

RADIFIED

Guide to Ripping & Encoding CD Audio

This guide presents what the **gurus** of the **encoding scene** consider the *best* way to **rip & encode CD audio**. By 'gurus' I mean those who've taken the time to participate in blind **listening tests**, and painstakingly compare a variety of music, encoded with a variety of encoders, at a variety of settings (called *arguments* or *switches*). I'm talking about folks who are **serious about music**.

The word 'best' used here refers to **audio fidelity**, not ripping or encoding *speed*. In other words, this guide is more concerned with the **quality** of your encoded files, than the **time** it takes to produce them. You'll be glad to know that all the requisite **software** referenced here is **free**. That's because the **best** software in the ripping & encoding world is **free**. So put away your credit card and break out your CD collection.

Note: If you download the PDF version, which contains all **8** web-pages, yet none of the ads **coded into** the online version, I encourage you to consider making a small contribution, since you can't click ads you don't see. (These clicks help pay our hosting bill.) Many sites which offer a PDF version **charge a nominal fee** for the download. If the spirit moves you, a small *Donate* button can be found on the **Home page**.

I also encourage you **share** the guide with any friend you feel might find it helpful. Word-of-mouth is still the most effective form of advertising, and helps multiply our digital karma. Lastly, I want to extend an invitation .. to stop by **the site** every few days and see what's happening here in **Rad-land**. I promise, you'll find the content (if nothing else) **original**. Enough entreaty. Let's get busy..

Introduction

If you've ever downloaded MP3s from the original **Napster** [now **shut down** by the courts], or one of the **other** file-sharing services, and found that those songs sounded like krap, it's because the people who encoded those MP3s **didn't know** the ripping & encoding **mojo** you'll learn here. [No, 128-kbps is **not** CD-quality.]

This guide will walk you through the steps of configuring **Exact Audio Copy** [EAC], the world's **best** CD audio ripper. Actually, I'll refer you to my buddy from **Belgium**, **SatCP** for that, or my buddy from **Calgary**, **Chris Myden**. No sense in **reinventing the wheel**.

With EAC's vaunted **secure mode**, we'll **rip** your **favorite songs** from your **favorite CDs**, and **encode** those digital songs as high-quality **MP3s**, using **LAME**, the world's **best** MP3 encoder .. or **compress** them as exact **digital clones** using **Monkey's Audio**, a popular Windows-based **lossless** audio compressor.

Along the way, you'll be introduced to some **alternative [non-MP3] encoders** that might suit your needs better .. such as **Musepack's** sub-band codec [also known as **MPC**, **MP+** and MPEG-Plus], and the new **Open Source** encoder: **Ogg Vorbis**. I will detail the pro's & con's of each codec, so you can **decide for yourself** whether one of these upstart encoders can do a better job for you than the nearly **ubiquitous MP3** format.

You'll also be introduced to some promising **new technologies** which have recently made a splash in the encoding pond, such as [the proprietary] **Advanced Audio Coding** [AAC], which is being developed as part of the new **MPEG-4** spec as a joint venture between companies such as Dolby, Fraunhofer (FhG), AT&T, Sony, and Nokia. [MP3 is part of the older, **MPEG-1** spec.]

This guide identifies both the strengths & weaknesses of each format, giving you the information you need to make intelligent encoding decisions .. based on your own specific needs .. such as ..

- .. amount of disk space you have available ..
- .. number of CDs you own ..
- .. how good your hearing is..
- .. **quality** of your audio hardware ..
- .. type of music you listen to ..
- .. **format compatibility** requirements ..
- .. file **restrictions**, etc.

Only *you* know what is right for you. Encoding is not a *one-size-fits-all* proposition .. not any more.

You'll also see a side-by-side [comparison of various audio codecs](#) in action, using a variety of popular settings [sometimes called *switches* or *arguments*]. I even share **my own** personal encoding preferences, and discuss the reasoning behind my approach.

With your freshly-encoded files, you'll be able to **create** custom [playlists](#), and **listen** to your pristine-sounding songs directly from your [hard drive](#) [in **any order** you choose], or **burn** them as custom CD-Rs, containing a **compilation** of your favorite songs, to play on your [home stereo system](#), in your [car](#), or perhaps even give to a [friend](#) as a personalized birthday present.

With the [right software](#), you can even **cross-fade** the songs on your compilation CDs, so the next song starts playing as the current one fades out. This adds a touch of artistic class to your CDs you won't find in any store-bought discs [which all contain a standard 2-second gap between each song].

Copyright Laws in the Digital Age

I probably should mention that the **copyright laws** which pertain to the ripping & encoding of legally-purchased CD audio music [for your own personal listening pleasure] **vary from country to country**. Most countries have what are known as [Fair Use laws](#), which allow you to make a back-up copy (in case your original gets scratched) and share your music with [friends](#) .. long as you don't profit from it.

Things become blurred when we try to apply *Fair Use* laws to digital media. [Digital technology](#) allows us to make *perfect* copies. This was not possible prior to the advent of the [digital age](#).

Before bits and bytes came along, copies of music recorded on audio cassettes [which is an [analog](#) format, not digital], and movies recorded on VHS tapes [also an *analog* format], experienced what is known as *generational loss*. In other words, the quality of the copy is **inferior** to that of the original. And the quality of each *successive* copy [generation] deteriorates further.

[Digital copies](#) experience no such *generational loss*. Rather, the quality of your *digital* copies are **identical** to that of the original. In other words, the millionth copy sounds just as sweet as the one you bought at the store. This is one of the things that [concerns the Recording industry](#) about digital technology [and for good reason].

The rapid growth and widespread use of [broadband](#) technology allows an ever-increasing number of [enthusiasts](#) to [transfer](#) encoded/compressed versions of digital media around [the world](#) in a matter of minutes.

[Here in the US](#), these laws are a jumbled [mess](#) of counter-intuitive confusion. No one seems to know for sure what the law [does and doesn't allow](#). It often [depends who you ask](#), and the answer [keeps changing](#). If you're interested, a good place to start understanding why US copyright laws are so confusing is Jessica Litman's book: [Digital Copyright](#).

If you are unsure of [your country's](#) stance regarding the legality of ripping & encoding legally-purchased CD audio music, [here's the [disclaimer](#), folks] you should **seek legal advice** from a qualified professional before using this guide.

