

Primer on CD-R

The following is a collection of the primer pages on recordable Compact Disc formats from my CD-R site. I have favored simple expression over technical detail. For that, I urge you to check other resources, as found in the URLs. In particular, the information from Adaptec is authoritative, where the following is not. The focus is on PC's and Windows; those with Unix or Mac systems are welcome to translate to the extent possible. (I try not to write about things I don't know.) This is a snapshot on the date of writing. Anything which happens to be valid at that time may or may not be next month. A word to the wise is sufficient (I hope).

Note that I have not edited the pages captured here. Therefore, they refer to links which are not implemented and may have other anomalies. I do not plan to maintain this Acrobat file on any regular basis; I'm providing it as a convenience for those who have requested it (or who were too bashful).

Inputs are invited for corrections and for additional topics. Best of all would be a draft for posting here; it will get you a byline and international exposure. (Great for the resume, you know.)

Current information may be found at <http://resource.simplenet.com/>

Reach me (Mike Richter) at mrichter@mindspring.com

Visit my opera (etc.) WWW site - with an autobiography and other stuff - at <http://mrichter.simplenet.com/>

Make Your Own CD's

*In slightly modified form, this article was published in **User Friendly** the journal of the Los Angeles Computer Society in October 1997.*

A quiet revolution in technology began near the end of 1995 with the release of the first practical consumer CD recorders. That revolution has been continuing as both hardware and software have evolved. By now (late 1997), do-it-yourself CDs and CD-ROMs are quite reasonable goals for the home user and are routine tools for many small offices.

The recording process

A CD-Recordable (CD-R) disc differs fundamentally from a CD or CD-ROM. A 'pressed' disc is made from a glass master with the information stored as pits and lands. That master is pressed against a softened plastic aluminized to give a shiny, metallic surface which reflects the reading laser. The signal generated is very strong and unambiguous. In a CD-R, a flat plastic substrate is plated with gold and covered with a dye layer. The dye, which may be gold, green or blue, has the property that when exposed to an appropriate, strong laser light, it changes state. In one state, it is nearly transparent to infrared; in the other, it is more nearly opaque. The reading laser passes through the dye layer twice: once on the way to the gold layer, once on the way back. The resulting signal does not show as great a difference between a zero and a one as does the pressed CD, but it is still enough for most readers to decipher. The quality of the signal from a CD-R depends on the medium used, the writing laser and the reader. As a result, there is no 'best' CD-R blank or 'best' writer.

Compact Disc audio

A Compact Disc Digital Audio (CDDA or CD for short) consists of a continuous stream of bits translated in a CD player into a digitized waveform which is then converted to analog for listening. Also on that disc is a Table of Contents (TOC) which consists of tiny files describing the start point and the length of the recording. They are specified in terms of recording 'blocks' on the disc; each block is 2K long and there are about 330,000 of them on a normal CD or CD-ROM. It is not important when listening to the music, but is very important when creating a CDDA to remember that the sound is not stored in files at all, but in that stream of bits. (The book of specifications for CDDA was originally bound in red and is now universally known as The Red Book; other colors apply to standards for other formats.) A redbook sound must fit the standards in detail and be recorded at 44.1 KHz sampling at sixteen bits' resolution and two channels.

CD-ROM

A CD-ROM differs fundamentally from a CDDA in that its information is stored in conventional files of the sort familiar to computer users. One of the jobs of CD recording software is to translate a redbook bit stream into WAV files and back. Using any standard CD-R package, that conversion is transparent to the user, but for technical reasons is very important. The process of retrieving the digital bit stream from a CDDA is called Digital Audio Extraction (DAE). Not all CD-ROM readers can do DAE; some do it only very slowly, very poorly, or both; and some do it quickly and well. Therefore, not all readers are equally good at supplying information for subsequent recording if you want to make a CD-R of favorite selections from your CD library. Incidentally, the directory to the information on a CD-ROM is not stored in a FAT (File Allocation Table) or other familiar file system; it's in the TOC. One key job of the Microsoft Extensions in MSCDEX is to make the CD-ROM's TOC look like a FAT to the operating system. That's why drivers are loaded either in DOS or in Windows (or both) to enable your computer to 'see' your CD-ROM drive.

Variations on the themes

As if CDDA and CD-ROM were not enough, there are variants which mix sound and data, CD Video interactive, Photo Discs and more. Different programs support more or fewer of those formats; if you have special needs in these areas, you must find the recording package which does the job you need. There are also two major innovations in recording which may lead to new uses of CD-R when they are fully integrated in software. Packet writing allows you to dump a collection of files onto a disc when you wish but to delay building the TOC until you are ready. The other major change is the rewritable CD-R. A packet writer using a read/write (RW) will have software to mount the drive as though it were a removable hard drive or large Zip. The next step is already sampling: DVD-RAM. With that, a rewritable packet system will hold gigabytes of data!

A final warning: At this writing (September, 1997), CD-R is not a science or a solid technology. It is essentially an art form, with the advantage that once you have the formula for what you're trying to do, it tends to work repeatedly. However, the tools of that art are changing at remarkable speed. New hardware and new software emerge monthly and the truths of today will be questionable or false in a few months. However, so far it has worked well enough for me to have seen five titles pressed into 4,000 CD-ROMs - and to have at least five more projects in active development.

Getting Started in CD Recording

This is a personal set of recommendations on starting with your new CD Recorder. The fundamental idea is K.I.S.S. Keep It Simple, Stupid.

1. Install your system - follow instructions.

No, you don't have to do it this way. You can use your existing SCSI adapter, run other software, terminate in ways that are not recommended. And all of that may work. But just for the heck of it, start off by following the instructions. Then, if it doesn't work, you call up the manufacturer and get useful information on fixing it.

2. Prepare your system to burn a CD-R.

If you're running Win95, check the information at the Adaptec WWW site and configure the OS as directed there. Running CD Creator in Win 3.x, increase the buffer settings in CD Creator to maximum values. In NT 4, run Easy CD Pro 2. If the software that comes with your drive does not run in your preferred OS, don't run it there.

Clear out enough space on your HD to be sure you can burn a CD-R. To be certain, use a drive (IDE or SCSI/AV) with 1,000 MB free. Ideally, use a separate drive from that on which your temporary files are stored. Make certain that all drives that may be used are fully defragmented.

3. Burn a simple data disc first.

Even though you will ultimately be doing much more difficult things, start out with the easy case: All data, Mode 1, files on the hard drive. For example, back up all of your C: drive files that will fit and that are not open while you burn. That means, exclude your CD-writing software folder/directory and the Windows and system directories. It's useful to have that backup disc anyway, but the main reason for this is to be sure that you can burn a disc.

