a serious challenge. Perhaps not coincidentally, Real's CEO is a highly competitive ex-Microsoftian who parted on bitter terms with the behemoth long ago.

While Real's algorithms were originally designed for very low bandwidth voice/radio transmission (remember, these were the days of the 9600 baud modem), they've long since been optimized for music as well. The Real codec is now capable of generating comparatively high quality at virtually all bitrates, and can deliver quality close to that of FM radio at 20 to 34 kbps (though a bit of high-frequency distortion is a common criticism at higher bitrates under careful listening tests). Interestingly, some audiophiles have even been led to wonder aloud why MP3 gets all the action when virtually everyone out there has a Real player installed on their system, and the quality of G2 encodings is so good (G2 being the name of Real's latest encoding/playback system at this writing), as shown in Figure 9-3. Users can create RealAudio files either in Real Jukebox (Chapter 3) or with specialized tools available at extra cost from Real. Of course, such an argument ignores the openness of MP3 and the wide availability of tools. Real's encoder is also very fast.

Of course, the presence of Windows Media could have a significant impact on the matter, and Real has some interesting observations about Real products vis-a-vis Windows Media. Microsoft, of course, has published test results aiming to show that Windows Media can provide a superior alternative to Real. And Real, of course, has published a rebuttal proving just the opposite. The two make for interesting back-to-back reading are www.microsoft.com/windows/windowsmedia/en/compare/ and www.real.com/msaudio/.

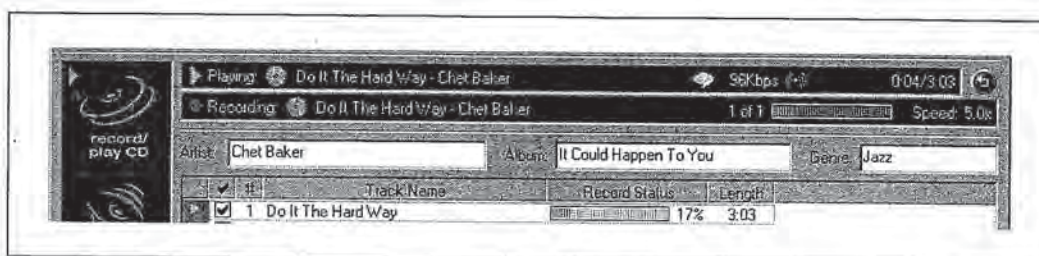


Figure 9-3. G2 files at 96 kbps often sound as good as MP3 files at 128 kbps

Perhaps one of the biggest reasons Real products haven't seen the widespread adoption of MP3 is because MP3 is primarily seen as a storage format, for building music collections, while Real has for so long positioned themselves as a streaming audio/video company. Few people even consider the possibility of saving Real files to their hard drives because for years they've clicked on a .ram link and had the file streamed to their Real players, without any option to save the file. There's nothing preventing anyone from downloading the G2 encoder and building a collection of music on their own hard drives, but it's just not what people think of doing first with Real technology. Nevertheless, you can easily use G2 encoder or the Real Jukebox covered in Chapter 3 to build your own collection of high-quality RealAudio files. If you want to use Real Jukebox to create Real Audio files rather than MP3, enter the Preferences menu, choose Encoding Options, and select "RealAudio."

What Real Player means for the future of MP3

Technically speaking, there's no real reason why users shouldn't be storing their collections in Real format rather than in MP3. But even Real has capitulated to the popularity of MP3 by promoting Real Jukebox so heavily, which creates MP3 and Real files with equal ease. While RealAudio files are ready to stream, Real's streaming server solution is still prohibitively expensive for casual users (though RealAudio can easily be pseudo-streamed from any standard web server). And music can be compressed with audio quality equivalent to that of MP3 at smaller file sizes. Real Jukebox even gives you the option of enabling or disabling security features, so users who choose to use the product illegally can do so.

Real Jukebox (Figure 9-4) is being installed by millions of users, and some computer manufacturers are even pre-installing the tool on new machines. But despite its technical advantages, users are going to continue focusing their music encoding efforts where the rest of the world is: on MP3. Because Real Jukebox "swings both ways," it's apparent that Real knows this and is doing what it can to not be left out of the scene entirely. Expect Real to remain a major player in Internet streaming, but don't look for it to take over MP3's slot in file-based storage and distribution.

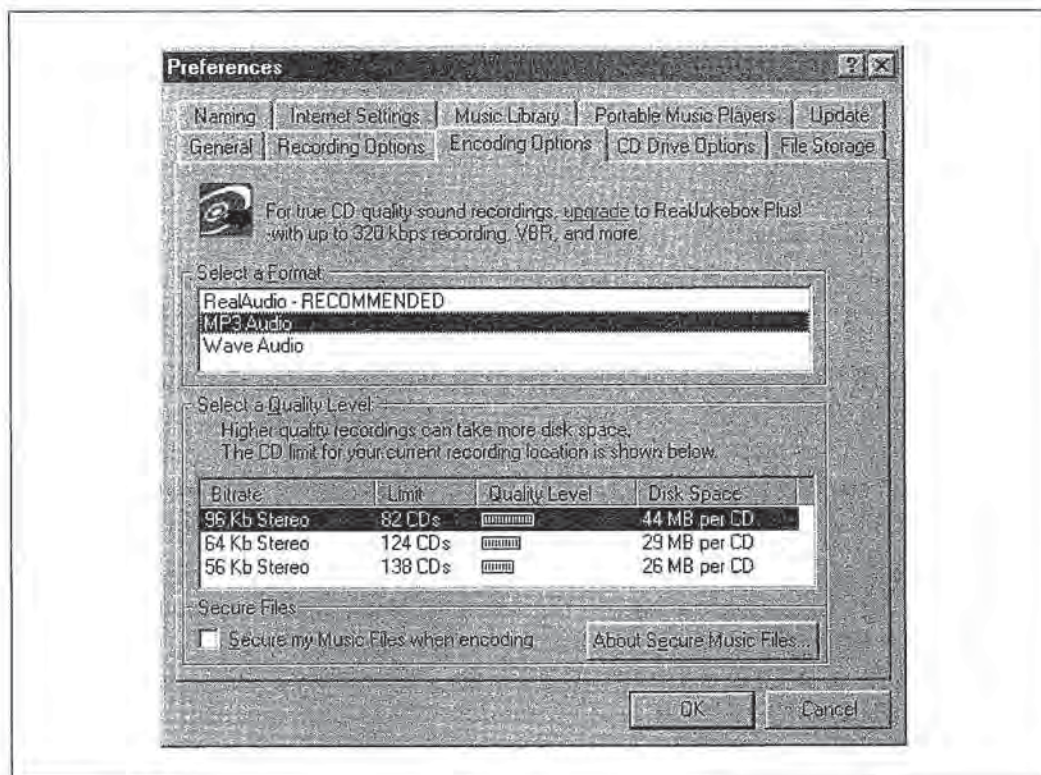


Figure 9-4. The free version of Real Jukebox is limited to 96 kbps encodings

The Codecs

In almost all cases, the alternatives to MP3 also use some form of perceptual codec, as described in Chapter 2, *How MP3 Works: Inside the Codec*. That means that all of these formats achieve their compression by discarding or reducing the amount of data representing inaudible sound (though the question of whether any sound is truly inaudible is still up for grabs; some diehard audiophiles maintain that even inaudible sounds contribute to music's overall atmosphere).



This section only covers alternative audio *compression* schemes and file formats; there are many noncompressed formats that aren't really of interest to us here.

Throughout this discussion, keep in mind that if you decide to test some of these alternative codecs or formats, most of which claim smaller file sizes and higher quality, you're not going to get smaller files at the same bitrate. In other words, regardless of what format you use, 128 kbps means what it says: For every second

of music, you store ~128,000 bits of data. The question is, how much quality can you get out of that much data? What you really want to discover is how low of a bitrate can you use and still get "CD quality" or "near CD quality" sound. Where there's money and improvements to be made, there's fierce competition. Don't expect it to let up anytime soon; anything could happen between now and next year.

AlbumDirect

At the time of this writing, it was too early to tell exactly what IBM has in store for us on the digital music distribution front. What we do know is that they've set up a broadband test system for the secure, paid delivery of digital music in selected geographical regions. The test, code-named "The Madison Project" and marketed as "AlbumDirect," will probably take on another name if it ever sees the light of day.

The intent of the test is to measure the reaction of the mainstream consumer, rather than focusing on people who use computers heavily and are already sold on MP3 or other formats; in other words, IBM wants to know how to sell John Q. Public on the idea of paid music downloads. The idea is that Warner Music, EMI, Sony Music, BMG, and Universal Music (The Big Five) are all looking for ways to cash in on digital distribution without losing control of the delivery chain, and IBM wants to provide them with the technology and mechanisms to do so.

We can expect whatever comes of AlbumDirect to be tied to a secure e-commerce system, flexible enough to interface with back-end databases at retail web sites. Users will be able to transfer downloaded music to hard drive or burn music directly to CD-ROM. One can also imagine cable-connected Internet kiosks in music stores that let users swipe their credit cards, enter the names of their 12 favorite songs, and have a custom CD burned on the spot. Similarly, online music retailers would be able to interact with Madison's back-end database to sell music directly to customers, undoubtedly skimming a commission for themselves in the process.

IBM has not revealed anything about the technology they intend to use; the codec, the security mechanism, and the distribution system are all "black boxes" at this point.

MPEG-1 Layer II

www.mpeg.org So whatever happened to this stalwart standard? Believe it or not, MPEG-1 Layer II (a.k.a. MP2) is still alive and kicking, and even in heavy use in some places. Defined as a standard for use in digital video, storage, and broadcast-

ing in 1994, MP2 should not be considered obsolete just because it's old. Many television and video production studios still use MP2 heavily, and the format remains popular among owners of older computers, due to the fact that decoding MP2 is less computationally intensive than MP3. And don't be fooled for a second into thinking that MP2 sounds bad. In fact, many people feel that MP2 is still the quality king, and is able to create output more faithful to the original source signal than MP3. But what does that statement really mean when you're dealing with MP3 bitrates high enough to make the encoding nearly indistinguishable from the source file anyway? For most people, the answer is "not much."

The MP2 codec is much less complex than MP3's, so MP2 players are more efficient (consume fewer resources) and MP2 encoders are very fast. Of course, there is a downside—MP2 files are roughly 50% larger than corresponding MP3 files, and it was, after all, MP3's small file sizes that made it a good Internet citizen and contributed heavily to its widespread adoption.

Some encoding professionals also appreciate the fact that there is a wide array of DSP-based MP2 hardware out there for use in creating live broadcasts and encoding large collections of audio material. And because MP2 is used for the audio channel in MPEG video, the codec integrates neatly into existing video production systems.

Most quality MP3 encoders include an MP2 encoding option (check your preferences), and most major MP3 players also include the ability to decode MP2 files. Do some listening tests of your own, and you'll probably find that at higher bitrates, there just isn't all that much difference in output quality, and the larger file sizes created by MP2 encoding just aren't worth the storage space. Unless you have a very slow machine, you're better off just using a higher bitrate and sticking with MP3. And if you need to distribute your work over the Internet, forget about MP2 altogether.

What MP2 means for the future of MP3

MP2 does not figure heavily into the future of audio compression because, well, it doesn't compress very well. MP2 encoders are fast and the quality is very high, but while MP2 is not dead, it's not exactly moving forward, either. It offers nothing in the way of security or commerce, lacks an ID3 tagging system, and is favored only by users of older, slower machines. It will remain a stalwart component of some video production facilities (especially those in the third world, interestingly enough), but does threaten the dominance of MP3 in any way.

QDesign

www.qdesign.com One of the lesser-known, but perhaps most remarkable audio compression technologies vying for mind-share is QDesign's Music Codec, or QDMC. Like everyone else, QDesign is going for high quality at low bitrates. However, the company really does appear to be accomplishing some pretty amazing compression ratios with very small files. In some tests, files have been compressed at a ratio of 60:1 while maintaining relatively good quality. Not stellar quality, mind you, but good considering that level of extreme compression.

The secret lies in the approach QDesign has taken in designing their codec. While it's still based on perceptual models, like MPEG, they don't use the static filter-banks (or "transforms") that MPEG utilizes, because these start to break down at very low bitrates. Instead, the QDMC builds a source model from the original signal and breaks it down into "audio building blocks" and develops a custom set of filters for that particular audio signal. Thus, the QDMC is automatically optimized for every piece of music or voice it encodes. The result is that audible distortion is minimized at thresholds well below those where audible distortion enters the picture with traditional codecs. QDesign claims to be able to retain decent quality at bitrates of one bit per sample second.

As you might expect, encoding music with this technique takes longer than it takes to encode MP3, but this is true of nearly all the alternative coders covered in this section. QDesign licenses their coder for use in specialized applications and hardware. For example, they've partnered with Texas Instruments to create new chips for portable players (Figure 9-5), which will reportedly be capable of storing five times more music than portable MP3 devices, in the same amount of memory.