As you might imagine, I receive much mail on this topic, most of it from people with an [axe to grind](#). In no way do I condone [piracy](#). Even the most bandwidth-happy among us agree that [artists](#) deserve to be paid. We simply want the [freedom](#) to listen to our music in whatever manner we choose. With those unpleasanties out of the way, let's move on to the good stuff.

Note that this guide does *not* address how to **rip or encode DVD video**. If that's what you're looking for, check out [Doom9](#), [Nicky Pages](#), [divx-digest](#), [divx.com](#) or [dvdripping](#). I'm sure you'll find what you need at one of those fine sites.

For your [<hypertext>](#) convenience, this guide can be found at any of these fine Radified URLs:

- [<http://mp3.radified.com/>]
- [http://mp3.radified.com/CDripper_MP3encoder.htm]
- [http://radified.com/MP3/CDripper_MP3encoder.htm]

It has become surprisingly popular since being discovered by search engines. Try searching for the terms **Ripping CD audio** or **MP3 Encoding** with either [Google](#) or [Yahoo](#) and you'll see what I mean. Of all the [Radified guides](#), this one has been the most difficult to keep current. Links tend to [rot](#)

faster here than in any other guide, because things are still evolving in the field of audio encoding.

For an idea of what folks are saying about this guide, check out [kurth.org](#), or [HotMP3gear](#), or [CADTutor](#), [listed near the bottom, under the question *Where can I find out more?*]. Before we begin tho, I want to mention <shameless plug> a few other **Radified guides** you might find helpful:

- The [[Guide to Norton Ghost](#)] is currently the site's **most requested** feature.
- The [[ASPI drivers guide](#)] is ranked **#2**, and has been **translated into** more languages than any other feature.
- [[Doc's FDISK Partitioning Guide](#)] is referenced by several Universities and even a hard drive manufacturer.
- Doc's guide comes with a **companion**: [[Hard drive Partitioning Strategies](#)] that shares insights into the best way to divvy up your new hard drive.
- Magoo's [[BitTorrent guide](#)] teaches you the mojo on how to get that hot new file-sharing program up-n-running on your system.
- Perhaps you know somebody who needs help with [[PC Maintenance](#)].
- The [[Windows XP Installation guide](#)] has become very popular.
- And [[several others](#)]. </shameless plug>

Okay. Let's get busy. This guide contains **8** pages and **9** steps to audio-encoding nirvana, organized like so:

- [[Page 1](#)] - Introduction [you are here].
- [[Page 2](#)] - Steps **1 & 2** [**download** the *Exact Audio Copy* CD ripper & the *LAME* MP3 encoder].
- [[Page 3](#)] - Step **3** [download *Monkey's Audio* & glean insights into **lossless** audio compression, including how it contrasts with the more-popular lossy codecs].
- [[Page 4](#)] - Side-by-side **comparison** of different audio codecs. Helps you see the big picture in a small snapshot.
- [[Page 5](#)] - Step **4** [Razorlame front-end, ABR vs VCBR vs BR, configuring LAME with switches and arguments, ID3 tags].
- [[Page 6](#)] - Steps **5-9** [r3mix, SatCP tutorials, **configuring EAC** ripper & **LAME** MP3 encoder].
- [[Page 7](#)] - **Alternative** [non-MP3] **lossy codecs**, such as Liquid Audio, AAC, *MPEGplus*, Ogg Vorbis, Windows Media, Real audio, and others that are likely to play a role in the future of audio encoding.
- [[Page 8](#)] - **Related info**, such as MP3 players and managers, Winamp plug-ins, the best CD-ROM readers and burners, burning and labeling CDs, sound cards, headphones, and other such items you might find interesting or helpful.

By the time you finish, you'll be able to dialogue intelligently with **the gurus** of the encoding scene, and move into the finer points. The next page contains the first two steps:

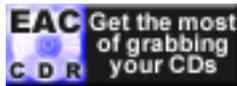
1. Download the **Exact Audio Copy CD Ripper**
2. Download the **Lame MP3 Encoder**

Fasten your seat belt. Here we go.

Quick Start

I like to understand the philosophy behind how ripping & encoding works. But I realize everyone isn't like me. If you're the *ready-fire-aim* type, who prefers to jump right in and begin ripping & encoding right away, a **Quickstart** guide is posted at [cd-rw.org](#). Or you can **try searching** for the terms: **eac ripping guide lame mp3 encoding**. I'm sure other sites have posted guides since I first published this one [in **2001**]. For those of you who'd like to delve a little deeper, step right this way...

Exact Audio Copy CD Ripper



Let's head to Germany & pay [Andre](#) a visit: Here we'll download his latest version of **Exact Audio Copy** [EAC]. The name of this software should give you a clue to understanding its function. EAC transfers songs from your CD [song_title.cda] to your hard drive [song_title.wav]. It does this better [more accurately] than any other software of its kind.

The downside of EAC is that it's not the easiest program to configure. No need to worry, tho. I have [friends](#) who can help. Once EAC is configured, you'll never have to worry about it again. And there's no reason to be concerned about the warning: "Beta Version!". I have ripped [extracted] over **500** CDs with this so-called "beta" software .. without a single problem [after configuring it correctly, of course]. [FYI: I rip with a [SCSI Plextor](#) reader, from [Windows XP Pro](#), using [Adaptec ASPI layer v4.60](#), installed with [ForceASPI](#), on an [Intel-based](#) system.]

The thing that makes **EAC** the world's *best* CD audio ripper is its **secure mode**. Far as I know [FAIK], EAC is the *only* ripper in existence that features a secure mode. If you deposit money into a "secure" bank, you're confident it will be there when you return to make a withdrawl. Likewise, a secure-mode ripper gives you confidence that the bits and bytes stored on your CDs [representing the songs] are transferred *accurately* to your hard drive.

EAC does this in a variety of methods, depending on the capabilities [features] of your particular optical drive. As you might expect, the painstaking process of ripping in secure-mode takes **longer** than does a normal single-pass ripper. Single-pass rippers operate on the philosophy: *We'll give it a shot, mate. If we get it right, great. If not, oh well, we tried.*

If there's a problem with your disc [e.g. scratch], EAC will return to the point of the problem, and do its best to accurately reproduce on your hard drive the audio information contained on your CD. The number of times it returns to the problem area [scratch] depends on how you configure the setting for "error recovery / correction quality" [low, medium or high]. In other words, you can specify how many times EAC will re-read the data stored on the CD before giving up.