While you're at it, make that first disc Disk At Once (DAO). Again, when you can do that, you may want to try Track At Once (TAO) and multiple sessions to see what they do. Try Digital Audio Extraction (DAE) from your CD-ROM reader or from your CD-R; play with copying a CD-ROM, then a CD; make an ISO file on the HD and burn that; and otherwise check the bells and whistles. But do those one at a time, after you know that the basic setup works.

4. One at a time, add functions and modify configuration to make the system work as you wish.

You will probably be able to run the CD-R on the same SCSI adapter that manages your other devices, even those (Jaz drive, HD, CD reader) that will be in use with your CD-R. You probably will succeed with various positions on the chain for the reader and with various terminations. But to attempt a complex installation first would mean that the error messages you get will be unique to your system and none of the manufacturers involved will be willing or able to help you solve the problem. If you make the system work in the manufacturer's configuration, then move one step at a time, you can find out why one of them fails and fix it. If you do three at a time ... forget it!

The same rule holds with other modes, mixed formats, audio files compiled from multiple sources and the other sources of flexibility you will ultimately want for full exploitation of your hardware. It will cost you a few blanks, but will save you endless hours of frustration and save the people who are willing to help you even more.

5. Don't use untried shortcuts.

It would be wonderful if your 12x IDE CD reader fed digital data to your CD-R, but it probably won't. Since you have a SCSI adapter for the CD-R, get a reasonable SCSI reader to do DAE instead of beating your head against an uncooperative manufacturer's wall of stubbornness. Or use your CD-R as a reader and make an ISO. Sure, it takes a little longer, but it works. And if you're going to write at 2x, it doesn't matter whether your reader will run at 4x or 12x - it won't read files any faster than they can be written. (Note, too, that the fact that your reader will work at high speed for other functions does not mean that it can do DAE comparably quickly. An 8x reader may not be able to extract digital audio at all; if it does, it may only do so at 2x, or even at 1x. If you must do high-speed DAE, ask around to find a reader that may do it, optimize your system - and hope.)

The same rule applies to your hard drive. It may well be possible to use a single, gigabyte drive for CD-R and the rest of the things you want of your system, but why? Another gigabyte drive will set you back \$100 or so, and that's a bargain. So is getting enough RAM.

The fundamental requirement for a successful burn on a CD-R is maintaining a steady flow of data. Anything that interrupts that flow will create a coaster instead of a usable disc. Turn off anything you can on your computer that is not needed to burn the disc: screen savers, AV monitors, fax receivers must not be active if you want to make a disc you can read. Your disc must be defragged to avoid wasting time while the heads seek the next data item. If you're copying a lot of small files, you will want to make an ISO file for the same reason. Again, you may get away with violating those rules, but until you know your system and its capabilities, don't take the chance.

Terms of Art

This is a selective list of terms relevant to CD-R with explanations rather than definitions. Organized and authoritative information may be found (as usual) in the FAQ at <http://www.cd-info.com/CDIC/Technology/CD-R/FAQ.html> and at Adaptec's site. Much of this information is controlled by standards; many of those standards are controlled by the International Standards Organization (ISO). Those pertaining to Compact Disc are in the evolving document, ISO 9660.

NOTE: I look forward to getting information to correct any errors and will consider suggestions or requests for additional entries. I am not interested in arguments about facts, particularly if 'someone' told you ... If you read the INet long enough, someone will tell you whatever you want to hear. You will 'hear' that a CD-ROM can write CD-R, that LaserDiscs are digital, and that Saddam Hussein and Benjamin Netanyahu are lovers. I am not interested in rumor, wishful thinking or anything but traceable fact.

Not Relevant - LaserDisc and CD-V

LaserDiscs and CD-V's are analogue, not digital. Their video and audio data are not read or written on a computer. Digital data on a LD are encoded into analogue to be recorded. Regardless, these media are not readable or writable with CD-ROM or CD-R.

Slightly Relevant - DVD

DVD used to stand for Digital Video Disc or for Digital Versatile Disc. It now stands for DVD. It is a medium with great potential for entertainment and for data storage - and is important in the development of CD-R. However, DVD is in flux right now and what it will become is TBD.

DVD - Movies

The DVD players in your local electronics superstore are the video counterpart of CD players. The format used on DVD movies is MPEG 2. First, that is not the same as MP2 - MPEG 1 Level 2 - which is an audio format. MPEG 2 is implemented in hardware, not software. Creating an MPEG 2 file at this point takes a workstation, hardware and software that would buy a small house in most parts of the world. This is not a job for a PC and probably will not be for some time.

DVD-ROM

A single DVD can hold over 5 GB of data and a few applications see value in exploiting it. While this is not the place to go into technical depth, it is worth pointing out some important technical differences between DVD-ROM and CD-ROM. Maybe it's enough to say that DVD's may hold two or more tracks on each side and may be written on both sides of the disc.

DVD-ROM readers can read DVD movies, DVD-ROMs and pressed CD formats. The first generation cannot read CD-R of any flavor, but later generations are Multiread and will read CD-R and CD-RW. Please note (again) that additional hardware - an MPEG 2 decoder - is required to play a DVD movie on a DVD-ROM.

DVD-R

There is a DVD-R on the market It writes DVD movies (built on a workstation) and DVD-ROMs.. It is commercially available for about \$17,000. If you're interested in buying or using one, find an expert; I can't afford even to think about it.

DVD-RAM

There are two competing formats for DVD-RAM and first production units (under \$1K) are on the market now. It is logical that at most one will win out eventually, but no one knows which format - or whether it will be another mechanism altogether. As I understand the standard format, it will read DVD movies and DVD-ROMs. A 2.6-GB blank (which may be readable in DVD-ROM) costs about \$20 today; a 5.2-GB blank, which is not readable in other hardware, is double-sided and costs about \$40.

DVD-RW

Logically, DVD-RW should include DVD-R - but it doesn't. It should be a variation of DVD-RAM - but it isn't. And it should have a single standard - but guess what? Fortunately, the competitive units are not yet available; if you want to learn about it at this stage, please look somewhere else. I have all I can do to keep up with commercial products.

Compact Disc

Now we get into the real world. Some of these things have been around long enough to be pretty well understood and even to be in regular use. The first topic is formats - in particular, those used on pressed discs. Note that discs are pressed from glass masters. The masters are made from tapes. The tapes may come from CD-R's or from other sources; the pros are not limited to starting with CD-R any more than cassettes have to come from cassette masters or video tapes from VHS tape masters. Therefore, the pros can break the 'rules' we follow in CD-R.