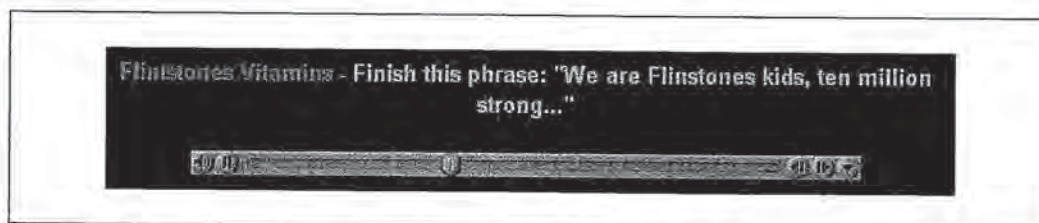


Figure 9-5. The QDesign player can be embedded directly into web pages

Perhaps even more significantly, Apple has adopted the QDesign codec as the basis of the audio component in QuickTime 4. If you've been playing QuickTime 4 files on MacOS or Windows lately, you may have heard the results of the QDesign codec without even knowing it. And since it's built into QuickTime, it automatically supports the Real Time Streaming Protocol (RTSP), rather than pseudo-streaming, as you might be forced to use with other formats. Finally, it means you can encode QDMC through any application that supports QuickTime 4

export, such as Apple QuickTime Pro or Terran Media Cleaner. You'll also find the codec built into popular tools such as Iomega's Record-It. QDesign also provides their own encoder.

If you do use QDesign's encoder, note that it's available in two versions: Basic and Professional. The Basic edition is limited to a few select bitrates, while the Pro edition gets you advanced coding parameters for further quality optimizations, pre-processing controls, faster compression options, and batch-encoding capabilities.

QDesign products plan to be fully compliant with the requirements of the SDMI, including watermarking and encryption mechanisms.

What QDesign means for the future of MP3

QDesign's codec achieves phenomenal compression ratios, but is, at this writing, gaining very few users. However, its inclusion in the QuickTime 4 architecture is liable to give it a much wider range of ears. As a proprietary codec, it lacks the kind of attention currently being shone on MP3 and other general codecs. But as a part of QuickTime, it may achieve a much broader user base. Nevertheless, expect much more attention to be directed toward the audio components of MPEG-4, Real Player G2, WindowsMedia, and QuickTime itself than on QDesign per se.

AAC

The Advanced Audio Codec, or AAC delivers better sound quality than MP3 at equivalent file sizes (or achieves similar quality at smaller file sizes),* and many people believe AAC may be the best candidate we've got for eventually replacing MP3. AAC was co-developed by engineers at the Fraunhofer Institute, AT&T, Lucent, Sony, Dolby, and others, and has been designated by the International Standards Organization (ISO) as a "next-generation MPEG audio standard." In addition, AAC is a component of the MPEG-4 audio coding specification, having been standardized in April, 1997 as ISO 13818-7. But while AAC may be the logical next step in the evolution of MPEG audio, AAC is just a small subset of the whole MPEG Layer IV specification—don't confuse AAC with MPEG Layer IV as a whole.

Unfortunately, AAC is not backwards-compatible with MPEG Audio Layers I, II, or III and was in fact called AAC-NBC (for Non-Backwards-Compatible) in its early stages. The advantage of breaking backwards compatibility was that AAC's developers were free to work with emerging technology breakthroughs without having

* According both to the author's own listening tests and to tests conducted by audio institutes including BBC research, NHK Science and Technical Research Labs, and the Communications Research Center of Canada. Double-blind listening tests demonstrate that AAC consistently provides superior audio quality when measured against popular codecs such as MP2 and MP3.

to saddle them with compatibility restrictions. The downside, of course, is that AAC files can't be played back by existing MP3 players without developers adding new modules or redistributing their applications.

Even though AAC was developed within the MPEG working group, many organizations and companies contributed to its development, including Fraunhofer, AT&T, Dolby Labs, Sony, the University of Hanover, and NEC. Unsurprisingly, Fraunhofer is responsible for the encoder's primary components and integrated the contributions of the other organizations and took responsibility for quality optimizations. Dolby Laboratories is responsible for managing all AAC licensing concerns.

But what makes AAC really interesting is not just the fact that it's better optimized than MP3, but the fact that it's built to work with up to 48 channels, rather than being constrained to mono or stereo. What could you possibly want with 48 channels? Granted, that's probably a little overkill for most applications, but think in terms of surround sound, multiple language options, and control and synchronization tracks, and you start to get the picture.

So, where can you get your hands on an AAC encoder and/or player? Unfortunately, that's the difficult part. At this writing, the pickings are mighty slim. Even though the spec has been in discussion since 1994, AAC has made almost no headway in the marketplace, due to two primary factors: Dolby's iron-grip holding of licensing rights and the complexity of the implementation itself. While you can find AAC plug-ins for some MP3 players and a smattering of AAC encoders at sites such as www.mp3daily.com, complete, full-blown implementations of the AAC codec are simply not out there except as proprietary implementations in players such as Liquid Audio's. As with MP3, the patent on AAC applies to the creation of a "compliant bitstream," rather than to any specific coder implementation. Theoretically, anybody could write their own software in a "clean room" environment capable of working with AAC files, but they would still owe the same licensing fees. And with the enormous complexity of the codec, it's not likely that this will happen anytime soon (if ever).

Meanwhile, the AAC source code available for download only allows developers to build a very low quality coder, certainly nothing that would impress an existing MP3 user. That pretty much leaves deep-pocketed investors to fork over big cash for the whole package. And right now, the biggest company who has decided to go for it is AT&T with their a2b player. And even the a2b player won't play AAC-compliant bitstreams, since AT&T has only implemented the portion of the spec they needed to satisfy their own criteria.

At this writing, there aren't many AAC encoders or players to choose from, and most of what is out there can't create bitstreams that are compliant enough to play properly in all decoders. Search the Internet for Astrid or Quartex AAC encoders,

or check out MBSof's AAC Encoder (fly.to/mbsoft/). K-Jöfol (www.kjofol.org) was one of the first audio players to handle AAC files for playback, but others are likely to support the format as well by the time you read this.

What AAC means for the future of MP3

While there are encoders and players for both AAC and VQF files out there right now, support is generally pretty sketchy, and you won't find a ton of files available for download encoded in these formats. The formats are not "vapor," but neither have they reached anything approaching widespread adoption.

The AAC format holds great promise, but until the industry loosens up around the format, it's going nowhere fast. Because MP3 got out of the industry's hands so quickly, one can surmise that they're going to try and hold onto AAC more tightly. Creating a second defacto standard may not be as easy this time. When and if the format does reach wide adoption, it will likely be on the industry's terms, not those of the public. Once again, ladies and germs, the best technology is not necessarily the winning technology.

Finally, encoding AAC is far more computationally intensive than encoding MP3. When and if AAC encoders do end up in public hands, users better be prepared to either be patient or get a faster machine.

a2b

www.a2bmusic.com One of the early corporate adopters of the AAC codec (covered earlier in this chapter) was AT&T's a2b, a proprietary player including security and digital rights management technologies for artists and labels. Because a2b is based on AAC, it's quality-to-compression ratio is very good. The player implements an Internet encryption scheme called CryptoLib and an electronic licensing system called PolicyMaker, for controlling terms of access and sharing of data (a2b files cannot be played on computers other than the ones to which they were downloaded). More information on CryptoLib and PolicyMaker can be found in the Technology section of a2b's site.

After early positive press, rumors have been floating around that a2b, as shown in Figure 9-6, may be stalling in the marketplace. For one thing, Liquid Audio established itself earlier with more backing and more technology (a2b does not, for example, support direct burning to CD or real-time streaming). In addition, the company was late to the table in taking a stand vis-a-vis SDMI and other initiatives on the industry's table. Meanwhile, AT&T executives appeared to be waffling over where to take the company next in the light of MP3's popularity. After a large group of key a2b engineers left the company en masse, some pundits saw failure in the cards for a2b.

However, a good number of big-name artists have lined up behind the company (including Aerosmith, Alanis Morissette,* and Bonnie Raitt), and of course AT&T has all the funding they need to stay in the game. If only they can find a strategy to set them apart from their competitors.

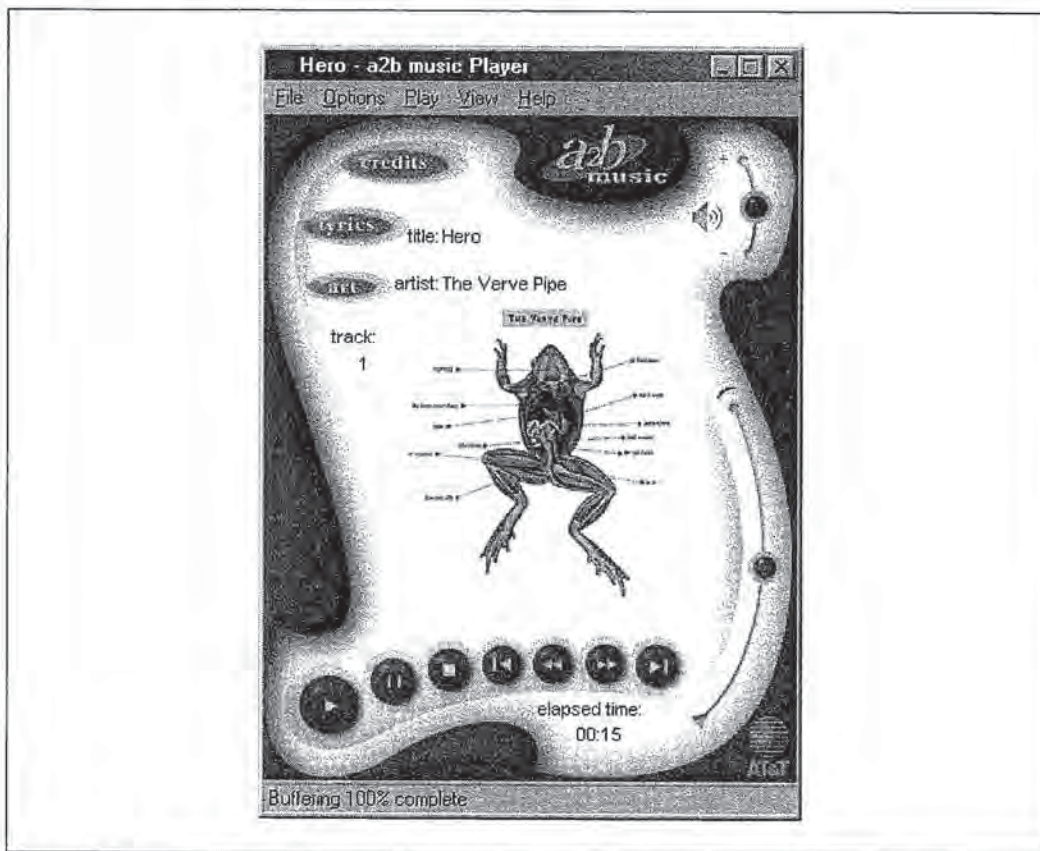


Figure 9-6. The a2b player is based around the excellent AAC codec and a pair of security technologies from AT&T

What a2b means for the future of MP3

a2b has been at this game for quite a while, but have yet to make a major dent in the market. The biggest reason, one can only surmise, is the fact that they've packaged up an excellent codec in a proprietary wrapper, which keeps interoperability at a minimum. Users are able to play back a2b files only on the same machine they downloaded to, and only through the a2b player. Without opening up their range of compatibility, a2b isn't going to increase their marketshare. Confusion within the company about MP3 isn't helping the situation any. It's the codec within the a2b player—AAC—that users should be excited about and waiting for.

* Morissette has also released tracks in MP3 format, so one shouldn't interpret her use of a2b as an exclusive endorsement.

Liquid Audio

www.liquidaudio.com Liquid Audio is one of the pioneers of the online music distribution business. Not that the product predated other methods of digital distribution, but they were one of the first to have a go at making money at it while simultaneously striving to satisfy the needs of the recording industry with integrated sales and security features.

The idea is that artists and labels are able to provide secure tracks or samples, along with built-in ordering capabilities. While you're listening, you can view album cover art, read the lyrics and liner notes, and take part in promotional deals. If you like a sample track, you can enter your credit card information and download the entire album digitally. Because Liquid Tracks are encrypted and watermarked, they must be played on the computer of the person who downloaded them. You can't just purchase Liquid Tracks and share them with your friends.

But while that part of the chain is no longer unusual among digital distribution systems, Liquid goes one step farther by integrating its capabilities with your CD-R device, so you can burn your purchases straight to CD, thus ending up with a product very similar to what you'd get at the record store (except that you never have to leave your chair and don't get the high-quality cover art). And before you get any big ideas, clever boy, no—you can't burn more than one CD copy of a purchased album (although you would still be able to create unprotected MP3 files from your newly burned CD...).

Liquid provides a three-part system, consisting of the Liquifier (their encoder), the Liquid MusicServer (which can be optionally used for streaming content), and Liquid Player itself, as shown in Figure 9-7. The encoder and player run on Windows and MacOS, while the server runs on Windows NT or Unix. The encoder includes digital watermarking options, as well as an "organizer" component that can be used to sift and sort through music collections, or prepare custom compilations for burning to CD. The organizer will let you mix and match Liquid Audio and MP3 files side by side. The Liquid Player offers access to complete lyrics, promotional deals, and built-in e-commerce technology so you can purchase tracks or entire albums online, then burn them to CD through the software's organizer component.