Most people who are concerned with audio fidelity don't mind spending the extra time necessary to ensure the most accurate audio extraction possible. You can help speed the process by taking care of your discs. Never lend the original. Make a back-up copy soon as you return home from the store, and keep the original stashed away in a safe place. In this way, if something happens to the copy currently in use, you can simply burn another from the original, which remains in pristine condition.

EAC will rip a CD in good condition fairly quickly. It's only when the CD is beat up that it takes a long time. But this is where EAC really shines .. with CDs in poor condition. **CDex**, another popular CD audio ripper, offers what is/are known as [Paranoia settings](#). This is *not* secure-mode ripping. You can read about [Paranoia here](#). Most ripping [gurus](#) feel that **CDex** is the second best ripper, after EAC. It's certainly [very popular](#).

A comparison of the top rippers is [posted here](#) [<-conclusion: EAC is the best .. something we already know]. You should [download](#) a copy of **CDex** just for the heck of it. [It won't cost you anything, except the time it takes to download.] Its primary strength is that it's **easier to configure** than EAC. Many people use CDex as their primary ripper for this very reason. Personally, I keep both handy.

EAC can [miraculously] extract songs from **badly scratched CDs**: songs no single-pass ripper can extract. I've seen it take up to an hour to extract a single song, but the result is always a file that sounds flawless, even when errors and "suspicious locations" are indicated. ["Suspicious location" is EAC's way of saying, "*I think there's something wrong here, hoss. Do me a favor and listen to this section to see what you think. Here's where to look.*"]

Whenever a CD has a problem [usually due to scratches], EAC identifies the *exact* location, so you can listen for yourself, and determine if you're able to hear any artifacts. [*Artifact* is a term that means, "I can hear something wrong." It usually indicates a *minor* problem, one you need good-quality headphones to hear.]

For the record, I've never been able to hear any problems with songs ripped by EAC, even when the program indicates errors while extracting media from the disc. In fact, I have yet to find a single CD that EAC is unable to extract. And several of my CDs look like somebody ate [pizza](#) off of them. This is what **Matt** [who writes the code for [Monkey's Audio](#) lossless audio compressor] has to say about EAC:

Ripping: Everything starts with copying your CD's to your computer using a ripper. If perfect quality is what you want, use **EAC**. It's an amazing program, a must have for anyone serious about audio quality. It does a lot of great stuff that other rippers don't, ensuring a perfect copy. Instead of ineptly attempting to tell you why **EAC** rules, just try it yourself.

Alternative CD Audio Rippers

Feurio [US\$29, [Germany](#)] and **BlindWrite** [US\$30, Toulouse, [France](#)] are also excellent rippers. Both programs are easy to recommend. I suggest you download both demos and give them a whirl when you get the chance. Unlike EAC & CDex however, neither Feurio nor BlindWrite are free. But Jens Fangmeier, who develops Feurio, generously offers an unrestricted trial period that never expires.

A PDF version of the Feurio manual is [posted here](#). Feurio can be good to use during those times when EAC takes too long to extract audio from a badly scratched CD. While Feurio **rips** well, it *excels* as an audio **burning** program [.. tho not as slickly as the ultimate CD audio burning program: **CD Architect**, from the boys at from Sonic Foundry. CD Architect is one of my all-time favorite programs. It's right up there with **Ghost**, **Photoshop**, **Dreamweaver**, **Sound Forge**, and **Avid Xpress**]. Even tho EAC's claim-to-fame is its ability to **rip**, it will also **burn** audio CDs. EAC is the opposite of Feurio. By that I mean: Feurio rips good, but excels at burning audio CDs. Conversely, EAC burns good, but excels at ripping audio CDs.

I use **BlindWrite** regularly, to "back-up" [copy] both audio and data CDs. Like **CloneCD**, and the hot, new **Alcohol 120%**, BlindWrite is particularly effective at *copying* CDs .. rather than **digital audio extraction** [DAE]. It is especially effective at copying CDs that contain sophisticated **copy-protection** schemes. BlindWrite is actually *two* programs in one. **BlindRead** copies the data from your CD and creates an image of the CD on your hard drive [file_name.BWI], while **BlindWrite** burns the image file to disc. Very slick. Very neat.

Word on the street is that **AudioGrabber** [US\$20] plans to release a secure ripper sometime soon, perhaps with v1.90 [v1.82 is current at time of this writing]. **AG** also offers a **free version**, but it only lets you rip *half* the songs without restarting the program .. which can get annoying. AG is probably the **most popular** ripper out there right now. Ease-of-use is its primary strength. AG is produced in [Germany](#) [birthplace of the **MP3**]. Have you noticed a pattern? None of this cool, ripping and encoding software originates in the [United States](#). I also heard **great things** about **PlexTools** [which includes ripping software]. PlexTools must be used with a **Plextor drive** however. Plextor [an international company based in [Belgium](#)] manufactures the world's *best* CD drives and burners.

When addressing the **quality** of CD drives, we should distinguish **speed** from **accuracy**. Drives are so fast these days that *speed* is no longer an issue. Therefore, judgments about performance should focus primarily on *accuracy* [accurate reproduction of bits and bytes]. Ripping speed doesn't mean much if your songs are full of pops and clicks.

If you take a look at BlindWrite's list of the **World's Best Burners**, you'll see that the majority of them are made by Plextor. Unfortunately, my Plextor drive is too old to support PlexTools. PlexTools used to ship only with *European* drives, but has recently become **available for drives** shipped here in the US. Prior to this, folks here in the States would have to get an original copy of PlexTools from a **friend** in Europe if they wanted to use the Europe-only software. Once an original copy of PlexTools was installed, you could then install updates as Plextor **posted** them.

If you want to learn a *little* more about ripping [digital audio extraction], [see here](#) [Computall]. If you want to learn a *lot* more, [see here](#) [emediapro]. For more info on which programs work well for burning & copying in different situations, take a peek at my **Programs guide**. <another shameless plug>

LAME MP3 Encoder

Step 2: Let's head to **Dibrom's HydrogenAudio site** and download LAME v3.90.3. This is not the latest version, but rather the **recommended** one, because it is specifically coded to take advantage of the **--alt-presets** we want to use .. which yield the best audio fidelity per unit bit used.