CD Audio

Properly called CD-DA (Compact Disc - Digital Audio), this is the sort of thing you drop into your Discman. Information about it is defined in the Red Book - a formal standard which is explicit and (largely) followed. The Red Book (named for the color of its cover) calls for a standard format which is not subject to argument; for convenience, I use redbook as a word for that format: uncompressed Pulse-Code Modulated (PCM) data sampled 44,100 times per second in two channels of 16 bits each and ordered and interleaved in a particular manner. With reordering, that format becomes RAW or SND on a computer. Wrapping appropriate header and footer information around a RAW or SND file produces an AU or WAV file. As discussed elsewhere in this primer, software which extracts digital audio produces WAV files; authoring software writes CD-DA from WAV files. It is worth repeating here that a conventional CD player reads exactly one closed session on a CD; anything recorded after the first session is invisible to it and if that first session is not closed or is not data, it will not be read at all.

CD-ROM

CD-ROM (Compact Disc - Read-Only Memory) is used to describe any CD format in which data (rather than audio tracks) are written. Hybrid modes incorporate both data and audio and are discussed below. A data track is a session and consists of a Table of Contents (TOC) and data. The TOC is translated into a FAT16 file system by the CD Extensions (such as MSCDEX). The files are any flavor of computer file. More information elsewhere in this primer addresses multiple data sessions and related matters.

Hybrid Modes

On a CD-Xtra disc, a CD-DA session is followed by one or more data sessions. Since the first session is audio and closed, it will play on an ordinary CD machine. The subsequent sessions read as they would on any data CD-ROM. To satisfy the standard for CD-Xtra, the audio session and first data session must be written at once and some additional folders and data must be recorded; however, for many purposes it is sufficient simply to record a closed CD-DA session, leave the disc open, then add one or more data sessions which do not import the first. Mixed Mode is similar to CD-Xtra except that the data session(s) is written first and the resulting disc plays only in a CD-ROM or special player. NOTE: The standard requires that the audio of a hybrid mode be written TAO; some software violates the standard to support emulating pressed discs.

CD Recording

Home recording of CD's in various formats has been practical since about 1995. The technology is evolving rapidly, various packages of hardware and of software implement different features more or less in compliance with the standards - and confusion is increasing even faster than brands and models.

CD-R

Compact-Disc Recordable (CD-R) uses a blank consisting of a polycarbonate base with a layer of dye under a metallized surface covered with a lacquer. Clear and blue dyes may be used, resulting in gold or green appearance; the metal layer is gold with those dyes. A different blue dye is overlaid with aluminum, giving a silver/blue disc. When exposed to intense laser light, the dye changes opacity to the less-intense illumination used to read a disc.

CD-RW

Compact Disc - Rewritable uses a different mechanism for recording. The sensitive layer is a silver-colored alloy which changes state (crystalline to amorphous) when exposed to intense laser light and which changes back to some extent when exposed to different illumination. Those state changes cannot be repeated indefinitely; the specified number of cycles for successful operation is currently 1,000. Although an erasable blank is fundamentally different from the kind used to write once, a CD-RW recorder will write CD-R without penalty.

Only some players - CD audio or CD-ROM - can read erasable blanks. CD-ROMs designated Multiread should read them reliably; others may or may not. Some audio players which read write-once CD-R media will also read erasables; some will not.

Packet Writing

Where CD-R and CD-RW were designed to make a disc which behave like a pressed CD or CD-ROM, packet writing is designed to allow writing comparatively small amounts of data in the way that one writes to a floppy drive. Two different systems are used, one for erasable blanks, the other for write-once; they are discussed in this primer under Packet Writing. However, it is important to note the following, which apply to reading in drives other than the writer.

1. CD-DA cannot be written in packets; a packet disc cannot be read in an audio player.
2. A packet-written disc may be finalized to a format readable under Win95 or Win NT, but cannot be read at all in DOS or Win3.x.
3. A write-once packet disc may be readable without finalizing in Windows 98 and an imminent Apple system.
4. An erasable packet disc requires a driver to be read. It cannot be finalized.

Media - Design and Durability

Let's take a look at how CD media are made and at what that means for their durability. There are darned few hard data available on many of these matters and even fewer reliable numbers, so apply a substantial fudge factor - and give me better numbers if you can find them. Please remember in what follows that the light used to read a CD is infrared and not visible to the human eye. What you see is not what the reader sees, so do not be misled. I am also neglecting here (out of ignorance, I must admit) the whole business of the grooves pressed into all discs and containing both positioning information and code for the type of medium.

To begin with, we should examine a pressed disc. Essentially, it begins with a fairly thick layer of tough, clear plastic. The top surface of the plastic has been inscribed with pits and is covered with a very thin layer of metal. On top of that is a thin layer of that plastic, then the graphics which are silk-screened on the very top. Going back to that inscribed top of the bottom piece of plastic, it is created by pressing a glass master into the surface. Where the result is flat (parallel to the face of the disc), the reflection from the metallized layer is strong; where there is a slant to the plastic, the illumination is scattered and there is no significant return signal. Thus, the maximum brightness is high and the minimum is quite low.

An erasable disc (CD-RW) is similar in some ways to the pressed disc. The top and bottom layers are the same, but the plastic is smooth. The big difference is that instead of a metallized layer which simply reflects the light uniformly, the erasable uses an alloy which changes state when illuminated strongly. In the crystalline state, the alloy reflects light quite well. But it can also be switched to an amorphous state in which its reflectance is substantially lower. The frequency of illumination for reliable reading of an erasable is different from that for a pressed disc or for a CD-R, so special hardware is needed in a reader to allow it to get the signal back. It is also significant that the change of state is not completely reversible. After a number of cycles, the alloy becomes stubborn at a spot and insists on being either crystalline or amorphous, ignoring all the urging that the writing laser may offer. As a result, the disc will gradually develop errors over repeated erasures; eventually, they will be too numerous for error correction to deal with and the disc will be a coaster.

A write-once disc (CD-R) uses yet another arrangement. The layers are similar: thick plastic, smooth surface, metallized reflecting layer, thin plastic over the top. The essential difference is that the thick plastic layer has a dye in it which changes color when illuminated by the writing laser. Before illumination, it may be relatively transparent to the reading frequency; after illumination, it will darken. The dye is deliberately unstable (otherwise, it would not change transparency) in two different senses. One is that it is able to be written - to respond to the writing laser by changing color - only for about five years; the other is that over time and depending on handling, the dye can fade. When working right, the dye absorbs some of the light when it would ideally be clear and allows some through when we want it to be opaque, so it does not offer the contrast of a pressed disc.

Writing and labelling the disc

The top plastic layer on a pressed disc is pretty sturdy - not as durable as the thicker bottom layer, but pretty good especially when it has a nice coat of silk-screen ink on it. An erasable is usually pretty good in this respect as well; not as resistant to mistreatment as a pressed disc, but a lot better than a typical CD-R. A write-once blank may have a durable top surface added or not. A durable surface, like one advertised for 'long life' or used to support ink-jet printing, is still much more vulnerable to scratches than is the top of a pressed disc, but the least expensive blanks, with no additional protection or surface printing, are the most fragile of all.