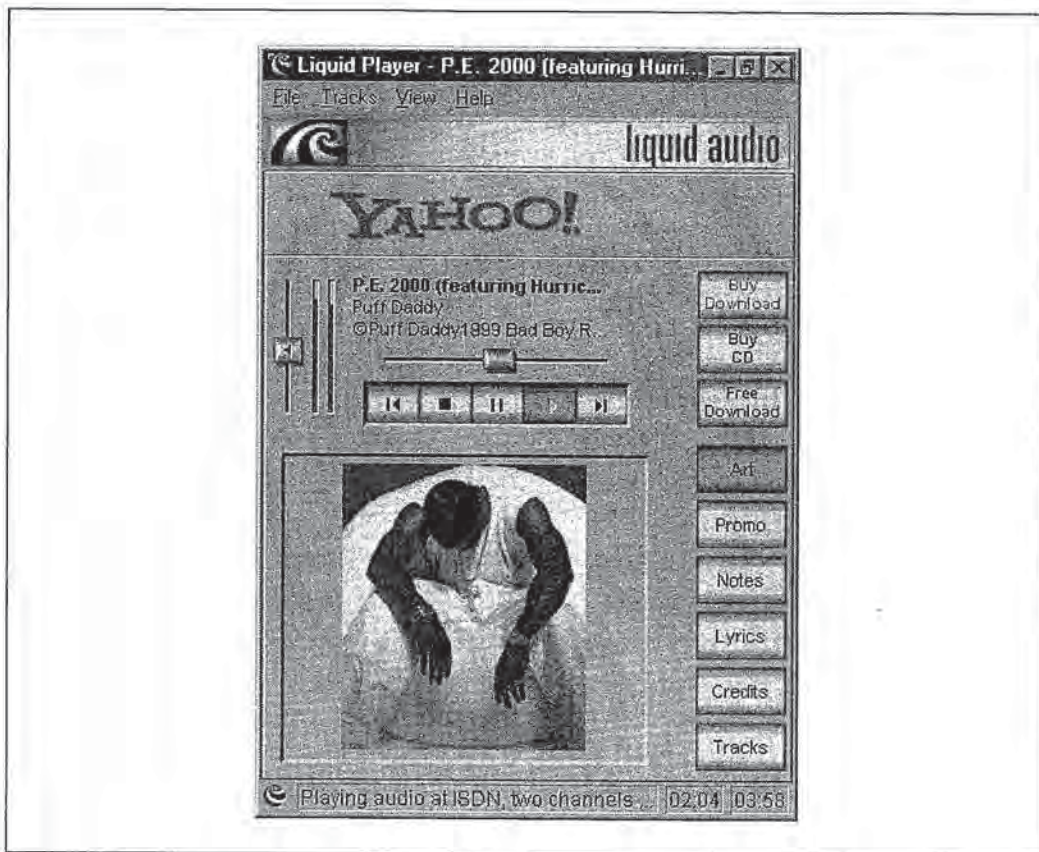


Figure 9-7. The Liquid Player offers complete access to e-commerce technology



Even though you can download the Liquid Player for free, you're still asked to enter your credit card information in order to receive a "Liquid Passport," a digital ID that gets you into their database so they can verify your future purchases and, of course, track the music you're listening to and purchasing. Whether this scheme of forced tracking has Orwellian implications for personal privacy is a judgment each user has to make for herself.

Partially because Liquid Audio got a huge head start in their part of the biz, they've got the de facto standard thing going for them, and fairly wide adoption by artists and labels looking for secure distribution (Figure 9-8). The company claims that more than 100,000 legally encoded songs are floating around out there, with quite a few big names on the roster (Dave Matthews Band and Marilyn Manson among them). Liquid Tracks also appear on a number of high-profile web sites, such as *Amazon.com* and *digital.yahoo.com*.

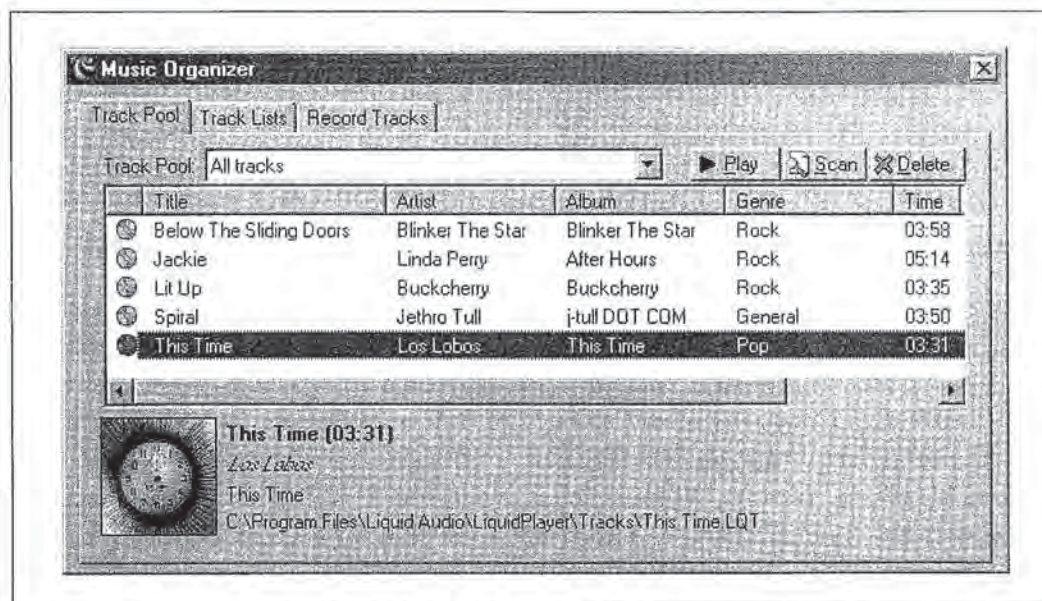


Figure 9-8. Liquid's organizer lets you organize and create playlists of your downloaded Liquid Tracks, then optionally burn them straight to audio CDs

Like QDesign (see later), Liquid is working with Texas Instruments to develop specialized chips with the Liquid Audio coder and security mechanisms built in, for use in the huge array of upcoming portable and home-based digital music playback devices. Many such devices could be in the channel on the market by the time you read this. While the Liquid Player is based originally on the AAC codec, it's important to note that the Liquid Player is not limited to playing files created with its own codec. Files created in Dolby Digital AC-3 (a format aimed chiefly at home theater and surround-sound implementations, not covered here) and AAC, Windows Media, and MP3 with the Genuine Music mark will also work in the Liquid Player. In addition, RealAudio fans can download a G2 plug-in that will let them listen to Liquid Tracks through the Real player. A variety of popular MP3 players also include Liquid support, including MusicMatch Jukebox, WinAmp, and the Diamond Rio and the Creative Labs' NOMAD (Chapter 6, *Hardware, Portables, Home Stereos, and Kits*).

The architecture of the Liquid System is constructed to support leading formats, not just to provide an added layer of convenience for users, but so that Liquid can easily be integrated with whatever mechanisms the SDMI churns out.

What Liquid Audio means for the future of MP3

Liquid is in basically the same boat as AT&T with their a2b player: excellent choice of codecs (AAC), shame about the proprietary implementation. One can't fault a company for wanting to add value to a spec and create a saleable package—it's a sound idea that used to work. But in today's climate, users want access,

openness, and interoperability, which Liquid does not provide on the same scale that MP3 does. Still, some users will appreciate the convenience of being able to burn legitimate CDs full of commercial music, and to partake in other online “value-add” functions. Again, watch for the success of the AAC codec, not Liquid Audio itself.

Global Music’s MP4

www.globalmusic.com Lest there be any confusion, this one is the poseur—a file format based on technology in MPEG-4, but proprietary. After creating the file format, Global Music One took matters into their own hands and called their files MP4 files. A smart business move, to be sure, but not one that gained them much respect among the technoscenti, since it introduces an unnecessary element of confusion into the picture. What happens when and if MPEG-4 becomes a popular implementation? If MP4 files are at all popular, they’ll probably be forced to come up with a non-intuitive name for MPEG-4 files. Urgh.

That’s not to say that Global Music One’s files don’t have their appeal. They are based on AAC, which as you read earlier in this chapter, is a high-quality format with great compression. And every MP4 file comes with its own built-in player, essentially “wrapped” around the audio data so users don’t need to have an audio player configured to work with the file format—it all just works (assuming you’re using Windows or MacOS, that is).

MP4 files encapsulated in a Digital Audio Postcard (DAP), as shown in Figure 9-9, are designed to be freely distributed by the copyright owner, and forwarded from one listener to another. The files can contain links to web sites where listeners can get more information, or purchase the album. The format received its first big boost in January 1999 when Public Enemy posted the song “Swindlers Lust” to its site for free download. Fittingly, the song is one long jab at the exploitative aspects of the recording industry.

While the format may be cool, its adoption has not exactly been rapid. Global Music posts new songs in MP4 format to its site regularly, and a few labels have jumped on board with it. But there’s very little buzz out there, and the number of songs released in MP4 format is a barely perceptible trickle compared to the number of songs appearing out there in MP3 and other formats, both legitimately and not. And because many perceive Global Music as having made a bald-faced market grab in the face of an open standard, they’ve generated a good deal of ill will in the fiercely open Internet community. Open standards guarantee a certain level of interoperability, and hence success. Closed protocols aim to corner a segment of the market for the benefit of a single company. Most of the time, closed protocols unleashed on the Internet don’t attract the biggest crowds.



Figure 9-9. Global Music's Digital Audio Postcard

What "MP4" means for the future of MP3

Global Music is onto something—until the time when we count on virtually everyone we know having MP3 playback software on their machines, Global's DAP system provides the ability to send music files to anyone without worrying in advance whether they have the right playback software installed or not. And the DAP package makes sure that artists can stay in touch with their latest creations and sales opportunities by allowing for embedded hyperlinks. But because Global Music (at this writing) does not offer an encoder that can be used by fans (Global Music works directly with labels), this version of "MP4" presents no threat to MP3.

VQF

www.yamaha-xg.com/english/xg/SoundVQ/ For a while there, the Internet went wild over the prospect of VQF. "MP3 is dead! Long live VQF!" said the fan sites. But at this writing, VQF seems to be going nowhere. Short for Transform-domain Weighted Interleave Vector Quantization, or TwinVQ for short, the format is very similar to AAC in terms of compression ratios and quality. In fact, this is no accident. Both technologies will ultimately appear in the MPEG-4 specification.

Originally developed by NTT (Nippon Telephone and Telegraph), the spec is now licensed and marketed by Yamaha, who make available one of the few VQF encoders available. VQF files typically retain near-CD-quality audio at an 18:1 compression ratio, similar to that of AAC. Because of the specification's complexity, decoders require 10–20% more CPU utilization than MP3 players, and the encoding process is much slower (though the ever-increasing speed of modern CPUs makes this kind of measurement less meaningful all the time). VQF files are known to stand up very well under conditions that stress MP3 encoding techniques. Some independent listening tests even showed that a 96 kbps VQF file had the same audio quality as a 192 kbps MP3 file, although detractors note that VQF sounds a bit thin when compared side-by-side with AAC.

It's difficult to explain the failure of VQF to catch on in a big way, except to note that it's a proprietary technology without cross-platform support. Perhaps it's not too late for VQF, but things aren't looking promising at this point.

You won't find nearly the proliferation of VQF files out there and available for download as you will MP3 files, but there are several thousand floating around. Popular MP3 players are capable of playing VQF, and curious users will find Yamaha's encoder and player kit at www.yamaha-xg.com/english/xg/SoundVQ/.

VQF files can be embedded into web pages or streamed, by means of a "locator" file similar to that used with ASF and RealAudio. A locator file consists of nothing but the URL of the .VQF file. Save the URL into a text document named with a .VQL extension and link to that instead of the .VQF itself. If you're serving .VQF files from your own server, you'll need to establish the appropriate MIME types (see Chapter 8, *Webcasting and Servers: Internet Distribution*, for more on that):

```
MIME type: audio
Subtype: x-twinvq
Extensions: .vqf, .vql
```

```
MIME type: audio
Subtype: x-twinvq-plugin
Extensions: .vqe
```

What VQF means for the future of MP3

Like AAC, VQF has a lot going for it... and is similarly hamstrung while music lovers wait for a variety of usable encoders and players to hit the scene. These, of course, are held up in politicking as the industry works out how to deal with licensing issues to keep VQF from becoming another runaway MP3. It's unfortunate that the needs of the industry and the desires of consumers are so consistently at odds. However, unlike AAC, VQF isn't even seeing the adoption into commercial products, as is true with the AAC coded in a2b and LiquidAudio. While users salivate at the opportunity to enjoy very high quality encoding at only 96 kbps, they'll continue to enjoy their MP3 files as they have been. And while VQF players and plug-in modules for WinAmp and other players were slowly beginning to increase in number, the single VQF encoder available—from Yamaha—had not been updated from its trial version of mid-1999.

It's not too late for VQF to take off, but the companies who hold its strings will need to act decisively before the format is written off in user's minds for good. At this writing, VQF does not present a viable threat to MP3.

While many of the file formats and codecs competing with MP3 for attention offer specific advantages, none have come close to usurping MP3's broad adoption, as shown in Table 9-1. MPEG-4 and QuickTime are not included here; their status as umbrella architectures makes comparison to straight codecs problematic.