Since 1998, **many programmers** have spent countless hours coding this software for you. Still others [like me] have spent even more time aiding **the development** of this codec by participating in double-blind **taste tests**. This is why the **home page** is able to say: **LAME** has the "*speed and quality to rival all commercial competitors*." The only thing I don't like about LAME is it's name. But you'll quickly discover it's not.

FYI: **Dmitry's site** always contains the *latest* version of LAME [v3.93.1 at time of this writing]. If you have trouble accessing Dmitry's site, try the mirror at **jthz.com** [which belongs to **Julius T & Hens Z**]. LAME binaries can also be downloaded from **SMPMan**. I mirror the latest version myself, on my **Downloads page**. Note again that, at this time, the *latest* version is *not* the recommended version .. for the **switches/arguments** we want to use [**--alt-preset standard** and/or **--alt-preset extreme**].

Before proceeding, check [here](#) for the latest info on the currently recommended version. The nice folks at **Hydrogen Audio** are good about keeping you posted with up-to-the-minute information. Things change fast in the encoding world, and you want to stay current.

LAME is a recursive acronym that stands for **LAME Ain't an MP3 Encoder**. The name was fashion back when the source was not an encoder. But it is

now. The acronym is fashioned after **GNU**, which stands for **GNU's Not Unix!** LAME is considered the world's best MP3 encoder by those in the know. If you have good ears, and good equipment [good **soundcard** and **headphones**], you can even help with its **development**. [-This is where the developers hang out, and they're always looking for folks with a golden ear.]

The downside to LAME is that it's not the easiest encoder to use/configure. Like me, you may have discovered that the most powerful software are usually those programs with the **steepest** learning curves, because they offer the most options. That's where this guide comes in. It'll show you how to put the LAME mojo to work for you, with a minimum of pain & suffering. My buddy, **Chris Myden**, will be glad to help you with the configuration details.

Note that the **LAME homepage** [maintained by Mark Taylor] does not contain [compiled] binaries. LAME is distributed as **source code**, which needs **compiling** [turns raw lines source code text into usable software]. You can configure EAC to launch LAME automatically, after it's done ripping your songs. In other words, EAC can call the LAME encoder and launch it to encode MP3s in the background, automatically, while you rip. Purists however, prefer to perform one step at a time, rather than ripping & encoding in tandem.

It shouldn't matter if you do both together, but this merely adds a *potential* for problems. Perhaps I should say that ripping and encoding separately has *less* potential for problems. Personally, I rip first, encode afterwards. Likewise, purists decode prior to burning, instead of decoding on-the-fly. [In reality, it shouldn't matter.] You can also use **EasyLAME** to batch encode wave files you already ripped with EAC. But I'm getting ahead of myself. We'll discuss the details of encoding later, on [Page 5](#) and [Page 6](#).

Compile your own Binaries

This is not necessary, but fun. If you'd like, you can compile your own LAME binaries. The Intel C++ Compiler is reportedly your best option for this. It's posted [here](#). You can download a 30-day **trial here** [42-MB]. **LAME** source code is [posted here](#).

Monkey's Audio Lossless Compressor

Step 3. Let's head to [Matt's site](#) and **download Monkey's Audio**. This software is subtitled: "*A fast and powerful lossless audio compressor.*" Sure, it's fast: much faster than any lossy codec you'll use. But speed ain't why we're here. Rather, it's that "lossless" part of the subtitle that most interests us. [Note: **CPU speed** affects/determines the speed of this compressor.]

The term "**lossless**" implies *no loss*. But no loss of what? > Audio fidelity. **Lossy** codecs [such as **MP3** and **Musepack**] generate smaller files by **discarding** selected bits contained in the original songs [bits it thinks you won't notice when they're gone]. Lossless codecs, on the other hand, keep *everything* .. **all** the audio information [bits] contained in the original song(s). Lossless codecs merely **shrink** the size of the original file/song by >compressing< it [usually by half]. In other words, songs compressed with lossless codecs represent **exact digital clones** of the originals [with regard to audio fidelity/quality]. Audiophiles should be drooling about now.

If you've ever used **WinZip** or **WinRAR**, you understand the concept behind how a lossless codec works. Suppose you compress a **Microsoft Word** document with WinZip. The document file [file_name.doc] will shrink to roughly **half** its original size when zipped [file_name.zip]. When you **unZip** it however, the entire document is there, right? Not a single letter is missing. Every period is present and accounted for.

That's the principle behind a **lossless** codec. They actually operate on a much simpler basis than **lossy** codecs, which must make many complicated decisions, based on the particular **psycho-acoustic** model programmed into them .. regarding which bits to keep and which to discard. Lossless codecs do one thing and one thing only: they >>**crunch**<< files. Lossless codecs which crunch [compress] files **faster** and **smaller** are considered **better**.

Note: the term **psycho-acoustics** refers to how the **mind** [psycho] interprets **sound/music** [acoustics]. Only **lossy** codecs employ psycho-acoustic models. Each lossy codec [**Fraunhofer**, **LAME**, **Xing**, **BladeEnc**, **Audioactive**, etc.] has its own idea of how to encode music so a **minimum** number of bits are used to represent **maximum** audio fidelity.

Lossless codecs don't use psycho-acoustic models. They are merely crunching machines. They don't have to agonize over which bits to discard, because they don't discard any.

Note that you can configure the **EAC** ripper to call and launch *Monkey's Audio* so it compresses CD audio straight to *.ape files directly from the CD.

Lossy Encoder v Lossless Compressor

I try to keep my terminology straight. In other words, I refer to **lossy** codecs as **encoders**, and **lossless** codecs as **compressors**. It's a subtle difference, and one you'll see ignored by some people, who use the terms interchangeably [encoder & compressor]. But lossless codecs do not "encode" per se. They merely compress [crunch]. Lossy codecs, on the other hand, do more than compress: they encode, which involves discarding selected bits deemed unnecessary. If this distinction confuses you, forget I mentioned it. It's not important. Encoders and compressors will work no matter what nasty names you call them.

Perhaps you're familiar with **digital photography**. Files compressed with a lossless codec can be compared to the **RAW** file format, which also employs **lossless** compression. **MP3**'s on the other hand can be compared to **JPEGs**, which offer various degrees of **lossy** compression. It's a known fact that the human ear has difficulty detecting a **soft** note played immediately after a **loud** one [e.g. a quiet whisper following a loud shout]. The lossy encoder will see this and discard the bits representing soft notes immediately following loud ones.