In general, if you want to write onto an uncoated disc, you should use a pen made for the purpose. TDK has one available for about \$3. Many uncoated discs will take writing from the felt-tip (NOT the metal-sheathed ultra-fine) Sharpie, but there is a slight risk that the solvent in the Sharpie's ink can etch some plastics which may be used. Other pens may well be safe, but why not hedge your bets and either find out from your medium's manufacturer or stick with something made for the purpose? As for pencils and ballpoints: you might as well use a dentist's drill or a sandblaster.

Another option is to apply an adhesive-coated paper label such as are available from Avery, Neato, Stomper and others. That's a fine solution, but there are some risks. Obviously, the label needs to be pretty well centered to avoid problems in high-speed readers. Also obviously, you don't want loose adhesive to foul up your reader or its optics. However, the biggest problem is the adhesive used. It must not let go. If it does, the label will peel away in part, snag in the drive and potentially peel some of the adjacent plastic layer. And that is disaster. If you scratch the thick layer moderately, there will probably be no effect at all. But an uncoated disc has a remarkably thin upper layer and even a tiny scratch or hole will disrupt the metal layer below; a disturbed reflecting layer means no reflections, which means no data - dead disc.

Lifetime

Okay, now you're prepared to handle the disc with due respect for the ultrathin lacquer layer on top. How long will the disc last if you don't sandblast it? If it's a pressed disc, the answer is likely to be 'forever'. That is, there is no decay mechanism known for the CD's sandwich of plastic and metal if it is made well. We may learn better some day, but for now when we find a CD that has gone bad, we suspect that it was made badly, that something led to oxygen reaching the metallizing layer or some extraordinary event - like a fission bomb - turned the plastic opaque.

Write-once media are another story and one which is less understood. Remember, that dye which holds the information is unstable. Time alone will cause the dye to fade or to grow opaque. Either way, it will mask the data. From accelerated tests, manufacturers will claim life expectancy of 100 or 200 years, but they don't know. On top of that, the real life depends on how you store the disc. Intense light - sunlight or most other forms - will change the dye. After all, the only difference between the sun and the writing laser is that the laser concentrates its light much more finely. How much exposure does how much damage is ... unspecified. Even worse is uncertainty about storage temperature. Life tests are accelerated by holding the disc at a higher temperature than normal; no one seems to be willing to speculate on what happens to that 100-200 year estimate if the disc is not held at 20 C (or whatever).

The erasable disc's decay is probably something like that of the write-once. There have been reports that intense illumination (in one case, with an ultraviolet PROM eraser) will erase an RW and that seems quite likely. My guess is that cosmic rays will also lock the alloy in one state, but I have found no data from manufacturers or others on the matter. But the real killer is that 1,000-cycle estimated life. Well, I certainly don't expect to write and erase the blank a thousand times - come on, be sensible! Unfortunately, you may be doing more erasing than you think. In particular, if you are writing fixed-length packets, the area which holds the directory (think of it as a FAT, though it isn't) is written and rewritten each time you modify a file. It does not take a lot of scrubbing of that part of the disc to reach a thousand cycles to become unreliable. By now, you probably will not be surprised to know that we don't know how to assess that vulnerability, to predict how much use of a disc written in fixed-length packets will kill it.

De Beers claims that 'A diamond is forever.' But diamond is an unstable form of carbon under normal conditions and degrades from the time it is mined. A CD-R or CD-RW treated well may last effectively 'forever', too - as long as we will care about it (though not as long as a diamond). Mishandled, 'forever' for a CD-R may be measured in minutes.

Mail me at mrichter@mindspring.com
Return to my CD-R home page

A Happy Medium

While I do not mean to shock you (prepare yourself, please): the CD-Record process is not perfect.

Now that you have recovered, let's take a look at what is imperfect and what can be done to minimize at least part of the risk. A commercial (pressed) CD or CD-ROM consists of a thin layer of aluminum inside a block of plastic. That aluminum has pits and flat areas; the flats reflect the laser light to provide the signal. That signal is normally very strong because the aluminum reflects almost all of the laser light shining on it, the pits reflect almost none, and almost anything will read the difference quite easily.

In contrast, a CD-R has less contrast. Recording a CD-R means subjecting a layer of dye to enough laser energy to make it change its state. If you leave a color photograph in the sun long enough, it will change color. Its dyes have changed state. The same sort of thing (though much better controlled) is going on in the CD-R. The dye then allows more of the light of a reading laser to pass through the area where it has been written. That light is reflected by the layer of gold behind the dye and passes through again. As a result, most of the reader's light is passed by the area which has been written and most of it is blocked by the area which has not had a state change. Even under the best conditions, the bright spots are less bright and the dark ones less dark than on a pressed disc. Reliable reading then becomes more difficult on a CD-R than it is on a commercial disc.

How well a blank will record in your CD-R depends on several factors. How well it plays back depends on others. Essentially, if a medium is 'right' for your recorder, it will change state very effectively when it has been written as you intend. However, if you write at a different speed, it may not work as well - it may no longer be as well tuned to the energy the laser lays down. (Perhaps you are recording more slowly; if the areas that should have changed state did so effectively at a higher speed, they can't change more when hit with extra energy. However, the parts that should have stayed opaque might get enough spilled energy at the lower speed to change a little - perhaps enough to make them a little less black when read out. Change to a less sensitive medium and you may have better results.)

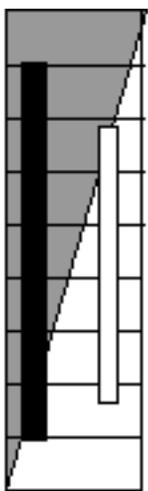


Figure 1

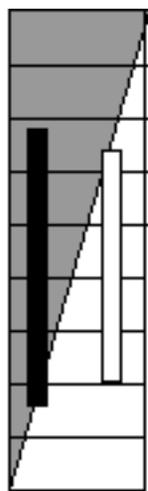


Figure 2

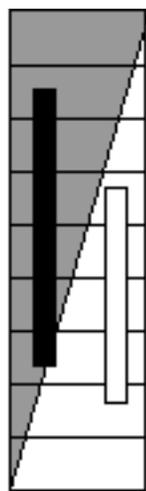


Figure 3

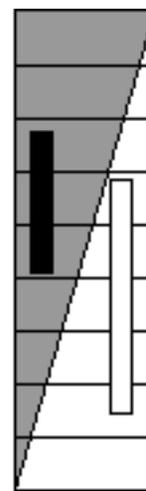


Figure 4

The pictures above are supposed to show the range of laser signal received from the disc when illuminated for reading. Perfect reflection would be the solid white at the bottom of each Figure; perfect absorption would be the solid gray at the top. The signal available in each case is the black bar on the left within each Figure; the detection ability of the reader is the white bar on the right. Figure 1 represents a good, pressed CD and a good reader. The maximum signal is much more than is needed for a the reader to see a one; the minimum signal is much less than that needed to be recognized as a zero. Figure 2 suggests what happens with CD-R: the signals are closer to the thresholds of the reader, but still unambiguous.