Table 9-1. MP3 Compared To Other File Formats

File Format	Codec Used	Source Code Available	Strengths	Weaknesses	Security	OS Support
MP3	MPEG-1, Layer III	Yes (but licensing fees apply)	Widest interoperability between apps and operating systems. ID3 tags ^a	Not the smallest, not the best sounding	Virtually none, though ID3v2 allows for possible implementations	Windows, MacOS, Linux, BeOS, others
MP2	MPEG-1, Layer II	Yes (but licensing fees apply)	Fast encoding, consumes few CPU cycles	Larger file sizes	Virtually none (watermarks possible)	Windows, MacOS, Linux, BeOS, others
RealAudio (G2)	Proprietary (Multiple internal codecs)	No	Multiple OS support, small files, leader in streaming	Few apps support playback, proprietary	AT&T	Windows, MacOS, Linux, BeOS, others
a2b	Modified AAC	No	Small files, great quality, secure	Requires a2b player, few users	CryptoLib	Windows, MacOS
LiquidAudio	Modified AAC	No	Small files, great quality, secure, integrated sales features	Comparatively few users	Proprietary (Liquid Platinum)	Windows, MacOS
Global Music Online's "MP4"	Modified AAC	No	Small files, great quality, player is built in	Comparatively few users, snubbed industry with "MP4" name grab	Proprietary	Windows, MacOS
Windows Media	Proprietary (Windows Media Technologies 4.0)	No	Small files, good quality, security optional	Windows only unless adopted as ISO standard	Proprietary (Digital Rights Management)	Windows

Table 9-1. MP3 Compared To Other File Formats (continued)

File Format	Codec Used	Source Code Available	Strengths	Weaknesses	Security	OS Support
VQF	VQF	No	Small files, great quality, security possible	Few encoders or players. Industry still working out details	Can be added by vendor	Windows, others likely
AAC	AAC	No (likely in the future)	Small files, great quality, security possible	Mostly built into proprietary products, few "pure" encoders or players	Can be added by vendor	Windows, others likely
QDesign	Proprietary	No	Extremely small files, included in QuickTime 4.0	Proprietary, slow to take off (though signs are good)	Can use QuickTime security features	Any format with full QuickTime support

^a Although ID3 tags can theoretically be supported by other formats, it's a question of implementation rather than limitations in other formats.

Conducting Codec Comparison Tests

If you're sincerely interested in finding out for yourself which of the available codecs actually produces the highest quality output, there are a number of factors and considerations you'll want to bear in mind. First of all, it can be very difficult not to compare apples and oranges. Some codecs are optimized for quality, others for file size. Most of them attempt to strike a balance between the two, but only you can say what's most important for your collection. If you're looking for compatibility with a wide range of hardware devices, you may be willing to make some sacrifices in order to have access to the widest range of files "out there." If you've got gobs of disk space to spare and don't plan on doing much Internet distribution, you may want to go with a format like MP2 rather than the more popular MP3. If you want to run an Internet radio station, you may decide that RealAudio has more built-in technology than MP3 or Microsoft's competing ASF.

All of that said, when it's time to actually start encoding samples, take care to choose a variety of types of music. In particular, seek out music which is difficult for some encoders to get right. If you've got a favorite song and you've noticed repeatable glitches when listening to MP3 encodings of it, don't *avoid* that track—*use* it! The idea is to stress-test encoders under conditions that will really show their flaws. If you use music that's very dense or very constant, the differences between encoders will be less noticeable. Seek out music that features isolated instruments, so you can hear their subtleties rather than having them masked by the other instruments in the band or orchestra. Solo vocal passages can be especially revealing, as can acoustic guitars, harpsichords, and snare drums. Orchestral music and loud or bright pop music may not be your best choices.

You'll also want to set aside sufficient time for thorough tests. If you look around on the Internet, you're liable to find a number of tests of downloadable samples encoded with various techniques. Unfortunately, some of these tests provide samples of only a few seconds length. Short clips may cover only the most dramatic pieces of music, ignoring more subtle passages where the real testing action is. Downloadable tests are interesting for informing yourself, but cannot substitute for a test run by you, with your own music, on your own system, with lots of time to sit down and listen carefully.

As described in Chapter 4, one of the best ways to listen to multiple encodings up close and personal is to decode them back to WAV format and burn them to an audio CD you can listen to on your home stereo. Unfortunately, it can be difficult to transform many of the alternative formats back to WAV without the assistance of either illegal tools or tools that pass signal through your sound card, thus introducing artifacts with an extra digital-analog conversion step.

—Continued—

Regardless, in all cases you'll be looking for the "transparency" of the given format. In other words, you want to discover the degree to which the encoder in question intrudes on the quality of the original source file. In a perfect world, the encoded file is tiny and its audio quality is indistinguishable from that of the source file. Such an encoder would be said to have total transparency.

Appendix: ID3v1 Genres

Genre identifiers in the ID3v1 specification can be referenced either by name or by identifying number (as is common with tools such as id3ren, covered in Chapter 4). Table A-1 lists the genre names and IDs commonly accepted as “standard;” officially, any other value will be considered as “Unknown.” Table A-2 lists WinAmp’s extensions to the standard, which are also filtering into wide usage.

Table A-1. ID Numbers Assigned to Common Music Categories in the ID3 Tag Specification

0 Blues	20 Alternative	40 AlternRock	60 Top 40
1 Classic Rock	21 Ska	41 Bass	61 Christian Rap
2 Country	22 Death Metal	42 Soul	62 Pop/Funk
3 Dance	23 Pranks	43 Punk	63 Jungle
4 Disco	24 Soundtrack	44 Space	64 Native American
5 Funk	25 Euro-Techno	45 Meditative	65 Cabaret
6 Grunge	26 Ambient	46 Instrumental Pop	66 New Wave
7 Hip-Hop	27 Trip-Hop	47 Instrumental Rock	67 Psychedelic
8 Jazz	28 Vocal	48 Ethnic	68 Rave
9 Metal	29 Jazz+Funk	49 Gothic	69 Showtunes
10 New Age	30 Fusion	50 Darkwave	70 Trailer
11 Oldies	31 Trance	51 Techno-Industrial	71 Lo-Fi
12 Other	32 Classical	52 Electronic	72 Tribal
13 Pop	33 Instrumental	53 Pop-Folk	73 Acid Punk
14 R&B	34 Acid	54 Eurodance	74 Acid Jazz
15 Rap	35 House	55 Dream	75 Polka
16 Reggae	36 Game	56 Southern Rock	76 Retro
17 Rock	37 Sound Clip	57 Comedy	77 Musical
18 Techno	38 Gospel	58 Cult	78 Rock & Roll
19 Industrial	39 Noise	59 Gangsta	79 Hard Rock

Table A-2. Audio Categories Specified in WinAmp's Extensions to the ID3 Standard, Commonly Supported in Other ID3-compliant MP3 Players

80 Folk	97 Chorus	114 Samba	131 Indie
81 Folk-Rock	98 Easy Listening	115 Folklore	132 BritPop
82 National Folk	99 Acoustic	116 Ballad	133 Negerpunk
83 Swing	100 Humour	117 Power Ballad	134 PolskPunk
84 Fast Fusion	101 Speech	118 Rhythmic Soul	135 Beat
85 Bebob	102 Chanson	119 Freestyle	136 Christian-GangstaRap
86 Latin	103 Opera	120 Duet	137 HeavyMetal
87 Revival	104 Chamber Music	121 Punk Rock	138 BlackMetal
88 Celtic	105 Sonata	122 Drum Solo	139 Crossover
89 Bluegrass	106 Symphony	123 A capella	140 ContemporaryChristian
90 Avantgarde	107 Booty Bass	124 Euro-House	141 ChristianRock
91 Gothic Rock	108 Primus	125 Dance Hall	142 Merengue
92 Progressive Rock	109 Porn Groove	126 Goa	143 Salsa
93 Psychedelic Rock	110 Satire	127 Drum&Bass	144 TrashMetal
94 Symphonic Rock	111 Slow Jam	128 Club-House	
95 Slow Rock	112 Club	129 Hardcore	
96 Big Band	113 Tango	130 Terror	

Glossary

AES/EBU

A high-end digital audio connection interface found almost exclusively on expensive and/or dedicated equipment. Use of AES/EBU in MP3-related situations is rare and probably overkill, though it does occur in some studio implementations.

AIFF

One of several standard uncompressed audio formats, popular primarily on Macintosh computers. AIFF is essentially uncompressed PCM audio wearing a slightly different file header.

Algorithm

A rule or series of rules and processes described in programmatic terms to achieve a conversion between formats. The MP3 codec is an algorithm utilizing mathematical descriptions and formulae to remove “unnecessary” data from uncompressed audio.

Big Five

Standard nomenclature for the five largest recording labels in the world, which together control a huge portion of the world’s music distribution industry. The labels in “The Big Five” are Warner Music, EMI, Sony Music, BMG, and Universal Music.

Bitrate

A measure of the number of bits allocated to each second in a given data stream. MP3 encoders allow users to specify a bitrate before encoding begins; higher bitrates mean larger files and higher MP3 quality.

Bitstream

A series of bits running through a device or conduit (such as an MP3 player or Internet broadcast). An MP3-compliant bitstream can be reassembled into an intelligible audio signal.

Broadcast

The practice of “pushing” MP3 over the Internet directly to an MP3 player, either via genuine streaming (i.e., from a SHOUTcast or icecast server) or via pseudo-streaming, wherein the player “pulls” a bitstream down from a standard web server. Broadcast differs from download in that the user does not explicitly save an MP3 file to her file system.

Buffer

A chunk of MP3 data stored in memory prior to the moment of playback, for the purpose of alleviating skips caused by network latency or slow/flaky storage devices (such as some CD-ROM drives.) Buffers can also be helpful for slow machines or systems that have to do a lot of processor-intensive tasks simultaneously. Buffer size can typically be adjusted by the user.

CBR

Constant Bitrate. The standard method of MP3 encoding, wherein each second of audio is allocated the same number of bits. For example, a 128 kbps MP3 file is a CBR file. Contrast with VBR.

CDDb

The Compact Disk Database. A public database accessible on the Internet at www.cddb.com, which matches unique identifying strings for hundreds of thousands of compact discs with the album name, artist name, and track listings. The CDDb enables many MP3 ripper/encoder tools to assign meaningful names and ID3 tags to tracks as they're being encoded.

CD-R, CD-RW

CD-Recordable and CD-Recordable/Writeable. These are abbreviations referring to recordable compact disc technology. CD-R drives and discs allow data to be written to a given area of the disk only once, while CD-RW devices can be recorded to multiple times. However, there are distinct drawbacks to CD-RW technology; see Chapter 5, *Ripping and Encoding: Creating MP3 Files* for details.

Codec

Short for compressor/decompressor, this term refers to any algorithm capable of compressing or decompressing a bitstream, such as MP3. The complex but efficient MP3 codec is built into both encoders and players.

DAC

Digital-to-Analog Converter. Any unit that performs the function of transforming digitally stored audio signal into traditional analog signal, which must happen at some point in the playback chain in order for the signal to be audible. Most modern sound cards include an on-board DAC, while many audiophiles prefer to use external DACs for their (typically) superior fidelity.

DAE

Digital Audio Extraction. The capability of most modern CD-ROM drives to extract digital data from an audio CD, which virtually all modern ripping/encoding tools require. Without a DAE-capable CD-ROM drive, users must trap the analog audio signal moving through the system to an uncompressed audio format, and then encode that.

DAO

Disc-At-Once. The capability of some CD-ROM burning software to let you burn discs without a two-second delay between tracks. Not generally an issue for mixed collections, but useful for live shows and for music containing audible inter-song segues.

Decoder

Any software or hardware capable of transforming an MP3 bitstream back into uncompressed signal, which may be either audible output or an uncompressed audio file on disk. Any MP3 player can be called a decoder, as can any tool that turns MP3 files into WAV or AIFF files.

DSP

Digital Signal Processor. A programmable chip dedicated to a specific task, such as decoding video or MP3 on-the-fly. Most portable MP3 units, for example, contain custom DSPs programmed solely for the task of turning MP3 bitstreams into audible signal.

Dynamic range

An audio term referring to range of volume changes in a given audio stream. A musical piece that has both quiet and loud passages would be said to have a lot of dynamic range.

Encoder

For our purposes, any tool capable of transforming uncompressed audio signal into an MP3-compliant bitstream, which may be either a live broadcast or an MP3 file.

Equalization

The process of adjusting the audible character of an audio signal by raising or lowering frequency bands. The purpose of equalization is to compensate for deficiencies in the source material, the playback hardware, or the listening

environment, although equalizers are often used simply to pump up the bass or other frequencies some listeners find attractive. Equalization can be easily over-used, resulting in poor fidelity.

Filterbanks

During the encoding process, the signal is broken down into multiple "sub-bands," or cross-sections of the signal that can be treated separately by the encoder for optimal results. These cross-sections are known as "filterbanks," or "transforms."

Flash memory

Any silicon chip that can have data written to it in blocks, rather than one byte at a time, and that retains the data it stores even when a power supply is removed. Flash memory is commonly used with portable MP3 players to provide reprogrammable storage space in a compact form factor. Users load up flash memory cards (such as a SanDisk MMCs or Sony MemorySticks) with MP3s and take them along when they leave the house or office, so they have access to a potentially unlimited supply of music throughout the day.

Fletcher-Munson curve

A specific configuration of an equalizer that 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. Boosting these frequencies somewhat above normal can help music to sound more natural at lower volumes (see Chapter 4, *Playlists, Tags, and Skins: MP3 Options*).

Frame

MP3 files consist of thousands of very short sections, much as a movie consists of thousands of tiny still shots. Each frame is stuffed with as many bits as are allocated in the encoder's bitrate preferences, and is preceded by header data describing additional information about the data to come (encoding method, ownership, etc.) See Chapter 2, *How MP3 Works: Inside the Codec*, for details.

Frequency (Hz)

The technical description of audio pitch, measured in cycles per second, or Hertz (Hz). In general, humans cannot hear frequencies below 20Hz (20 cycles per second), nor above 20KHz (20,000 cycles per second).

FTP

File Transfer Protocol. One of the simplest (and oldest) methods of distributing data on the Internet. Many MP3 files are distributed through FTP servers rather than via the Web. While most web browsers are capable of accessing FTP servers, you'll probably want to use a dedicated FTP client if you end up downloading many files this way.

Huffman coding

A method of reducing the size of any given data set by seeking out redundancies and describing them mathematically. Huffman coding is used in virtually every computer compression technique, including ZIP and JPEG. MP3 encoders make a Huffman pass for added compression after running through the psychoacoustic techniques outlined in Chapter 2.