The goal here is to generate a smaller file with [hopefully] minimal loss of audio fidelity. It's also known that humans have trouble hearing sounds above **16-KHz**. This is why lossy codecs will discard all audio information above the 16-KHz threshold when they're configured to output the smallest possible files [lowest bit-rates]. Some even set the cut off at a more-aggressive **12-KHz**. [The human voice typically falls between **1-KHz** and **3-KHz**. This band represents the "sweetspot" for most encoders. It shouldn't surprise you that humans hear best in this frequency range.]

Most of the time, lossy encoders makes good decisions regarding which bits of audio information to keep and which bits to discard. But not always. Certain types of music are more difficult to encode than others. [Music that contains a lot of "transients" is particularly difficult to encode.] There are a **dozen** good lossless audio codecs out there. And there isn't a whole heck of a lot of difference between any two of them. Unlike **lossy** codecs, **lossless** codecs don't vary much in their speed or out-putted file size. If they're all so similar, then why did I select Monkey's as my lossless-codec-of-choice? [Note that songs compressed with *Monkey's Audio* receive a ***.ape** file extension: song_name.ape]

Monkey's has been around a while. At the time of this writing, Matt is working on **v3.98**. Compare that with **FLAC** [Free Lossless Audio Codec: my second-favorite choice], which is currently at **v1.10**. I concede that this doesn't necessarily mean Monkey's is better than FLAC, but it's a valid point of comparison.

Sure, Monkey's **compares favorably** to other lossless codecs, with regard to **speed** and **compression** ratios. But what really convinced me to use Monkey's is **its front end**. Not all other lossless codecs come with their own front-end. The front end is **the interface** you use to configure the codec in order to compress and decompress your songs. Monkey's also comes with a neat **tagger**.

Monkey's biggest weakness is that it supports **Windows-only**. Obviously, this is not a problem if you use nothing but Windows. But if you only use **Linux**, Monkey's ain't for you. [Everybody ought to **install** a copy of Linux on their system (**multi-boot**) and familiarize themselves with it. I prefer **Mandrake** myself.] If you need support for Linux, hardware or streaming, I recommend **FLAC**. The folks who hang out at the **Hydrogen Audio forums** use FLAC as their lossless codec of choice, because it allows everyone to participate in listening tests, regardless of which operating system they use.

Transparency and Archive Quality

The **ideal** job of a lossy encoder is what's called **transparency**. That's when the encoded version sounds **no different** from the original. Altho I've never seen a formal definition, the term **archive quality** usually refers to a *high-quality* file that contains minimal or negligible audible differences from the original file. Whereas **transparency** means, "I hear no difference," **Archive quality** means, "I can hear a difference, but it's not a problem." Are we splitting hairs? Yes. Are we picking nits? You betcha. People get mighty serious about audio quality. Head over to the **Hydrogen Audio forums** and watch them pick nits. [You'll learn a lot in the process.]

Because some people have better hearing [ears] & better equipment [hardware] than others [I have good equipment, but not the best ears], **transparency** is defined differently by different people. Factors that affect transparency can include:

- Ears [hearing ability]
- Equipment [headphones, sound card, amplifier, speakers, etc.]
- Type of music [**classical**, **rock**, **jazz**, **trance**]
- The individual **song**
- The particular **encoder**
- Bit-rate

Better **ears** and better **equipment** make it *more difficult* to achieve transparency. The more difficult it is to achieve transparency, the *more bits* you'll need to use. All things being equal [same encoder, same song, same ears]: more bits = more audio information = higher bitrate = bigger files = better audio fidelity/quality. In other words, if your hearing sucks, and your speakers suck, or your headphones suck, you're not likely to notice a difference in quality between the original song and an MP3 encoded to a low bit-rate with a **crappy encoder**. [Altho it's *fast*, the **Xing MP3 encoder** is generally considered the world's **crappiest** by those who have conducted extensive listening tests.]

Hearing deteriorates with **age**, especially the ability to hear the higher frequencies. Good audio equipment, and especially a set of **good cans**, is the best way to discriminate subtle nuances in audio quality. If you want to participate in some of these listening tests, a good set of headphones is a must. FYI: I use the **Sennheiser HD-580** headphones. They feel like slippers on your ears. I also use the [24-bit] **CardDeluxe** soundcard [made by *Digital Audio Labs*]. It is highly-rated at the benchmark comparison test **posted here** [scroll down].

I also use the **baby** model **VLZ series Mackie mixer**. [Wish I would've spent the extra money and got the next-better model: the **1402**, which has sliders instead of knobs.] I have some **Event studio monitors**, but I don't use them to participate in listening tests. Lossless codecs represent the **holy grail** of audio archiving, as far as **fidelity** is concerned. Files produced by a lossless codec are **transparent** by definition .. because there is no audible difference between a file compressed with a lossless codec and the original.

Downside of Lossless Codecs

"If lossless codecs are so good," you might be wondering, "why isn't everyone using them? Why do people even bother with MP3s and lossy codecs?" Good question. The **bad** thing about lossless codecs is that they offer **poor file-size reduction** .. compared to MP3s and lossy codecs.

In other words, a song compressed with Monkey's Audio or other lossless codecs will consume **more hard drive space** than the same song encoded with an MP3 or other lossy codec. It will also take **longer** to send this same file to your friend(s) over the Internet .. which can become a major factor if you still have a **dial-up connection**. Lossless codecs typically reduces file sizes to **half** their original bulk. The exact amount depends on a number of factors, such as the type of music and the specific song. Live music doesn't compress well because of the background noise. Each song will be **different**.

High-quality MP3's typically require bit-rates of **175-** to **225-kbps**. Let's use **200-kbps** as a happy median for purposes of making a comparison. Wave files from the original CDs are **1411-kbps**. This number is calculated like so: **44.1 KHz * 2 channels * 16 bits = 1411.2-kbps** [or nearly **10-MBytes/min**]. Note that this calculation assumes k=1000-bytes. Most computers assume k=1024 bytes. So if your actual sizes differ slightly from what you calculate, this is why. Also the "b" in **kbps** = "bits" [not **Bytes**].