In Figure 3, a less sensitive reader (needing more light) is looking at a CD-R which produces less signal at both ends. Notice that here a one is ambiguous; it lies between what the reader would call a zero and a one, so that there is a significant chance that a bit will be misread. If you are reading a megabyte file, there will be about 4 million ones and about 4 million zeros. If the chance that a one is misread is only one in a million, there will be about four errors in reading the file. That means that four times, there will have to be a reread; it's unlikely that the reread will be wrong again, but the reading process has been slowed a great deal. Of course, if the one is even weaker or the reader needs a little more signal, the chance of error goes up until the reread also has a significant chance of failure. Especially if the disc is imperfect - perhaps a spot of dust - it can become completely unreadable.

Figure 4 represents what happens with an erasable (RW) disc. Notice that the signal level for a one becomes much lower; the dye is never as nearly transparent as it is for a CD-R. As a result, a conventional reader cannot make sense of the disc at all. In order to read both RW and conventional discs, the reader must adjust its parameters based on the signals it receives. Such dual-mode readers are becoming available at this time (Fall 1997), but few are on the market. If RW discs become common, the demand for dual-mode readers will be significant and they will become the standard for new drives.

Finding a happy medium

Now that we can see why it is the combination of writer, medium and recorder that makes a difference, we can look at how to determine what combinations will make you happy. If you have a writer, a reader and some blanks, you are ready to burn them and try them. If the result leaves you unhappy - reading is slow, audio is noisy or whatever - clearly, any of the three components can be changed. A writer costs about \$400; a good reader will set you back about \$100; blank media run around \$3. Which one would you change first?

There are three major kinds of media with substantial variations in two of them. Relatively few companies make blanks, but many sell them; those who do not make their own buy them from one of the manufacturers. The major differences in formulation are indicated by the color of the writing surface: gold, green or blue. There is more variation among the different varieties of gold or of green than between different colors on average, but at least you can be sure that you have a different formulation if you change colors. Because that is what you will want to do: try different media until you get what you want.

Let's assume that you're going to go with one of the major brands - perhaps one that you expect to find available consistently and at low cost. Buy a few blanks for your first runs. If you expect to copy audio CD's on the fly from your reader, do one of them. If you often write data-only CD-Rs from your hard disc, do one of them. Then check out how the discs perform on the different readers you may want to use. For example, check your audio CD-R in your car player if you expect to use it there. The mobile units tend to be more fussy about CD-Rs than good home audio players or computer CD-ROMs. If everything works well in all of your expected applications, lock in on that brand and variety (if you can tell the variety from the package). Buy a supply of blanks and keep a few in reserve so that even if a new batch turns out to be bad, you still have some that you can rely on.

There are many ways to determine how good a copy is. There are thorough tests which are tedious and simple ones which go quickly. One of the simplest takes advantage of the fact that a poor track will often need to be reread. The rate at which errors occur in blocks of data (called BLER) is what you want to determine. When you insert a disc into a reader, the activity light goes on while the Table of Contents (TOC) is being read. If the BLER is high, even reading the TOC will take some time; if the TOC cannot be read reliably, the drive will take a long time to determine the problem and then may spit the disc out. Suppose that you have just made a copy of a disc that you own. When you put the original into the drive on which you plan to read, the drive light turns on for five seconds. Put in your copy; if the light goes on for five seconds, the BLER is very low and the combination is good. If it is on for, say, nine seconds the combination is marginal, but not hopeless - look for something better. If it's on for thirty seconds (even if eventually it goes out), the combination is useless and you need to scrap that medium and look for something else for that job. Remember, too, that because one medium works best for you at one speed for one purpose and in one reader, you can't be sure that it will be the best choice when any of those changes.

Media

The medium - the blank onto which the recorder writes - is one key to successful CD-R. Unfortunately, there is not the consistency one would wish among the various types. At least three different forms are common at this writing (July 1997), distinguishable by the color of the active surface - the one that goes down and onto which the laser writes. They are gold, green and blue. None of them is 'best', and the gold and green vary greatly among brands. Even a single brand may have different product, depending on the actual manufacturer.

It is the combination of medium and recorder that determines how well your discs come out. If someone finds that Brand X makes great audio discs, you may find them terrible because your recorder does not use them as well. Thus, you may want either to stay with the hardware manufacturer's recommendation or with the results reported on the newsgroups or the mail list. Note, too, that an audio disc that plays well in some CD machines may not be acceptable in others and that what works best for data may not handle audio as well.

All media on the general market are of essentially the same length. There are slight variations, but whether the package is marked 650 MB, 680 MB or something else, it is almost certainly about the same size as any other with similar markings, as attested by the claim of 74 minutes. The longer discs (80 minutes) and shorter ones (63 minutes) are hard to find and often disproportionately expensive; the 80-minute discs are also reported to be harder to write successfully. Otherwise, the apparent difference in capacity is largely a marketing difference in counting megabytes and is of no practical interest. There is slight variation in actual length - a matter of a few seconds - but that is seldom important and may not be consistent even within a brand.

Wrapup

If you have gotten to this point, you should know the answers to questions often asked in the newsgroups and elsewhere:

1. What's the best medium (variations: for my drive, for audio, for mixed mode)
2. What's the best writer (or reader) - sometimes specifically for audio or data or mixed format
3. Why won't this reader work on discs that work elsewhere? Is it broken?
4. Why won't my writer read discs that it writes?

Since you can answer all of those and many more, you are an expert. Just as I am. It doesn't take much to become an expert, but let's not spread the word. :-)

Erasable Media

Erasable (RW) blanks are fundamentally different from write-once media. They may look similar, but that is misleading.

A write-once blank (CD-R) uses a dye which changes composition when exposed to strong light of the right frequency. That change is not reversible; the new form is stable and absorbs the reading laser differently from the original. An erasable disc (CD-RW) uses an alloy instead of a dye. When exposed to light of the right properties, it changes from crystalline to amorphous; another form of exposure switches it back to amorphous. The reflectance of the alloy depends on its form. Essentially, as a crystal it reflects the read laser as though it were a mirror; in amorphous form, it scatters the illumination.