ID3

MP3 files allow users to store "meta-data" in the file itself in the form of "ID3 tags." There are two versions of ID3. ID3v1 stores only artist, album, song, genre, year, and a brief comment, and appears at the end of the file. ID3v2 allows for the storage of up to 256 MB of arbitrary data, and appears (more usefully) at the beginning of the file. See Chapter 4 for more information.

ISO

International Standards Organization. This group exists to codify and create formal standards out of defacto standards created by independent organizations working either solo or in concert. ISO has formally codified the MP3 work done by Fraunhofer, AT&T, and others.

Jacking

A colloquial term referring to the process of extracting an audio signal from the computer before it heads out for the sound card and toward the speakers. "Jacking" is commonly used to circumvent piracy protection schemes.

Kbps

Kilobits-per-second. A measure of the bitrate of an MP3 stream in terms of thousands of bits per second. 128 kbps is the defacto standard bitrate for MP3 audio downloaded from the Internet, but is considered sub-par from a fidelity standpoint. Most serious MP3 users encode at 160 or 192 kbps.

Lossiness

Compression techniques are either lossless or lossy. In lossless compression, the act of decompression restores every bit of the original source material. In lossy compression, some data is irretrievably discarded in the compression process. MP3 is a lossy compression format.

M3U

In on-demand streaming situations, URLs to MP3 files are listed in a simple text file. The M3U file is downloaded and passed to the MP3 player, which takes responsibility for downloading the MP3 files at the specified URLs. M3U stands for MPEG URL.

Masking

The portion of the MP3 encoding process that analyzes signal for subtle differences in pitch or notes close together in time. Because the human auditory

perception system is only sensitive to such changes if they're sufficiently separated in time, some of the data describing them can be discarded. Auditory and temporal masking processes are explained in Chapter 2.

MIME

Multipurpose Internet Mail Extension. The Internet-standard technique used for mapping filename extensions to filetypes, so that browsers can launch downloaded files into appropriate helper applications. For example, the MIME type of an MP3 files is *audio/x-mpeg*, which is associated with the extension *MP3*. It is up to the operating system or browser on the receiving end to determine what to do with any given MIME type.

Moore's Law

Gordon Moore, who helped to invent microprocessors and was once the chairman of Intel Corporation, noted that the number of transistors that could be squeezed into a given space effectively doubled every 18 months to two years, in a pattern dating back to the very invention of computing devices. As a result, processors made two years from now will be either half the size of today's at the same speed, or the same size and twice as fast. Additionally, the next cycle always lets you buy twice the horsepower at the same cost, or the same horsepower at half the cost, in comparison to the current batch of processors. Moore's Law has successfully predicted the evolution of computing horsepower since around 1900, with no end in sight (until we run up against the lower size limit imposed by the width of atoms themselves, but that's another story). See *The Age of Spiritual Machines*, Ray Kurzweil, Viking Penguin 1999.

MPEG

While this technically stands for "Motion Picture Experts Group," the term is almost always used to refer to a family of audio and video compression technologies. The family is broken down into major classes and, within those classes, "layers" defining specific strengths, bandwidths, and samplerates. See Chapter 1, *The Nuts and Bolts of MP3* for more information.

MP3

An abbreviation for MPEG-1, Layer III, a subclass of the MPEG family of audio and video compression technologies. Also the filename extension typically used with audio files of this class, e.g., *Grateful_Dead-Black_Star.mp3*.

PCM

Pulse Code Modulation. The standard format for raw, uncompressed signal, audio in particular. Each PCM sample is represented by eight bits, for a total of 64 kbps. Data on an audio CD is stored in PCM format, for example.

Perl

Practical Extraction and Report Language. A popular, cross-platform scripting language which excels at text handling and processing. Available for every operating system under the sun, Perl is the most common language used in CGI (Common Gateway Interface) back-ends for web servers, and is often used for building custom MP3 solutions for home networks, broadcast interfaces, and for building indexes of MP3 files on a given system.

Playlist

A text file referencing one or more paths to MP3 files on a system. Playlists allow users to create custom playback sequences on their hard drives without physically rearranging files. Playlists can typically be created by dragging MP3 files or directories into a player's playlist editor window, by running system queries, or by using a specialized playlist creation tool such as Helium for Windows.

Plug-in

A software module designed to extend the functionality of some other software. Many popular MP3 players accept plug-ins, and have spawned entire cottage industries of plug-in development. Plug-ins generally fall into four categories: Visualizers, DSP/Effects, Input/Output, and General Purpose/Miscellaneous. See Chapter 4 for details on specific plug-ins.

Pseudo-streaming

A streaming technique wherein the client (user's) MP3 player downloads a file directly to itself, rather than being handed a file that's already been downloaded. The process differs slightly from true MP3 broadcast. Also called "MP3-on-demand." More information on streaming techniques can be found in Chapters 4 and 8.

Psychoacoustics

The study of the relationship between the human hearing mechanism and the brain. MP3's tremendous compression capabilities are a direct result of findings originating in the field of psychoacoustics.

Relay server

An intermediary server in a broadcast system that intercepts an MP3 broadcast and redistributes it to multiple users, thereby making more bandwidth available to the source streamer so multiple users can tune in simultaneously.

RIAA

The Recording Industry of America. A trade group representing the interests of artists and labels, and involved in an ongoing battle to stamp out music piracy. The RIAA is also actively investigating methods to accommodate the new era of file-based digital music distribution.

Ripping

The process of extracting audio files from audio CDs, using a computer. Because audio CDs don't contain a filesystem similar to that of consumer operating systems, audio data must be accessed via other means. Most modern MP3 encoders do their ripping transparently in the background, although one can also use separate ripping tools. See Chapter 5 for details.

Samplerate

A measure of how many times per second a given audio signal is digitally represented (whether it be "live" or stored on disk). Typically described in Kilo-hertz, "samplerate" should not be confused with "bitrate," which refers to the amount of storage dedicated to each second of signal, rather than the frequency with which samples are taken from the source material.

SDMI

The Strategic Digital Music Initiative. An initiative proposed by many corners of the recording industry to fight music piracy at both the hardware and software levels, by creating a security specification for audio CDs, sound cards, and music playback software. By getting all links in the chain to speak the same language, the hope of the industry is that piracy will become more difficult. The proposal, however, has many detractors, and its ultimate acceptance faces many obstacles. More on SDMI can be found in Chapter 7, *The Not-So-Fine-Print: Legal Bits and Pieces*.

Signal

A somewhat abstract term. Technically, signal could be described as "a fluctuating AC voltage, current, or electrical field strength representing the part of the music, tone, audio, etc., that you want, distinct from the part that you don't want, which is noise." Noise, however, is an equally abstract term, and can refer to line hum, audio byproducts from recording/playback equipment, or even portions of a recording the artist feels detracts from the musical experience. Variations in signal represent coded information, such as music. The terms "signal" and "noise" are interdependent, defining one another, and are also context-dependent: they may mean different things in different situations.

Skin

A collection of bitmap images that can be assembled together and used as a "dressing" for the user interface of software, such as an MP3 player. Many popular MP3 players for multiple operating systems accept WinAmp skins, while other players define their own skin requirements. Skins have no effect on playback performance; their purpose is merely aesthetic.

Sound stage

The aural illusion of a sense of "space" in music, even when heard through two speakers. The "perfect" sound stage is of course perceived at a live,

unamplified concert, although quality recordings and playback equipment are able to recreate much of the "presence" of an actual sound stage. Low bitrate MP3 encodings have difficulty projecting a quality sound stage.

S/PDIF

Sony/Phillips Digital Interface. A digital interconnection schema yielding higher quality than the plastic "Toslink" optical connectors found on some equipment. S/PDIF connectors are found on many modern sound cards, and can be used with a digital amplifier or outboard DAC to obtain maximum fidelity from computer sound equipment.

SSFDC

Solid State Floppy Disk Card. An industry specification providing flash memory cards with a set of common mechanisms and interfaces to ensure interoperability across devices. For example, an SSFDC memory card will work as easily in your portable MP3 player as it does in your digital camera or your cell phone.

Streaming

Any technique wherein MP3 (or other audio/video formats) is played back on the client machine as it's being downloaded, rather than after download is complete. The two main types of streaming are "broadcast," where the server is preprocessing the stream to meet bandwidth restrictions, and "MP3-on-demand" or "pseudo-streaming," which takes place on a standard web server.

Toe-in

The technique of pointing a pair of speakers slightly in toward the center of the listening space to achieve the best possible sound stage.

Toslink

Plastic optical cable used for interconnecting some digital output devices (such as sound cards or DVD players) with some digital amplifiers and DACs. While of fairly good quality, S/PDIF is generally considered to yield superior audio quality to Toslink.

USB

Universal Serial Bus. A connection specification for low-bandwidth computer peripherals, including everything from mice and keyboards to MP3 players. USB circumvents the "IRQ dilemma" resulting from limitations in the x86 architecture. USBv2 will offer the same flexibility, but will be capable of faster throughputs, making it suitable for use with video devices as well. When shopping for MP3 hardware, you'll probably want to make sure it offers a USB connectivity option.

UUencoding

A technique for encoding binary data as ASCII data, useful for posting binaries (such as MP3 files) to Usenet groups. Most modern newsreaders are capable of decoding UUencoded data on the fly, though independent UUdecoders are available for virtually all platforms. See Chapter 3, *Getting and Playing MP3 Files* for details.

VBR

Variable Bitrate. A technique for encoding MP3 audio wherein each frame is allotted an amount of data optimized for the moment of music it represents. Passages with little musical variation can occupy very little space, while more complex passages are allotted more. Users can specify a VBR “threshold,” so they can still control whether they end up with large, high-quality files, or the opposite. Contrast with CBR.

Watermark

A digital “imprint” embedded in a music file, with the intention of preventing piracy. A watermarked file is intended to be played back only on the machine on which it was created. While it is possible to watermark MP3 files via ID3v2, no existing implementations were known at this writing. Watermarking will likely become more prevalent in nonMP3 file formats.

WAV

The defacto standard in uncompressed audio data, used extensively under Microsoft Windows. WAV files are usually straight PCM audio files with a WAV header. There is no difference in quality between PCM, AIFF, and WAV.

Wavelength

The distance between the peak of one wave and the peak of the next, measured in Hz, or cycles per second. Wavelength corresponds to pitch: The closer the peaks of the waves, the higher the pitch.

Webcasting

The process of running an “Internet radio station” by broadcasting MP3 or other audio formats over the Internet in real time. Users tune in to webcasts and hear what’s being broadcast at that moment in time (as with traditional radio), rather than having random access to a playlist, as they do with MP3-on-demand.

WMA/ASF

Windows Media Audio/Active Streaming Format. A set of Microsoft technologies for compressing audio and video signal, designed to be competitive with MP3 and RealAudio alike. These formats are described in detail in Chapter 9, *Competing Codecs and Other File Formats*.

Index

Numbers

3dsound for BeOS sound editor, 162
8hz ripper/encoder, 193

A

- a2b, 351
- AAC format, 17, 349–351
- Adaptec burning software, 204
- ADC (analog-to-digital conversion), 111
- Advanced Audio Codec (see AAC format)
- Advanced Streaming Format (see ASF files/format)
- AES/EBU connection, 213
- AHRA (Audio Home Recording Act of 1992), 264, 268–271
 - tax on blank digital audio media, 269
- AIFF files/format
 - converting, 205–208
 - obtaining from CDA files, 173
- AlbumDirect, 346
- AlsaPlayer, 71
- alt.music.mp3 Usenet group, 90
- American Society of Composers, Authors and Publishers (ASCAP), 16
- Amp3.com, 14
- Ampapod, 148
- analog sources, encoding from, 196–201
- analog vs. digital audio, 38
- analog-to-digital conversion (ADC), 111
- Apache built-in web server (Linux), 297
- Apollo MP3 player, 121
- AppleScript, 79, 128
- architectures, 330–344
- artists, impact of MP3 on, 15–17
- ASCAP (American Society of Composers, Authors and Publishers), 16
- ASF files/format, 330–337
 - creating, 333
 - linking to/streaming, 333–335
 - playing, 332
- ASX files, 334
- asymmetric connections, 296
- @soft, 76
- ATAPI specification, 175
- AT&T Glass connection, 214
- Attr2Tag utility, 131
- attributes supported by BeOS filesystem, 82
- audio
 - adding effects to signal, 144
 - analog vs. digital, 38
 - editing tools, 162
 - PCM, 162
 - types of, 45
 - uncompressed, preparing for encoding, 162
- audio data frames, 43
- audio-editing tools, 162
- audio enhancers, 113
- Audio Home Recording Act of 1992 (see AHRA)

- audio signal
 - adding effects to, 144
 - pip-ing in via sound card, 173
- audio tests, 107–110
 - checklist for, 111
- AudioCatalyst ripper/encoder, 176–177
- AudioGrabber ripper, 177
- AudioRequest home stereo player, 236
- AudioStocker plug-in, 145
- auditory masking, 28–30

B

- bandwidth
 - free, 296
 - MP3-on-demand and, 289
 - requirements for MP3 Internet
 - distribution, 294–296
 - running your own server, 296
 - upstream/downstream, 296
- batch files, generating playlists via, 125
- BeOS, viii, 87
 - applications, associating with MP3s, 87
 - attributes, support for, 131
 - built-in web server, 297
 - ID3 tags, editing, 131–133
 - attributes, indexes of, 132
 - MediaPlayer, 83
 - MP3 players, 79–87
 - skins, sharing, 84
 - system resources, checking, 156
- BIFS (Binary Format for Scenes), 340
- Big Five, 21, 346
- binary mode, 288
- binary usenet groups, downloading
 - MP3s, 4
- bit reservoir, 172
- Bitcasting, 290
- bitrate fields, 42
- bitrates, 31–33
 - encoding, which to use for, 110
 - file size and, 31
 - recommended, 161
 - threshold/tolerance, 32
- BladeEnc encoder, 171, 178
- blank disks, 204
- broadcasts (see MP3-on-demand; webcasting)
- Brujo MP3 CD player, 240
- buffering, 151