A **200-kbps** MP3 file is ~**14%** the size [about 1/7th] of the original wave file [**1411-kbps**]. In other words, lossless files [typically 600-900 kbps] will be roughly **3 to 4 times larger** than MP3s [consuming 3 to 4 times the hard disk space], but **half** as large as the original wave file ripped from the CD. You should now have a good feel for the pro's and con's of how lossy codecs compare with lossless compressors.

The Trade Off

Decisions regarding encoding and compression involve balancing **3** factors:

1. Audio **fidelity**/quality
2. File **size** [affects hard drive space and Internet transfer time]
3. **Time** [to encode/compress]

Time-to-encode/compress is typically the **least** important factor listed. Many audio-quality fanatics "batch-encode" **overnight**, while they sleep. They wake with a bunch of new MP3 on their hard drive. Overnight batch-encoding eliminates *time-to-encode* as a factor for consideration. You will find that most encoding decisions boil down to a trade off between **file size** and **audio quality**. Everybody seeks maximum fidelity with minimum file sizes [bit-rate]. You should read that last sentence again, because it represents the crux of the encoding issue.

This guide [and most audio enthusiasts] are more concerned with maximizing audio **fidelity** .. than minimizing files size [bit-rates] .. especially since hard drive space became so cheap. For example, you can purchase a **120-GB drive** [Western Digital Caviar] for **~US\$119**. [**Newegg** also offers attractive prices on hard drives.]

A **120-gig** drive will store over **300** CDs encoded with lossless compression. That's **~40-cents** per CD. You could even get a **250-GB drive**, and prices continue to drop. The popularity of lossless compression has been **growing** with the hard drive capacities. An **ideal** encoder would yield perfect transparency at the tiniest bit-rate possible, and it would encode very quickly. Unfortunately, such an encoder does not [yet] exist. Today, if you want **quality audio**, you have to **pay with bits** [requires more hard drive space].

All things being equal, more bits = greater audio fidelity. Only you can decide how much audio quality you want/need, and how much you're willing to pay for this quality in terms of file sizes. All encoders are *not* created equal. Better encoders produce files with higher audio fidelity using less bits [lower bit-rates]. Transparency at the lowest possible bit-rate is what everyone is after. As we mentioned before, LAME is the best **MP3** encoder at achieving this delicate balance. It yields files with the best audio fidelity at reasonable bit-rates.

Personally, I compress all my **favorite** songs with Monkey's [-1/3rd], and encode everything else with a lossy codec [-2/3rds]. If you're anything like me, you don't rip *entire* CDs. Rather, you rip only your **favorite songs**. This means you can typically store the songs from over **600** CDs on a single, **120-GB** hard drive. [Some CDs have only a few good songs.]

Monkey's Audio vs Shorten

SatCP of The Coaster Factory has this to say about lossless compression:

As the name implies, lossless compression reduces the size of the files without any corresponding loss in quality. After decoding, you have exactly the same file as you started with. The drawback is that the compression ratio is not very high [50-60%]. Trading groups that distribute music over the Internet almost always demand lossless compression for file distribution.

Monkey's Audio:

1. Fastest and best compression with the default settings (though compression is only slightly better than Shorten).
2. Support for ID3 tags.

Both Shorten & Monkey's are excellent lossless compressors, but Monkey's definitely wins. On the other hand, Shorten is very popular on trading lists.

When the MP3 wars rage about which encoder & arguments are best, you'll be glad to know that - no matter how good lossy encoders get - they'll never surpass the quality of lossless audio compression.

Direct Comparisons of Audio Codecs

This table will give you an idea of how different audio codecs (both lossless & lossy) compare with one another, using a variety of popular settings (called *arguments*). Results are arranged from highest to lowest bit-rate (kbps). Results will differ for different songs and CPUs, but this table gives you an idea of the **relative** capabilities of each encoder/compressor, and what the various settings offer.

Note that I do not ascribe any **quality** value to the results listed below. Quality is **subjective**. It depends on a number of factors, such as the particular song, equipment used (headphones, speakers, sound card), and most importantly, the ears of the individual listener. The empirical data listed on this page is *not* subjective. I leave that up to you, the listener, to judge for yourself.

Hardware

- **CPU:** Intel **P3-700 @938MHz** (overclocked)

Note: 03aug2002 - While these tests were run with the stated CPU, I have since upgraded my system to a **P4-1.6a Northwood CPU**, running at **2138 MHz** (Asus P4T-E mobo). This CPU allowed me to get **\$600** performance (at the time) for **\$135**. I'm very happy with its **stability**.

- **RAM:** 512MB **Mushkin** rev 1.5 PC133, 2-2-2 timings
- **CD-ROM drive:** **Plextor** UW SCSI PX-40TW (firmware 1.05); supports C2 error correction
- **Hard drive:** 36GB **Seagate Cheetah**
- **SCSI adapter:** **Tekram DC-390U3W**

Software

- Windows 2000 SP2
- **Exact Audio Copy** v0.9b3 (Secure mode with external **ASPI interface**)
- **Razorlame** v1.1.5.1342 (for encoding with LAME)
- **WavPack** lossless compressor v3.93 (Monkey's used as front end)
- **FLAC (Free Lossless Audio Codec)** v1.02 (Monkey's used as front end)
- **Monkey's Audio** v3.95
- **MPEG Plus** (aka MPC and MP+) v0.90u (Speek's front end)
- **Speek's MPC Batch encoder** v 2.10
- **LAME (LAME Ain't an MP3 Encoder)** v3.91 (Razorlame front end)
- **Ogg Vorbis RC3** (Monkey's used as front end)

Audio File

Positive Vibration (5:48), recorded by **Bob Marley & The Wailers**, from **Babylon By Bus** album. Bit-rate calculations: Encoded/compressed file (MBytes) x 1024 KBytes/MByte x 8 bits/Byte / 348 secs = **kbps**. If you want numbers that reflect true **bits** (not **kbits**), then multiply my results by 1.024. See [here](#) for the table of results. The PDF will not allow me create that many cells.

Comments

Time to encode is generally considered the *least* important encoding variable, because you can always batch encode overnight, while you sleep, and wake with a bunch of freshly encoded files on your hard drive. Most encoding decisions involve a **trade off** between **file size** and **audio quality**. But audio quality is *subjective*. It depends on a number of variables which only you can judge for yourself.