Changing from one form to the other is not perfect. In practice, the reflectance difference between the crystalline and amorphous states decreases with each cycle. After about a thousand cycles, the ones and zeroes can be mistaken and data are lost beyond the ability of ECC to correct the errors.

In order to provide the appropriate erasure options, the software has to know whether the disc is in ISO format or UDF. If it cannot determine the format because of a failure in writing the TOC, it gives up. How then can the disc be erased? One way is said to be exposing the working surface to strong ultraviolet - either the sun for a long time or an EEPROM eraser. I have no specifics on the process, so I only report the word on the street. It is also possible to fool the software with a good blank. Insert that disc into your drive and start the s/w which you hope to use. Just before you hit the last OK to wipe out the data, swap the good disc for the faulty one. If you're lucky (it may take several tries), the disc will be erased. Erase it again to be sure and to get everything really zeroed out.

Since the illumination needed to read an erasable is different from that needed to read a CD-R, only drives marked MultiRead are able to read your CD-RW. Very few audio CD players will read RW discs. Otherwise, you can use mastering s/w on CD-RW exactly as you would on CD-R and expect the same results. That makes erasable media valuable for a couple of reasons. If you're trying something you have not done before, your failures will not fill the trash with non-recyclable plastic. More to the point, the variation in erasable blanks is much less than that in write-once. Therefore, if you cannot tell whether a failure is due to your system or your medium, try an erasable.

UDF is implemented quite differently on erasable and write-once media. For details on that, please see the pages on packet writing. In summary, a CD-RW written in fixed-length packets cannot be finalized to ISO 9660 Level 3, but it can be read in a MultiRead drive with an appropriate driver installed.

Keep It Clean

This note has nothing to do with X ratings. Put whatever information you wish onto your discs, but it won't get there unless the laser can see to write it.

Most of us have encountered a pressed disc - CD or CD-ROM - which would not play. On examination, it may disclose the reason: a gouge, a smear of grease, a blob of dirt or something else which keeps the laser from focussing on the pits. A simple rule: if the laser cannot see the information, it cannot read it. If it cannot read it, your stereo or your computer cannot make use of it.

The solution is usually simple enough: polish out the gouge or clean off the disc, then try again. Voila! It's fine.

Unfortunately, life is not so easy when you're writing instead of reading. If the laser encounters that blob or smear when writing the disc, the information is lost. The name for the result is: coaster. There are hundreds of millions of bytes on a CD-R. If all are good, it's a disc. If a few are bad, it may be a coaster.

If you're lucky, the coaster isn't all bad. Maybe just one faulty file, maybe a few. Or perhaps the glitch is in the TOC and nothing can be read - or it's detected in trying to work with a UDF disc and it can't be finalized. At worst, it becomes an unreadable mess and everything is lost.

That doesn't happen to me!

True - most of the time, it doesn't. You remove the shrinkwrap and take a dust-free blank from a jewel case. You burn it and - no problem. Or you take the blank from a spindle and drop the cover back on. After burning, if you get a speck of dirt, you wipe it off. So, no problem - right?

Maybe.

There's that disc you wrote with 200 MB and now you want a second session. Or the packet disc to which you add a few MB of downloads each week. Hey, it's been in the almost-closed jewel case over by the ash tray; what's the problem?

Oh.

Well, maybe you can keep it dust-free. Sure, and maybe it will turn out to be one of those 80-minute blanks mispackaged. In the real world, the best way to keep the disc free of dust and dirt is to clean it before each burn. Remember, you don't get to clean it after you find the problem, as you do when reading. Well, that's not true; you can clean the coaster so that it's bright and shiny in your mobile. You just don't get to clean it off and then use it as though nothing had happened - as you do when reading it.

What to do about it

First, let no coaster go undiagnosed. The first thing to do is to look at the writing surface under a strong light with magnification if needed. Look at the circle where the discolored area you've written ends. Look closely for the bit of a hair that came at just the wrong spot. You may not find it - it could have fallen off or slipped to another spot on the disc. Or there may be another cause this time. But if you don't know what made this coaster, then whatever it was will be ready to create more for you.

Another move - highly recommended - is to look before you burn. Keep a can of 'compressed air' and a clean, lint-free cloth handy. Examine the writing side before you insert the disc to be sure there is nothing that can be blown away or wiped off. It makes sense to blow the surface clean before you look; it will get most of the schmutz (a term of art; see any good computer dictionary) so there will be less to see and less to miss. And even though the cloth was clean and lint-free, if you use it, examine and blow the disc afterward.

Note that these precautions are far more important when you are reusing a disc than when you are burning a new one. Even if dirt got under the shrinkwrap, all you lose is that blank. If this is a second session, you risk the first; if a UDF, you risk it all.

Now, I happen to be the careless type and a smoker, too. So I use only caddy drives and a disc stays in its caddy until I'm through burning it. I tie up a \$2 caddy on every \$10 fixed-length packet blank as long as it has data on it. Even a disc in variable-length packets stays in its caddy if it has information of value on it. But if you find caddies inconvenient, you have the other options: single session; keep it clean; make it clean; build a coaster mobile.

How Many Bits??!

In a discussion of error protection, Guy G. Sotomayor, Jrggs@shiresoft.com explained how playing back a CD-ROM at 1x requires reading the disc at more than 5 megabytes per second. If you are concerned about matters such as making copies accurate to the bit level or if you want to understand some of the reasons behind the imperfections of CD-R, this section is worth the time it takes. Before going into his text, let me note that a 'symbol' is a character or a value - basically, just an 8-bit byte. Each sample of a redbook waveform consists of sixteen bits for each of two channels: 32 bits, 4 bytes or 4 symbols.

The first thing to remember is that the data on the CD are stored as 14 bits/symbol rather than just 8. There are several reasons for this, but it makes reading the bits easier. Here are the criteria for selecting which patterns can actually be used:

1. 11 is not allowed
2. 1001 is as close as two 1s are allowed to get
3. 100000000001 is as distant as two 1s are allowed to get (ten 0s).

Each 14-bit symbol is separated from its neighbor by 3 bits, called merging bits, coupling bits, connecting bits or packing bits. They allow the rules above to be applied continuously. Remember at this point we just have a string of bits - there is still nothing indicating where bytes start and stop. To recap, 8 bits is represented by 17 bits - a little over 2x.

Now, we look at how data are actually stored on the CD. Everything depends upon a 588-bit frame. The 588 bits are organized as follows:

Description	Bits/each	Total bits
Sync Word	24 bits + 3 padding	
Control Word	17	17
Audio Samples	6 samples, 2 symbols each	$17 \times 12 = 204$
Error Correction Q	17×4	68
Audio Samples	6 samples, 2 symbols each	$17 \times 12 = 204$
Error Correction P	17×4	68

Total 588.