- buffers, settings for, 151
- built-in web servers, 297
- burning software, 203
- byte offsets, 166, 173
- byte reservoir, 35

C

- car players, 241–250
- Casady & Greene, 74
- Castlewood Systems, 99
- CBR (constant bitrate), 32
- CD burners, 203
- CD distribution of MP3s, 14
- CD players for MP3s, 240
- CD recorders, 203
- CDA (Compact Disc Audio), 173
 - vs. data CDs, 202
- .cda handlers, 93
- cdda2wav ripper, 184
- cdda-fs driver, 189
- CDDB (Compact Disk Database), 57, 167
- cddbblinkd daemon, 190
- CDFS.VXD file, 178
- cdindex tool, 168
- cdparanoia ripper/encoder, 183–184
- CDPlayer, 191
- CD-R devices, 201, 203
- CD-ROM drives
 - accuracy of, 159
 - DAE-capable, 40, 173
 - digital audio extraction and, 174
 - recordable, 99
 - tips for purchasing, 175
- CD-RW devices, 203
- CDs
 - compared to hard disks for storage, 202
 - creating your own, 201–208
 - burning, software for, 203
 - format conversion for, 205–208
 - databases of, 167
 - for distribution of MP3s, 14
 - vs. MiniDiscs, 208
 - recorders for, 203
 - ripping from, 173–175
 - unique identifying string, extracting
 - from, 166
 - writable, 201
- cell phone with MP3 playback, 230
- checksum, 43

- Chiariglione, Leonardo, 12
 - Claesson, Leif, 144
 - CL-Amp, 84–86
 - classes, 8
 - codecs, 7–9, 25, 171, 345–362
 - comparative table, 358
 - comparative tests, running, 361
 - installing for SHOUTcast system, 308
 - MPEG-1 Layer II, 346–347
 - perceptual (see perceptual codecs)
 - coding, 7
 - collections of MP3s
 - generating lists of, 125
 - master playlist, generating, 125
 - organizing, 94–99
 - colocation options, 296
 - comments, storing with MP3 files, 6, 23
 - Compact Disc Audio (see CDA)
 - compact discs (see CDs)
 - Compact Disk Database (see CDDB)
 - compression formats, 19, 27–28
 - computer
 - AHRA and, 264
 - connecting to home stereo, 102
 - external devices, connecting to, 211
 - home stereo, connecting to, 210
 - piping audio signal through, 173
 - remote control system, adding to, 242
 - constant bitrate (CBR), 32
 - convergence, 104, 148
 - CoolEdit 2000, 162, 199
 - CoolEdit Pro sound editor, 162, 199
 - copyright, 262, 268–283
 - (see also legal issues)
 - copyright bit, 43
 - copyright encoding option, 170
 - CPU speed, checking for upgrade, 156
 - CPU/resource meters, 154–156
 - CPUs, required for encoding/decoding, 39
 - cradles, 215
 - CRC checks, 174
 - Creative Labs, 212, 223
 - Crossfading plug-in, 143
 - cross-fading (see fadeout)
- D**
- D'Music CD player, 241
 - DACs (digital-to-analog converters), 100, 110
 - DAE-capable CD-ROM drives, 40
 - DAE (Digital Audio Extraction), 40, 174
 - DAM (Digital Audio Music System), 290
 - Dart Pro 32 noise reducer, 200
 - data frames (see frames)
 - data streams, locking onto, 43
 - database, virtual, 116–133
 - db subdirectory (RealJukebox), 56
 - dead air, removing, 172
 - decibels, 28
 - Deck sound editor, 162
 - decoders (see MP3 players)
 - decoding, 8
 - vs. playback, 143
 - de-emphasis encoding option, 170
 - Diamond Rio portable player, 209, 219–222
 - digital audio
 - vs. analog, 38
 - file transfers, legal issues and, 261–268
 - tax on blank media, 269
 - Digital Audio Extraction (see DAE)
 - Digital Audio Music System (DAM), 290
 - digital coax cable, 213
 - digital connections, 212, 214
 - Digital Media Association, The (DiMA), 273
 - Digital Millennium Copyright Act (DMCA), 272
 - digital readout, 51
 - digital signal processing chips (DSPs), 99
 - DigitalDJ, 145
 - digital-to-analog conversion, 212
 - digital-to-analog converter (see DACs)
 - DiMA (The Digital Media Association), 273
 - directory structures, building logical, 95–97
 - DirectSound, 53, 143
 - DirectSound3D, 65
 - disk space
 - checking for upgrade, 158
 - required to store MP3s, 97–99
 - disk speed, checking for upgrade, 158
 - DiskWriter, 143
 - DJ tools, 145
 - DLL (dynamically linked libraries), 141
 - DMCA (Digital Millennium Copyright Act), 272
 - DOS shell, closing, 126
 - download managers, 91
 - downloads vs. streaming, 6
 - downsampling, 294, 303

- downstream bandwidth, 296
- DSP plug-ins, 144–146, 307–309
- DSPs (digital signal processing chips), 231
- dual-channel encoding option, 43, 169
- DVDs
 - capacity for storing music, 3
 - ripping from, 196
- dynamically linked libraries (DLLs), 141

E

- EEF (Electronic Freedom Foundation), 21
- effects, adding to audio signal, 144
- effects plug-ins, 144–146
- eGo device, 218
- 8hz ripper/encoder, 193
- Electronic Freedom Foundation (EEF), 21
- empeg unit, 244–249
- emphasis bit, 43
- emphasis encoding option, 170
- emphasis field, 43
- emusic, 70
- EMusic.com, distributing MP3s via, 291
- encoders, 2
 - checking CD databases with, 168
 - finding the best, 171
 - freeware, legal issues and, 277
 - joint stereo option, 36
 - speed vs. quality, 164
- encoding, 8, 93, 160–172
 - from analog sources, 196–201
 - as single-pass process, 160
 - as two-pass process, 26, 161
 - options for, 168–170
 - with RealJukebox, 56
 - steps taken prior to, 161
 - decreasing samplerates, 163
 - test encodings, automating, 108
 - tools for, 175–193
 - types of compared, 163
 - utilities for, 93
 - your own MP3s, 92
- encoding bitrate related to file size, 98
- Enlarged Mode (Sonique), 59
- EQ button, 51, 112
- EQ presets, 114
- equalization, 112–115
- equalizers
 - built-in, 115
 - graphic, 51
 - improving MP3 quality, 172

- equipment, 99–103
- Escient, 168
- external devices, connecting to
 - computer, 211
- extraction, 193

F

- fadeout
 - BeOS attributes, attaching to, 144
 - with Crossfading plug-in, 143
 - with SoundPlay MP3 player, 82
- fair use (copyright law), 263
- fidelity, 163
- file formats
 - comparative table, 358
 - other, 329–362
- file size
 - bitrate and, 31
 - impact on MP3 development, 10
 - reduced through Huffman coding, 39
 - related to encoding bitrate, 98
- File Types panel (Windows Explorer), 67
- FileMaker Pro, 128
- filenames, 94
- files, splitting large, 201
- FinalScratch, 145
- Fletcher-Munson curve, 114
- floating-point CPU extension, 164
- folder structures, building logical, 95–97
- fps (frames per second), 45
- frames, 23, 26
 - calculating size of, 45
 - header, 41–44
- frames per second (fps), 45
- Franke, Norman, 78
- Frankel, Justin, 143
- Fraunhofer, 2, 34, 182
 - codec-based encoders, 171
 - licensing fees, 277
 - MP3 patent, 276
- free bandwidth, 296
- FreeAmp, 70
- freedb, 168
- frequencies, 24
- FTP access, 287
- FTP (file transfer protocol)
 - obtaining MP3s via, 4, 89–91
 - running your own site, 300

G

GarageBand.com, distributing MP3s
 via, 294
Garfinkel, Simson, 297
genre identifiers (ID3 specification), 363
GIF compression format, 28
Global Music, 356
glossary, 365–374
GOGO, 186
Goldenhawk burning software, 204
GoldWave sound editor, 162, 199
GoodNoise (see EMusic.com)
grabbing streams, 187
graphic equalizer, 51
graphical frequency spectrum analyzers, 51
Grip interface, 186
Groove Mechanic noise reducer, 200
Guillemot ISIS sound card, 212
guilt sites, 14

H

hand-held computers, 230–233
Handspring Visor, 7, 230–233
hard disks
 burning CDs, making space for, 205
 compared to CDs for storage, 202
 ripping directly to, 194–196
hardware
 for playing MP3s, 7, 209–260
 requirements for creating CDs, 203
 upgrades, checking for, 156–159
header files, CBR and, 33
header frames, 41–44
 setting bits for ownership attributes, 170
headers, 23
headphones, 102
hearing, 24
 thresholds of, testing, 35
Hearon, Patrick, 127
Helium utility, 122–125
Hertz (Hz), 24
high-frequency spatialization, 36
home bit, 43
home stereo
 connecting to computer, 102, 210
 MP3 players for, 233–241
 choosing the best, 234–236
 MP3s, playing through, 210–214

HTML file, outputting playlists to, 148
Huffman coding, 26, 38
 reducing file size through, 39
Hum software, 230
human voice
 frequency range of, 24
 recording, 98
HyCD ripper/encoder, 176

I

IBM
 AlbumDirect, 346
 MicroDrive, 218
icecast system, 6
 playlists, using with, 148
 samplerates and, 163
 webcasting, 319–322
ID bits, 42
ID3 data, storing during encoding
 process, 23
ID3 specification, 44, 116
 ID numbers, 363
 resources for further information, 119
 tag storage space in, 117
ID3 tags, 6, 44, 116–119
 creating from data on CDs, 167
 editing, 122–133
 in BeOS, 131–133
 in Linux, 128–131
 in MacOS, 127
 in Windows, 122–127
 fields, separators between, 125
 storing relative volumes in, 146
 upgrading in older MP3 files, 118
id3attr utility, 133
id3ren utility, 128
ID3v1 tags, 44, 116
ID3v2 tags, 44, 116
 adding lyrics to, 167
 security mechanisms and, 170
IFPI (International Federation of
 Phonographic Industries), 9
IIS built-in web server (Windows NT), 297
I-Jam portable player, 226
in-dash players, 244–250
InnoGear, 230–233
input plug-ins, 142
intellectual copyright (see legal issues)

International Federation of Phonographic Industries (IFPI), 9

International Lyrics Server, 167

international servers, 265

International Standards Organization (see ISO)

Internet

- broadcasting MP3s over (see MP3-on-demand; webcasting)
- MP3s, obtaining via, 88
- radio stations (see Internet radio stations)
- sites (see URLs)
- temporary storage on, 287

Internet distribution, 284–323

- bandwidth requirements, 294–296
- via commercial sites, 290–294
- from ISPs, 285–288
- webcasting (see webcasting)
- (see also MP3 streams)

Internet Explorer

- default handler for MP3s, 67
- setting preferred player, 153

Internet radio stations, 6

- playlists, storing in, 51
- (see also webcasting)

Internet Service Providers (see ISPs)

Internet Underground Music Archive (IUMA), distributing MP3s via, 291

in-trunk players, 244–250

Iomega drives, 99

IP address

- dynamic, dealing with, 299
- static, obtaining from ISP, 297

IP multicasting, 323

iQ audio enhancer, 113

IRC channels, downloading MP3s, 4

Irman, 148, 242

ISO (International Standards Organization), 2, 34

ISO source code-based encoders, 171

ISO specification (ISO 11172-3), 34

ISO standard, VBR files and, 32

ISPs (Internet Service Providers)

- colocation options, 296
- legal issues and, 286
- MP3 distribution, 285–288

IUMA (Internet Underground Music Archive), distributing MP3s via, 291

J

Jaz drive, 99

joint stereo effect, 36

joint stereo encoding option, 37, 43, 169

JPEG compression format, 27

Jupiter Communications, 11

K

Kernel Toys, 154

Kilohertz (kHz), 163

kit players, 250–260

KizunoMP ripper/encoder, 193

K-Jöfol, 58

KonaKoder ripper/encoder, 193

L

LaFoe, Andy, 71

LAME, 171, 185

Lash, Bruce, 292–294

law, MP3s and (see legal issues)

layer encoding option, 169

layers, 8

legal issues, 9–19, 261–283

- bogus permissions, 265
- copyright, 262, 268–283
- digital audio file transfers, 261–268
- downloading MP3s, 3
- international considerations and, 275
- ISPs and, 286
- link-related, 266
- MP3 patent, 276–278
- obtaining MP3s, 86
- webcasting, 302
- (see also piracy)

L-Emen resource meter, 154

licensing fees, 277

line-in jack, 173

- encoding from, 197–198

LineRipper encoder, 198

Linux, viii, 297

- ID3 tags, editing, 128–131
- MP3 players, 68–73
- system resources, checking, 155

Liquid Audio, 14, 353–356

Liquid Player, 353

listening tests (see audio tests)

liveCaster program, 323

liveice streamer, 319

- keyboard shortcuts, 321
- running, 320

- locking onto data streams, 43
- Löfqvist, Claes, 84
- look ahead buffer, 41
- lossless compression format, 27
- lossy compression format, 23, 27
- LP Ripper splitter/encoder, 201
- LPs, encoding from, 196–201
- Lydstrøm, 238
- Lyra portable player, 224
- lyrics, searching for, 167