My Preferences

The good thing is that you're not limited to a single codec or setting. I take an approach that **uses a combination**. I compress all my **favorite** songs with **Monkey's** (lossless). I have to **really** like the songs, because songs compressed with a lossless codec consume a lot of hard drive space .. at least twice as much as those encoded with a lossy codec.

You probably have some songs that are so special to you that you don't want to discard a single bit. Bob Marley's *Redemption Song* is like that for me. I would never encode this song with a lossy codec .. even if I couldn't hear the difference. When I play it back, I want to know that I'm hearing **all** of it. The **majority** of my songs get encoded with **MPC** with the '**xtreme**' preset. This setting is fast and offers great quality. MPC is the best codec to use for bit-rates above 160kbps.

Actually, the *standard* setting is excellent, but I don't want to be nagged by the question of "should I have...", so I opt for the larger file sizes of *xtreme* (~200kbps vs 175 for standard). The *insane* setting throws bits at acoustics that only dogs & bats can hear. To me, *that* is insane. Next let's take a look at Holger's **Razorlame**, a popular front-end for the LAME command-line encoder.

RazorLame: A LAME Front-end



Step 4: Go to Holger's site [\[Germany\]](#) and **download** the latest version of RazorLame. LAME is a command-line program. In other words, it has no graphical user interface, which means it's difficult for most people to use. RazorLame is a GUI for LAME, also called a Front End. RazorLame makes it easy to configure & use the LAME MP3 encoder. There are other front ends for LAME, such as **LAMER**, but I have no experience with anything but RazorLame.

You can use the RazorLame/LAME combo as a stand-alone encoder. You can configure LAME using RazorLame to the specs you want, then **copy-n-paste** the switches for LAME directly into EAC. This may be easier than typing in the settings manually, where it would be easier to make a mistake [type-o]. The only problem is that the setting for ABR [average bit rate] is not easy to find. I finally asked Holger. He said:

Yes, that's currently a bit hidden. For v1.1.4 beta use 'Edit' from the Main menu, then 'LAME Options', choose the 'VBR' Tab, check 'enable VBR', and finally check 'use ABR instead of VBR' at the bottom of the dialog and enter the target bitrate. I'm aware that this isn't optimal and planning a change in the future.

Regards, Holger

You can also copy-n-paste pre-configured settings into the 'Custom options' from the Expert tab in order to configure ABR encoding. Don't forget to put a check in the box labeled Use only custom options. Copying-and-pasting these Lame switches directly into EAC (or Razurlame) is more fool-proof than typing the switches by hand. If you type an error into the command line (for example, - b vs -b) the Lame DOS-looking box will open then close immediately (provided you have have you system configured to show the Lame's DOS-looking box).

You can also go to jthz.com and copy the settings that he has already configured for you. Again, it's easiest to copy-n-paste switch configurations. If you want to learn a little more about LAME's switches, they're [posted here](#). Some people call switches "arguments".

LAME Settings

People who know the most about maximizing MP3 audio fidelity for your bit-rate buck use the following setting with [LAME 3.90.3](#):

--alt-preset standard [yields files ~190-kbps, depending on the specific song, and the type of music]

This is a VBR setting, which means it'll take longer to encode than an ABR file. Personally, I have tons of hard drive space, so I'm not very concerned with file sizes. If you don't mind spending ~250-kbps, you can use:

--alt-preset xtreme

In order to use any of these presets however, you will need to use v3.90 or 3.90.3. [[download here](#)]. These presets are not programmed versions prior to 3.90. Versions after 3.90.3 are not tuned as well for these alt presets. After participating in a blind listening test, I am now using "--alt-preset xtreme" as my switch of choice. In that test, I could definitely hear a difference between three of the files and the others. Those three (best-sounding) files were:

```
MPC -insane -nmt 99 -tns99 +lowpass 19.5 (700+ kbps)
--alt-preset xtreme (232kbps)
--alt-preset standard (222kbps)
All other files sounding noticeably inferior to these three.
```

The alt presets use VBR encoding. If you want to know what each of these presets represent, here it is:

```
Standard Preset --nspstune -V2 -mj -h --lowpass 19.5 -b112 --nssafejoint --athtype 4 --ns-sfb21 3
Xtreme Preset --nspstune -V2 -mj -h --lowpass 19.5 -b128 --nssafejoint --athtype 2 --ns-sfb21 3 -X3
Insane Preset --nspstune -b320 -h -mj --nssafejoint --lowpass 20 --athtype 2 -X3
```

[Thanks to Olcios (from Poland) for digging up info on the preset switches [here](#), line 966 for standard, line 991 for xtreme, line 1019 for insane.] Update 19dec2001: [Dibrom](#) released his new ALT presets for LAME. He says they're better than the regular presets. You'll need a recent version of LAME to use these presets.

ABR

The documentation that comes with Lame (usage.txt) has this to say about ABR: ABR is still in beta, but it should always give results equal or better than CBR. The best explanation that I've heard comparing alt preset standard [VBR] with 256-kbps ABR comes from CiTay:

ABR at this average bit-rate usually sounds better than VBR, by definition. ABR is sort of an enhanced CBR, and has not much to do with VBR. The difference is that VBR depends on LAME's psychoacoustic model. If the psy-model says that something is easy to encode, VBR takes a low bitrate frame.

But flaws in that model could lead to the situation where something is hard to encode in reality, but the psy-model misinterprets the signal and says "easy to encode". Then VBR chooses a low bitrate where it really would need a 320 kbit frame.

With CBR and ABR, this can't happen. `--abr 256` always takes a bitrate that's high enough and doesn't rely on the psy-model (which is far from perfect). Yes, you'll waste bits at those places in a song where you don't need many bits, but you also have better overall quality .. ABR yields better sound quality [tho many people won't hear the difference].

If you have plenty of hard drive space, you can rip your songs in one session (to .wav files), then encode them to mp3 files later, preferably overnight, while you sleep. This is referred to a batch encode, and RazorLame supports it. You'll wake with a bunch of new mp3's on your hard drive.

I prefer to have EAC rip & encode in one step. I currently use ABR, but have used VBR (`--r3mix`) in the past. VBR is noticeably slower. I have a P4-1.6a@2138-MHz, and I never notice myself waiting for any files to encode. But I have definitely found myself saying, "Come on already," when using VBR. So if you're using VBR exclusively, you probably want to look for ways to encode overnight, especially if you have plenty of hard drive space to store the .wav files.