To figure out how many 588-bit frames per second we're dealing with we go back to what we see on the output of the CD (namely 176400 bytes per second). A stereo "frame" is 24 bytes, so if we take $176400 / 24$ we get 7350 frames per second. Now take 7350×588 and you get 4321800 bits per second. Q.E.D.

BTW, the P, Q, G, etc channels are derived from "stacking" the control words (98 of them to be exact) to get the encoded data. As you will no doubt notice, audio data has ECC applied to it. This is the CIRC that I spoke of previously. This is always there. Data because of its more exacting nature has an additional layer of ECC applied to it to further reduce the chance of an error. Hope this clears it up!

Footnote from Mike:

There are additional issues including the nature of that 'additional layer of ECC' and the numeric values for error rates in audio and in data. Since ECC discussed by Guy is always imposed, I usually do not consider it in the discussions in this primer. The only specifications I have found for accuracy on CD-R suggest a bit error rate of one in a million for audio and one in a billion for data. Both appear to be worse than we experience in practice, particularly the one for data; that extra layer of ECC should cut the rate by a lot more than a factor of a thousand. Error rates for pressed discs are much lower than those for CD-R.

Read Errors

We like to think of the computer reading and writing information perfectly. In fact, we seldom even think about that at all because perfect operation is so common. Unfortunately, perfection is not achievable by man. In Islam, that reality is recognized by requiring that a flaw be woven into each 'Persian' rug. In computers, real hardware provides all the errors we need.

According to the specification, CD-R errors start out as being not more than one in a million. Sounds good, right? 99.9999% accuracy should satisfy even the most critical person - but it won't do for computers. There may be $8 \times 650 \times 1024 \times 1024$ bits on a CD-ROM. If so, 5,000 of them may be faulty and reading that disc will be a major problem. So almost 15% of the space on the disc is 'wasted' in coding which helps automatic error correction - ECC or Error Correction Code. With ECC, the specified error level is cut by a factor of 1,000 so that there would be only five bits wrong on a disc. Even those may be transient so that when an uncorrectable error is found the sector is reread in the hope that good information can be retrieved. In addition, most discs will exceed the specification significantly. Even so, if rereading is necessary the data flow is reduced substantially. The read mechanism must stop its rapid progress, back up and start the sector over.

That saving ECC layer is not applied to audio. If it were, the longest playing time on a CD-DA would be about 64 minutes instead of 74. Similarly, on a VCD there's no ECC on the video files. The idea is that the extra playing time costs only slight errors on playback and that you won't notice the ones that sneak through any error correction applied on the analogue side. For video, the fault is usually a momentary spot in the picture and no correction is used. For audio, the player has some limited ECC of its own, but if it is used there can be a loss of brilliance in the sound.

Another key factor in the quality of information read from a CD-ROM or CD-R is the quality of reading. Some errors derive from the way the track is formed on the disc and are related to the term 'jitter'. Even more significant is that, in general, the faster the data fly past the pickup, the greater the chance that a bit will be misinterpreted. With a pressed disc and ECC, that is not significant. Listening to a CD-DA or watching a VCD occurs at 1x regardless of the drive, so again speed errors don't matter. However, when you extract digital data through DAE or in raw form from a VCD, the story becomes less happy. All other terms being equal, the faster you run DAE, the greater the number of ticks and pops - bit errors - you will encounter.

There is also a quality issue here - and in this, I am providing personal opinion, not objective data. With the increasing pressure from the marketplace for faster and faster drives, other things are being sacrificed. The general public neither knows nor cares about DAE, so the ability to get a good signal without ECC is not a selling point on CD-ROM readers. Therefore the manufacturer will work to find a way to claim 24x for a poorly made drive which might work well if designed for, say, 12x and provide competent DAE at 1x or 2x. In practice, the 24x will be slowed by the need to reread even data and may deliver very little more in effective speed than a 12x, while being unable to provide satisfactory DAE even at 1x. Even a pressed CD-DA may sound dull on that drive, though typical computer speakers are unlikely to show that since they, too, seem aimed at bass boom instead of clean high-end response. And as has been reported elsewhere, running a disc at high speed increases vibration and mechanical noise.

Finally, a CD-R is not a pressed disc and it does not perform as well. In particular, a pressed disc has great contrast between its strongest reflection and its maximum scattering of the reader's light. A CD-R or CD-RW has a much lower maximum signal and a much higher minimum signal for the reading diode. As a result, the chances of reading errors from a CD-R or CD-RW are higher than for a pressed disc and a copy of a CD-DA copy (especially audio) is likely to sound noticeably worse than the original. Of course, the better the hardware, firmware and software are for the job, the better the results are likely to be. But there is no free lunch, there is no guarantee of perfect writing or reading, and if you do not take that into account as you choose your reader and test your CD-Rs, you may regret it.

Tick - It's the TOC

The Table of Contents (TOC) on a CD-R is the repository of all the information required for your OS to understand the pattern of bytes on the disc. Essentially, it contains everything needed for your system to figure out what information is where and in what format, so that your request to receive information can be satisfied.

For example, on an audio disc the data are not stored in files, but in a continuous stream. The TOC informs the OS (or the player) that a specific block is the beginning of a track. When you play a track, a command is given to go to that block and to begin to retrieve the bit stream which follows. Note that the track itself - in the audio sense - need not begin where the TOC says it does, but that is irrelevant to the player. There are also subcodes which can be used to interpret how to backspace and so on -but they are beyond the present scope. (And if you want to read that as meaning that I don't understand them, I couldn't argue.)

When you go to write a Disc At Once (DAO), all the information needed for the job is available when you start. Mastering software assembles that information and decides what is to be in the TOC. When everything is ready, the laser is positioned and turned on and writing begins. The first thing written is the TOC which says: this is a closed disc of a single session and here is the information on what you will find where among the following bytes. The laser then proceeds to write those bytes in sequence and in accordance with the information saved to the TOC. When the last byte is written, the laser turns off and you have a good disc. If that burn is interrupted after the TOC is written but before the end, everything is accessible up to the failure. An audio disc which fails when track 5 is being written will be usable for tracks 1-4 and maybe for the written part of 5 - but what happens when the track runs out depends on the player.

When you write Track At Once (TAO), the procedure is quite different. The writer begins by inscribing a special area on the blank called the Program Memory Area with information needed to close the session later. Then it records track information and writes the track itself. When it's time to close the session, the writing laser turns off, the mechanism returns to the PMA and the information inscribed there is read back to close things up, record how many tracks there are in all and, in general, to complete the TOC. So if the burn fails before the PMA is read (or if the PMA cannot be read at all), the TOC is not complete and a reader cannot make sense out of it. However, not all is lost! Although the reader needs a complete TOC to do its job, a writer is prepared to make do with less and, in fact, has what it needs in the part of the TOC which is complete to retrieve the valid data.