M

- .M3U files, 120, 288
- MACAST, 76–78
 - ID3 tags, editing, 127
 - WinAmp skins, using, 139
- MACAST Lite, 76–78
- MacOS, viii–79
 - ID3 tags, editing, 127
 - MP3 players, 73–79
 - setting MP3 file associations in, 77
 - system resources, checking, 155
- MacPower Peripherals, 241
- Madison Project, 346
- MagicGate, 13
- mailing list, mp3stereo, 251
- manual track separation, 199
- Marty, Hugues, 155
- masking effects, 28–30
- Master List (SoundJam MP), 75
- master playlist of MP3s, generating, 125
- MBone, 323
- Media Player (Windows), 65, 330
- MediaPede, 191
- MediaPlayer (BeOS), 83
- memory, 40
 - checking for upgrade, 157
 - modules, 217
 - portable MP3 players, 215–219
- Memory Sticks, 217
- meta-data (see ID2/ID3 tags)
- MicroDrive, 218
- Micronis, 231
- micropayments, 13
- Microsoft Windows, vii
 - playlists, adding files to, 121
 - (see also entries at Windows)
- MIME types, establishing, 298
- MiniDiscs vs. CDs, 208
- MiniJam player, 230–233
- MMC (Multi Media Card), 217
- mode field, 43
- mono encoding option, 43, 169
- Moore's law, 104
- Moving Picture Experts Group, 7
 - (see also MPEG audio compression)
- Mozart's Music Box CD player, 241
- .mp3 filename extension, 2
- MP Shuttle II car player, 249
- MP2, 346–347
- MP3 AI utility, 132
- MP3 broadcast (see MP3-on-demand; webcasting)
- MP3 Cutter tool, 172
- MP3 files (see MP3s)
- MP3 Flashlight, 132
- MP3 players, 2, 5
 - additional functions, accessing, 52
 - benchmarking, 153–156
 - on BeOS platform, 79–87
 - CD players, 240
 - choosing, 47–52
 - control buttons, 49–50
 - equalizers, 172
 - built-in, 115
 - formats played, 48
 - home stereo, 233–241
 - choosing the best, 234–236
 - types of, 234
 - interface customization, 49
 - on Linux platform, 68–73
 - on MacOS platform, 73–79
 - MP3 streams, associating with, 152
 - networking capabilities, 257
 - older, problems with, 117
 - ownership attributes, displaying, 170
 - performance, improving, 153–159
 - portable, 214–229
 - comparative table, 226–229
 - memory/storage, 215–219
 - preferred, setting, 152
 - resizing, 51
 - title bar, 52
 - using, 47–52
 - VBR files and, 32
 - on Windows 95/98/NT platforms, 52–67
 - (see also players)
- MP3 Radio Station In A Box, 290
- MP3 streams, 88, 288–290
 - configuration for, 152

- MP3 streams (*continued*)
 - listening to, 148–153
 - playback, improving, 151
 - players, associating with, 152
 - saving, 318
 - (see also webcasting)
- MP3 technology, 1–21
 - bitrate, 31
 - competing formats, 17
 - future trends
 - a2b and, 352
 - AAC and, 351
 - MP4 and, 357
 - MPEG-1 Layer II and, 347
 - MPEG-4 and, 342
 - QDesign and, 349
 - QuickTime and, 338
 - RealAudio and, 344, 355
 - VQF and, 358
 - Windows Media and, 336
 - how it works, 2, 22–46
 - implementation freedom, 34
 - ISO specification, 34
 - making illegal, 15
 - MP3 patent, 276–278
 - other uses for, 17
 - playlists (see playlists)
- mp3asm utility, 115, 172
- MP3.com, 16, 322–328
 - distributing MP3s via, 290
 - MP3s, downloading, 3
- MP3ENC encoder, 179
- MP3Fix utility, 152
- MP3LiT.com, 17
- MP3-on-demand, 148–150, 285
 - bandwidth considerations, 289
- MP3s (MP3 files)
 - AIFF files, converting to, 205–208
 - analyzing, 78
 - attributes, attaching in BeOS, 133
 - audio file formats, competing, 329–362
 - BeOS applications, associating with, 87
 - broadcasting (see MP3-on-demand; webcasting)
 - collections of
 - generating lists of, 125
 - master playlist, generating, 125
 - organizing, 94–99
 - corrupted, repairing, 115
 - creating your own, 4, 160–208
 - cutting beginning/end, 172
 - default handler in Windows,
 - configuring, 67
 - disk space required to store, 97–99
 - distributing (see Internet distribution)
 - encoding (see encoding)
 - fee-based, 13
 - filenames, 94
 - obtaining, 4, 86–93
 - older, upgrading ID3 tags in, 118
 - parts of, 41–45
 - PCM, converting to, 207
 - playing, 5, 47
 - through home stereo, 210–214
 - quality, improving in existing, 172
 - renaming singly/in batches, 124
 - samplerates, 33
 - sharing, 4
 - sorting, 125
 - storing on CDs, 201–208
 - temporary storage on Internet, 287
 - WAV files, converting to, 143, 205–208
 - (see also players; MP3 players)
- mp3stereo mailing list, 251
- mp3tool utility, 128
- MP3Trim tool, 172
- MP4 (Global Music), 356
- Mpecker encoder, 182
- MPEG audio compression, 7, 22
 - (see also lossy compression format)
- MPEG-1 Layer II, 346–347
- MPEG-1, Layer III, 2, 8
- mpeg-1/mpeg-2 encoding options, 169
- MPEG-3, nonexistent, 8
- MPEG-4, 339–343
 - creating/playing, 341
- mpg123, 68–70
- MS Audio, 331
- M/S (middle/side) mode, 37
- MSPs (Music Service Providers), 290
- Multi Media Card (MMC), 217
- multicasting, 323
- Music Library (RealJukebox), 55
- Music Service Providers (MSPs), 290
- musicians
 - impact of MP3 on, 15–17
 - MP3 distribution and, 292–294
 - self-publishing, 16

MusicMatch Jukebox, 53–54
 encoding from line-in jack, 197–198
 ID3v2 tag and, 118
MusicVac utility, 78

N

N2MP3 ripper/encoder, 182
Napster, 4
nav-state (Sonique), 58
Nelissen, Marco, 80
NET Act, 274
NetDrive, 240
Netscape Navigator, setting preferred
 player, 153
networking capabilities, 257
Night 55, 57
Nitrane decoder, 62
No Electronic Theft Act, 274
noise reduction, 200
NOMAD portable player, 223–225
normalization, 145, 165
NullSoft, 61
 SHOUTcast system, 306
numbering tracks, 96

O

OK Uncook utility, 115
Oliphant, Mike, 186
open source
 music, 19
 server system, 319–322
OpenMG, 13
Orb drive, 99
original encoding option, 170
output plug-ins, 142
ownership attributes, 170

P

padding bits, 42
patch cord, 210
PCM (Pulse Code Modulation), 30, 173
 MP3s, converting to, 207
PDAs (Personal Digital Assistants), MP3s
 and, 230, 232
Peak for the Macintosh sound editor, 162
Pebbles sound editor, 162
perceptual codecs, 22–41
 tasks in, 26

Performance Monitor resource meter
 (Windows NT), 154

Perl

 creating playlists with, 127
 modules, resources for further
 information, 258

Pernick, Ari, 133

Personal Digital Assistants (see PDAs)

Personal Jukebox, 215

personal use, 264

Pine, 241

piracy, 9

 downloading pirated music, 3
 shutting down pirate Internet sites, 11
 (see also legal issues)

PitchFork plug-in, 144

PL button, 51

playback, 5, 47

 through cell phone, 230
 vs. decoding, 143
 devices/hardware for, 7
 hardware considerations and, 156–159
 through home stereo, 210–214
 improving, 151
 joint stereo option and, 37
 normalizing, 145
 software (see MP3 players)
 time, 120, 218
 (see also MP3 players)

players

 car, 241–250
 CD, 240
 hand-held computer, 230–233
 home stereo (see MP3 players, home
 stereo)
 kit, 250–260
 portable (see MP3 players, portable)
 (see also MP3 players)

playlist window/editor, 5

playlists, 5, 51, 116–119

 button on MP3 player, 51

 creating

 custom, 125

 using Perl, 127

 editing, 122–133

 files, adding to from Windows, 121

 formats, 120

 HTML, 122

 HTML file, outputting to, 148

playlists (*continued*)
 loading/manipulating, 121
 master list, 125
 non-MP3 formats, including with, 120
 operating systems and, 119
 randomizing, 50
 PlayMedia Systems, 62
 playrate, controlling, 51
 .pls files, 120
 plug-ins, 139–148
 accessing, 141
 obtaining, 140
 remote control, 148
 types of, 140
 polyphonus passages, 39
 PoorMan built-in web server (BeOS), 297
 portable MP3 data CDs, 218
 portable players (see MP3 players, portable)
 POSIX command-line tools, 48
 post-masking, 30
 pre-encoding optimizations, 161
 pre-masking, 30
 private bit, 43
 private encoding option, 170
 Process Spy resource meter (MacOS), 155
 Process Watcher resource meter (MacOS), 155
 processor speed, 39
 ProcessWatcher (BeOS), 156
 protection bits, 42
 protection encoding option, 170
 pseudo-streaming (see MP3-on-demand)
 Psycho 1/Psycho 2 encoding options, 169
 psychoacoustics, 23, 25
 resources for further information, 31
 waveforms and, 24
 public servers, 317
 Pulse application, 156
 Pulse Code Modulation (see PCM)

Q

QDesign, 348
 QDMC (QDesign's Music Codec), 348
 QSound, 113
 quality
 achieving optimum, 161–162
 of encoders, 164
 improving in existing MP3s, 172
 sound, 106–111

QuickTime, 337–338
 MPEG-4 and, 341
 QDMC codec and, 348
 Release 4, 74

R

RAM, 40
 randomizing playlists, 50
 RAW files/format, 172
 RCA cable, 210
 Real, 55
 real streaming, 150
 RealAudio format, 56, 343
 RealJukebox, 55
 playlists, proprietary format for, 56
 RealJukebox Gold, 232
 recordable CD-ROMs, 99
 Recording Industry Association of America (see RIAA)
 recording industry, impact of MP3 on, 9–14
 re-encoding sources, 305
 reference tracks, 107
 ReGet download manager, 91
 remote control, 148
 remote control system, adding to
 computer, 242
 renaming MP3s, 124
 ReQuest, 236
 reservoir (see byte reservoir)
 resource meters, 154–156
 resources for further information
 BeOS-based stereo components, 254
 convergence, 148
 copyright, 263
 legal issues, 262, 283
 mp3stereo mailing list, 251
 multicasting, 323
 Perl modules, 258
 players, 259
 psychoacoustics, 31
 SDMI, 283
 RIAA (Recording Industry Association of America), 9, 209
 rights (see legal issues)
 Rio portable player, 219–222
 RioPort, 219
 RipChord ripper/encoder, 193
 RipEnc (BeOS), 192
 RipEnc (Linux), 189

- ripping, 4, 173–175
 - from CDs, 173–175
 - from DVD, 196
 - from other sources, 193–201
 - direct to hard disk, 194–196
 - encoding from analog sources, 196–201
 - with RealJukebox, 56
 - tools for, 175–193
 - on BeOS platform, 189–193
 - on Linux platform, 183–189
 - on MacOS platform, 180–183
 - on Windows platform, 176–180
 - utilities for, 93
- royalties, paying, 14
- S**
- samplerates, 33, 163
- SCMS protection, 264, 269, 278
- scripts
 - MP3 files, searching through, 130
 - playlists, generating via, 125
 - using with SoundApp, 79
- SCSI specifications, 175
- SDMI (Secure Digital Music Initiative), 12, 278–283
- SDMI-compatible players, 216
- security
 - hackers and, 336
 - ownership attributes and, 170
 - web servers and, 297
- security checking, RealJukebox, 55
- self-publishing, 16
 - presumption of guilt, 17
- Serial Copy Management System (see SCMS protection)
- Setup Options panel (Sonique), 59
- shout streamer, 319
- SHOUTcast system, 6
 - configuring server, 309, 311–314
 - distributing other formats, 316
 - playlists, using with, 148
 - remote administration, 315
 - samplerates and, 163
 - webcasting, 306–318
- side channel, 37
- side information, 37
- signal, equalizing (see normalization)
- simultaneous masking, 28–30
- Skin Selector menu item (WinAmp), 134
- SkinConverter utility, 139
- skins, 49, 51, 133–139
 - creating your own, 135–139
 - templates for, 136
 - WinAmp
 - obtaining, 134
 - supported by SoundPlay, 82
- skipping problems, 40
- @soft, 76
- Solid State Floppy Disk Card (SSFDC), 218
- SongBank home stereo player, 238–240
- Sonique, 57–61
 - controlling/launching from command line, 60
 - hotkeys, 60
- Sony
 - memory modules, 217
 - MiniDiscs, 208
 - VAIO Music Clip Personal Network Player, 233
- sound cards, 41, 99
 - checking for upgrade, 157
 - pipng in audio signal via, 173
 - video card interaction, checking for upgrade, 159
- sound editors, 162, 199
- Sound Forge sound editor, 199
- Sound Laundry noise reducer, 200
- sound quality, 106–111
- SoundApp utility, 78
- SoundBlaster Live! sound card, 212
- SoundForge for Windows sound editor, 162
- SoundJam, 181–182
 - support for Rio, 221
- SoundJam MP, 74–76
- SoundPlay burner, 207
- SoundPlay player, 80–83
- source streamers, 305–307, 311, 319
- source streams, 303–306, 313, 319–320
- spaces in filenames, 94
- S/PDIF connection, 213
- speakers, 100–102
 - checking for upgrade, 157
- specifiers for audio frequency, 42
- splitting large files, 201
- SSFDC (Solid State Floppy Disk Card), 218
- Stardust decoder engine, 57
- static file sources, 305
- stereo encoding option, 169
- stereo, home (see home stereo)
- stereo option, 43