The MP3 encoding format supports a maximum of 320kbps. It's possible to exceed this, tho. For example, Lame can exceed 320kbps with the `--freeformat` switch. The MAD decoder will play MP3s up to 640kbps (see the last page of this guide for info about the MAD decoder), but the standard Winamp player supports a max of 320kbps.

MPC supports a maximum of 1200kbps, but MPC is a non-MP3 format. See the page on Alternative Lossy Encoders for info on MPC, also called MPEGPlus and MP+. Altho it makes little sense to use a lossy encoder to encode at bit-rates this high, since you can usually achieve loss-less compression at ~700kbps. Audio quality vs file sizes vs encoding methods is a hotly debated topic. Everybody has different ears, different equipment, different musical preferences, different songs, etc. This means that everybody has a different opinion. For example, you can read what Julius thinks about VBR (variable) vs CBR (constant) vs ABR (average) at jthz.com. Razorlame does not currently support ID3 tags. This is probably its weakest point. If ID3 tags are important to you, you can:

Use a 3rd-party tagger, something like [renatager](#), or [id3 master](#), or [mp3 tag studio](#). There are a million taggers out there. You may also want to check out [Lameb](#). Encode directly from EAC or CDex, which will pass the ID3 info directly. Use Monkey's as the Lame encoding front end.

If you want to use Monkey's to encode MP3s using Lame, you need to put a copy of Lame into the directory Program Files\Monkey's Audio\External. Click the down-arrow beside the 'Compress' button, and go to 'Compression level,' then 'External,' then put a check mark next to Lame. The default is `--r3mix b128`. If you prefer something else, select thru: External > Lame > Configure, and enter whatever you want (usually copy-n-paste). When you hit the 'Compress' button, you'll get mp3 files, not .ape files. Don't forget to put a check in the box labeled Automatically tag files on compress under Options | Tagging (F9). Monkey's also allows you to customize the tag. Click the Help button under Tagging (under Options). You can learn about ID3 tags [here](#). Another good place to learn about MP3 quality is here, at mp3-tech.org.

r3mix.net

Step 5: Back in the day, Roel's site [r3mix.net] was the authority for MP3 quality. He was the only one basing his assertions on technical [rather than subject] analysis. Sadly, the original site has shutdown, but I've found a [mirror posted here](#). As I've discovered, it is difficult to keep a site updated with the latest info regarding audio encoding. Things change fast. This is why [Dibrom's HydrogenAudio forums](#) have become so popular: forums operate in real-time. [Have you noticed the absense of pop-ups and ads there? Uncommon for a site with that much traffic.]

I'm not sure what happened to Roel's site, but I'd guess that he found it too much work to keep it updated with current information. Even though the content no longer represents the cutting edge, it's still worth skimming through. In particular, you might want to look at the following sections labeled [see the black buttons to the left]:

- Ripping
- Encoding
- Quality
- Myths

Nothing to downloads at this site. The main point Roel made was that: **128-kbps** is *not* CD-quality. At the time, everybody was using **128-kbps**. Many still do. He also proved that the [Xing MP3 encoder](#) was not very good. At the time, it was the most popular, because of it's ease of use. [LAME](#), on the other hand, was a complicated beast. Much debate revolves around the issue of *MP3 quality vs bit-rates*. Folks get worked up over the subject. This is because everybody comes to these discussions with different ears [hearing ability] and equipment. Any everyone likes to listen to different music.

So it shouldn't surprise you that consensus over the "perfect bit-rate" is impossible to arrive at. But things in this area have become much less complicated. Most gurus agree that the `--alt-preset standard` setting produces the best bit-rate. Obviously, many factors come into play. But no one is

going to complain about an MP3 encoded with this setting. [Need to use [LAME v3.90.3](#)] If you don't mind sparing the extra bits, **--alt-preset extreme** will give you an extra added measure of quality .. that you may, or may not, be able to hear. [FastForward](#) and [cd-rw.org](#) are two other sites where you can quickly learn a lot about MP3 encoding lore [both have pop-ups].

SatCP & Elite DAE

Step 6. Install EAC. Now to configure it. There are two guides. [Chris Myden's \[Elite DAE, Calgary, Canada\]](#) is quick-n-dirty. [SatCP's \[The Coaster Factory, Belgium\]](#) explains in detail what the different settings do. You decide which one you want to use. They each come with their own strengths and weaknesses.

RazorLAME

Step 7. Install both [LAME](#) and [RazorLame](#). Then go back to SatCP's site [Belgium] and read [his RazorLAME tutorial](#). It's only a single page, and won't take long. Copy the LAME encoder to the RazorLame directory. It's it's not there, Razorlame will ask you to navigate to it. RazorLAME is simple to set up. The most important part are configuring the [encoding settings](#).

Options: 1 or 2 steps

Step 8: There are two basic ways to encode MP3s. You can either configure EAC to automatically call and launch the LAME encoder [using the [appropriate settings](#)] options, or rip first and encode later, in a separate step. I prefer the second option. But this is merely personal preference. I rip my songs first, and use RazorLAME to batch encode overnight, while sleeping. I wake up with a bunch of new MP3s on my hard drive. You can even configure RazorLAME to delete the original *.wav file when it's done encoding.

Note that when you rip & encode at the same time, it still occurs in **two separate steps**, even tho it may appear as one. In other words, EAC will still rip to a *.wav file, then call and launch the LAME encoder, and finally delete the *.wav file it ripped [if EAC is configured to do that, or course]. Some people think EAC rips straight to MP3. But, if you think about it, you'll see that this is not possible. If EAC didn't first rip to *.wav, LAME would have nothing [no *.wav file] to encode. Whether the *.wav file goes to your hard drive, or simply remains in memory, while LAME is encoding, I don't know. [It doesn't really matter.]

Rip CD Audio & Encode MP3s

Step 9: We're done. Have at it .. knowing you'll have the highest quality ripping & MP3-encoding set-ups on the planet .. and you didn't have to pay a penny for it .. thanks to many generous people .. from various countries. The Internet has brought together some of the world's finest coders and their software programs right into your living room. Welcome to the **21st** century. Next, let's take a look at some **alternative [non-MP3] lossy encoders**. I am still in the process of updating this guide. See [here](#) for the remainder. I have not yet completed updating and guide. When I do, I'll complete the PDF.