- storage capacity, portable MP3 players, 215–219
- streaming
 - vs. downloads, 6
 - samplerates and, 163
 - types of, 148–152
 - (see also MP streams; MP3-on-demand)
- streams, grabbing (Linux), 187
- subscriptions, 14
- Sundström, Jonas, 131
- symmetric connections, 296
- sync blocks, 41
- sync header, 45
- sysmeter resource meter, 154
- Sysmon resource meter
 - (Windows 95/98), 154
- system requirements, 156–159

T

- tag editors (see ID3 tags, editing)
- Tag2Attr utility, 131
- TagWorld utility, 132
- tape, encoding from, 196–201
 - most effective technique, 199
- tape storage units, 99
- temporal masking, 28, 30
- testing
 - audio, 107–110
 - checklist for, 111
 - codecs, 361
- Thomson Multimedia SA, 9, 182
 - MP3 patent, 276
- 3dsound for BeOS sound editor, 162
- time utility, 156
- TManager process controller (BeOS), 156
- Toast burner, format conversion, 207
- tolerance level, 23
- top utility, 155
- Toslink connection, 213
- track numbers, 96
- Tracker (BeOS), 80
 - ID3 tags, displaying in, 131
- tracks
 - manual separation, 199
 - numbering, 96
 - reference, 107
- Turtle Beach sound card, 212

U

- UI customization, 49
- UltraEncode ripper/encoder, 191
- Uncook95 utility, 115
- uncook.exe utility, 115
- underscores in filenames, 95
- unique identifying string, CDDB-ID, 166
- Unix operating systems, 48
- upstream bandwidth, 296
- URLs
 - AudioCatalyst, 176
 - AudioGrabber, 177
 - Be, Inc., 79
 - BladeEnc, 178
 - CDDB, 167
 - cdparanoia, 183
 - CL-Amp, 84
 - copyright, 263
 - empeg unit, 244
 - emusic, 70
 - Fraunhofer, 9
 - FreeAmp, 70
 - icecast, 6
 - ID3 specification, 119
 - I-Jam portable player, 226
 - Liquid Audio, 14
 - MACAST/MACAST Lite, 76
 - MP3 broadcasts, 150
 - MP3 technology
 - ISO specification, 34
 - licensing issues, 9
 - MP3.com, 16
 - MP3ENC, 179
 - MP3LiT, 17
 - MP3-related, 48
 - MP3s, downloading, 3
 - MPEG, 9
 - mpg123, 68
 - NOMAD, 223
 - playlists of, 5
 - RealJukebox, 55
 - Rio portable player, 219
 - SDMI, 278
 - SHOUTcast, 6
 - skins, 134
 - Sonique, 57
 - SoundApp, 78
 - SoundJam, 181

- URLs (continued)
 - SoundPlay, 80
 - stored in ID3v2 tags, 117
 - UltraEncode, 191
 - WinAmp, 61
 - XAudio, 73
- USB, 215
- Usenet, 92
- Usenet groups, 90
- user interfaces
 - customization, 49
 - unique, 58
- utilities for repairing MP3s, 115
- uudecoders, 92

V

- VAIO Music Clip Personal Network Player, 233
- van Zoest, Sander, 322–328
- VBR (variable bitrate), 32
 - encoding option, 170
 - files, 33
- vibrations, 24
- video cards, checking for upgrade, 159
- virtual database, building, 116–133
- virtual DJ, 144
- visualization plug-ins, 146
 - system performance and, 147
- voice memos, 216, 223, 231
- voice (see human voice)
- volumes, equalizing (see normalization)
- VQF format, 17, 357–358

W

- watermarks, 12
- WAV files/format
 - converting, 143, 205–208
 - MP3s decoded to, 161
 - obtaining from CDA files, 173
- waveforms, 24
- wavelengths, 24
- WaveOut plug-in, 142
- web servers
 - built-in, 297
 - running your own, 296–300
 - security and, 297
 - testing, 297
 - (see also webcasting)

- webcasting, 301–323
 - commercial, 303
 - from icecast server, 319–322
 - from SHOUTcast server, 306–318
 - how it works, 303–306
 - legal issues and, 273, 302
 - (see also Internet radio stations)
- wget download manager, 91
- WinAmp, 61–65
 - DSP plug-in, 307–309
 - EQ curves, associating with tracks, 114
 - format conversion, 206
 - hotkeys, 63
 - little known features, 135
 - MP3 equalizer, 114
 - playlists
 - HTML, generating, 122
 - loading/manipulating, 121
 - pure tones for testing, 146
 - Skin Browser, 63
 - skin layout, 49
 - skins, 133
- window shade mode, 51
- Windows 95/98
 - default handler for MP3s,
 - configuring, 67
 - ID3 tags, editing, 122–127
 - MP3 players, 52–67
 - system resources, checking, 154
- Windows Media Audio (see WMA files/format)
- Windows Media compression format, 19
- Windows Media Player (see Media Player)
- Windows NT
 - built-in web server, 297
 - default handler for MP3s,
 - configuring, 67
 - ID3 tags, editing, 122–127
 - MP3 players, 52–67
 - system resources, checking, 154
- WMA files/format, 68, 331, 336
- World Intellectual Property Organization, 272

X

- X Windows, players for, 70
- XAudio output plug-in, 144
- XAudio player, 73

Xeonon MP Shuttle II car player, 249

Xing

AudioCatalyst, 176–177

codecs, 165

encoders based on, 171

Xmms, 71

XWave for Linux sound editor, 162

Z

zip compression format, 2, 27

Zip drive, 99

About the Author

Scot Hacker has been obsessed with music since his early teens, and has been a digital audiophile for half a decade. After earning a B.A. in Philosophy at UC Santa Cruz, he made a living writing reviews of jazz and improvised music for *The Utne Reader* and *The Cadence Journal of Jazz and Blues* before graduating to the role of Content Manager and Production Editor at ZDNet. Hacker's interest in digital audio and fine computer systems has evolved into a series of regular articles for *PC Magazine*, *Byte.com*, *Windows Sources*, *ZiffNet*, Japan's *ASCII* magazine, and the CompuServe network, as well as television appearances and trade show gigs. Hacker is the author of the definitive *The BeOS Bible*, a best-selling guide to the high-performance operating system optimized for digital media, including audio compression, encoding, and playback.

Colophon

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The animal on the cover of *MP3: The Definitive Guide* is a hermit crab (*Coenobita perlatus*). The hermit crab is commonly found in the Indian and Pacific Oceans, and inhabits the areas surrounding the Islands of Aldabra, Mauritius, and Samoa. Despite the name of the hermit crab, which alludes to a solitary lifestyle, *Coenobita perlatus* are very social creatures. They characteristically travel in groups of 25 or more, and have been found living in colonies of over 100 in the wild.

Hermit crabs make their homes by occupying the discarded shells of gastropods in order to protect their soft, coiled abdomens and inner organs. They prefer shells that fit snugly in order to prevent evaporation of moisture. Most hermit crabs carry water in their shells, which they use for breathing and a water source when they are far away from the sea.

Maureen Dempsey was the production editor and copyeditor for *MP3: The Definitive Guide*. Colleen Gorman and Clairemarie Fisher O'Leary were the proofreaders. Melanie Wang, Nicole Arigo, and Madeleine Newell provided quality control. Brenda Miller wrote the index.

Hanna Dyer designed the cover of this book, based on a series design by Edie Freedman. The cover image is a 19th-century engraving from the Dover Pictorial Archive. Kathleen Wilson produced the cover layout with QuarkXPress 3.32 using Adobe's ITC Garamond font.

Alicia Cech designed the interior layout based on a series design by Nancy Priest. Mike Sierra implemented the design in FrameMaker 5.5.6. The text and heading fonts are ITC Garamond Light and Garamond Book. The illustrations that appear in the book were produced by Robert Romano and Rhon Porter using Macromedia FreeHand 8 and Adobe Photoshop 5. This colophon was written by Maureen Dempsey.

Whenever possible, our books use RepKover™, a durable and flexible lay-flat binding. If the page count exceeds RepKover's limit, perfect binding is used.

A DIFFERENT KIND
of Animal



O'REILLY®
web.oreilly.com

When you're stuck on the Web... Ask Someone Who Knows.™

Our readers expect a lot! They won't settle for hastily spun answers to their sticky technical questions. And when they need a quick, dependable antidote, they don't have time to work through a spidery maze of confusing information. That's why they turn to O'Reilly's Animal Guides for reliable, no-nonsense solutions to the everyday challenges of Web technology.

Our books are written by real-world experts—experienced Web designers, programmers, and administrators. They offer tried-and-true solutions to both routine and unusual problems. So the next time you're caught on the Web, turn to O'Reilly. We'll help you get untangled.

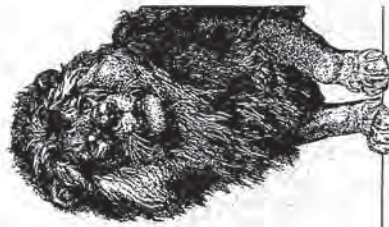
Ask your favorite bookseller about O'Reilly Web-technology books.

For more information go to
web.oreilly.com
 or call **800-998-9938**
 or 707-829-0515

THE O'REILLY® WEB SERIES



Part # 8257B © 1999 O'Reilly and Associates, Inc. O'Reilly is a registered trademark of O'Reilly and Associates, Inc.



O'REILLY

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472-9902
1-800-998-9938

Visit us online at:
<http://www.ora.com/>
orders@ora.com



O'REILLY WOULD LIKE TO HEAR FROM YOU

Which book did this card come from?

Where did you buy this book?

- | | |
|---|---|
| <input type="checkbox"/> Bookstore | <input type="checkbox"/> Computer Store |
| <input type="checkbox"/> Direct from O'Reilly | <input type="checkbox"/> Class/seminar |
| <input type="checkbox"/> Bundled with hardware/software | |
| <input type="checkbox"/> Other _____ | |

What operating system do you use?

- | | |
|--------------------------------------|--|
| <input type="checkbox"/> UNIX | <input type="checkbox"/> Macintosh |
| <input type="checkbox"/> Windows NT | <input type="checkbox"/> PC(Windows/DOS) |
| <input type="checkbox"/> Other _____ | |

What is your job description?

- | | |
|--|---|
| <input type="checkbox"/> System Administrator | <input type="checkbox"/> Programmer |
| <input type="checkbox"/> Network Administrator | <input type="checkbox"/> Educator/Teacher |
| <input type="checkbox"/> Web Developer | |
| <input type="checkbox"/> Other _____ | |

☐ Please send me O'Reilly's catalog, containing a complete listing of O'Reilly books and software.

Name _____

Company/Organization _____

Address _____

City _____

State _____

Zip/Postal Code _____

Country _____

Telephone _____

Internet or other email address (specify network) _____

Nineteenth century wood engraving
of a bear from the O'Reilly &
Associates Nutshell Handbook®
Using & Managing UUCP.



PLACE
STAMP
HERE



NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 80 SEBASTOPOL, CA

Postage will be paid by addressee

O'Reilly & Associates, Inc.

101 Morris Street

Sebastopol, CA 95472-9902



MP3: The Definitive Guide



MP3 is the most popular compressed file format for easy digital distribution of sound and music over the Internet. *MP3: The Definitive Guide* is the bible of this phenomenon. It's a fast-moving, well-written guide to the juicy bits for MP3 power users. Ripping and encoding are covered in depth, from a true audiophile's perspective. You'll learn exactly what it means to listen to a perceptual codec and how it affects the way you should record, make tests, and listen to your MP3 files. You'll find a detailed examination of the codec itself and its development, as well as how MP3 files compare with other common compression formats.

Like MP3, *The Definitive Guide* doesn't limit itself to one operating system or even one player. It contains advanced information for using MP3 files on all the most popular operating systems, including Windows, Linux, Macintosh, and BeOS. You'll find plenty of information about WinAmp, but players on other platforms are also covered in depth. You'll also find an expert's guide to MP3 files online, web sites to look for, ways to work with newsreaders and ftp, and the latest word on the rapidly evolving technologies of the MP3 audio revolution.

This book also covers:

- MP3 storage, including databases, file systems, and shared playlists. You'll learn about tagged MP3 files and working with global Internet databases to automatically download track and artist listings.
- Security and legal issues, so you can create your own MP3s and stay within the law. Or if you're just interested in the social and legal issues surrounding a file format that could make the record industry irrelevant, you'll learn about the protocol.
- Hardware options. You'll find a buyer's guide to the myriad hardware players including portables and kits. You can listen to MP3s on your PDA, car stereo, or a rack-mounted audio system.

This book will succinctly answer every question, frustration, or problem you might have while creating, using, and listening to music in MP3 format. No matter what your level of involvement with the technology, *MP3: The Definitive Guide* is your blueprint to building a technologically advanced musical library.

Visit O'Reilly on the Web at www.oreilly.com

ISBN 1-56592-661-7	US \$29.95 CAN \$43.95
Warehouse - BK16389502	
MP3: The Definitive Guide	
Used, Very Good	
(uVG)	S