Suppose that you have written your disc - and the software reports errors. If the error has to do with reading, then it is almost certainly the inability to read the PMA when it came time to close the session. The disc is trash, but you can retrieve the information from it in a writer. That information may not include what was on the last track, so if it was a data session you're in trouble. But if it was an audio session, you should be able to recover all the tracks but the last. Note that the particular error message you get is critical here. If it says buffer underrun, the data are not there so they can't be read and (in TAO) the TOC was not written, so you couldn't find them if they were. If it reports a communication error, it might well have occurred when seeking for the PMA; in that case, your writer may actually be able to read everything on a TAO disc. Finally, if a communication error came at the end of a DAO burn, it may mean only that the runout track was incomplete. Since most modern readers need very little runout, the disc may work perfectly for you although it is 'too long'.

Audio Basics for CD-Recording

One of the functions many people seek with CD-R is creating audio CD's. Software for creating CD-ROM's supports that desire, but each product varies in the ways in which the job is done. For information on your software - read the manual. This note is intended to cover the basics of audio file formats and problems.

CD Audio and WAV files

A selection on an audio CD is very nearly in WAV format. It is a variant of a specific type (redbook) of WAV file: 44.1 KHz sampling, stereo, 16 bits depth. Unlike an ordinary data file on a CD-ROM, the sound file does not contain error correcting codes (ECC) within it to handle data lost in transfer. Instead, a layer of ECC is provided on the disc. One of the ways in which CD players vary in sound quality is that they are more or less effective in using ECC to repair errors in reading the audio.

NOTE that an audio player is required to read only the first closed audio session on a disc. To avoid problems of poor readability, mysterious noises and so on, the beginner should not try to leave the disc open when closing the first audio session. You may write the disc in DAO, write the whole session at once in TAO, or write repeatedly *leaving the session open* until the last track, when you should close the disc. When you understand CD Extra and are willing to experiment, multisession can be fun.

Getting a WAV file

There are two different ways to create a file on a CD-ROM: an audio CD track or a WAV file. Only the former will play in a CD player. However, CD-R software will convert a WAV file (CD Creator will take one at less than redbook standard) into the necessary format and will read the audio file from a CD into a redbook WAV file on your hard disc. In that process, your reader may run at anywhere from 1x to its maximum speed. The higher the speed of reading, the more chance that some information will be lost. Since there is no effective error correction, those faults will show up in the WAV file. Fortunately, that file is on your hard disc and you can listen to it. If it sounds good there, it should sound good when converted to the CD format and recorded. If the copy on your disc has clicks or other problems - don't give up.

The first step is to persuade your software to read the CD at a lower speed. First, you need a player which supports Digital Audio Extraction (DAE). Your CD-R certainly will do that; a CD-ROM reader may or may not. Most SCSI readers provide DAE; very few IDE readers do, and then only with the right software. If your reader does not support DAE, you can use D/A/D conversion or your CD-R. Let's assume we have DAE. Using whatever tools your software provides, cut the read speed down until you get a clean WAV file. That tells you how fast you can run with that CD. Now check with other CD's to find out whether your first one was unusually good (or bad). Ultimately, you will decide that you can use a particular speed reliably. Don't try to cheat on it: If your reader won't give you a good signal at 4x, no software or CD-R media or whatever will improve on that.

Note that you are now beginning to understand why some 8x CD-ROM readers are slower than some 4x. They may run at 8x under ideal conditions, but if they aren't able to retrieve information reliably, even error correction may not be enough to get a good file. So the system rereads that file until it comes out right. There's no standard around to say what 8x (or any other speed) means - except that under some circumstances as many as 8x150 KB/sec will come from the drive. Whoopee!

D/A/D

The information on a CD or CD-ROM is stored in digital format. To listen to that signal, it must be converted to analog (a.k.a. analogue to us old-timer purists). The conversion is done in the reader and uses the ECC of the disc to a greater or lesser extent. If you have put an appropriate jumper from your reader to your sound card (so you can listen to a CD while you 'work'), you can record from that with controls on your mixer. If not, any CD player can be connected to the Line In jack on your sound card - as can any other appropriate source of analog audio. The quality of the signal going into your sound card then depends on the quality of the CD reader/player you are using.

On the sound card, an analog signal is converted to digital - and with that comes a set of problems. Essentially, converting between analog and digital signals is a matter of approximation. D/A is hard enough, but A/D is tougher still. The result is that only very good and very costly equipment will do first-rate D/A/D. Is your present sound card good enough for you? No one can answer that but you. Would the BrandX SuperSound do enough better to be worth the cost? 'Better' depends in part on your taste, and 'enough' is meaningless to anyone but you. Perhaps someone on the mailing list or in a newsgroup can give you insight, but don't expect easy answers. There aren't any.

At least, if you do use A/D/A instead of DAE, you will use the CD's ECC. If a CD you want to use insists on crackling on DAE at 1x, you may have no other choice.

Editing a WAV file

One of the advantages of making a WAV file through DAE or D/A/D is that you can edit it. Tools for that purpose are included in the audio section of the links at this site. Those editors will let you modify the sound in many ways: denoising, adding effects, repitching, cloning, ...

There are some tricks to this operation. One is that some software figures that whatever's tacked onto the end of the file for its own purposes won't do any harm. As far as the computer is concerned, that's so. However, on a CD those bytes become a 'click', and for some strange reason most people don't like extra clicks and pops on their recordings. Here, the solution is easy: don't end your editing with such a program. If you want to use an editor which you know leaves the click behind, do it. Then open the file in another editor that you know doesn't make popcorn (such as GoldWave) and save it. The click will be left behind.

Recording on the fly

"On the fly" means writing the CD-R from the source (here, a CD reader) without going through the extra step of writing to the hard drive. Let's dispose of the obvious first (not that you don't already know it, but for the benefit of those who don't). You cannot read from the CD-R while you are writing to it; therefore, you need a separate CD reader to record on the fly. Recording on the fly requires that the data being read are digital, so your separate reader must support DAE. If you don't want to use a second, DAE-capable reader, you cannot record audio on the fly. Given current prices for a SCSI CD reader under \$100, it doesn't take many coasters to pay for a second drive - and you have a SCSI adapter already for your CD-R, so you probably don't need to buy one of those.

When you go to record on the fly, your software assumes that you can run at full writing speed without a problem. So you're tempted to think that your CD-R that writes at 2x will work fine with a reader that runs at 2x or above. You may be disappointed. For audio, you should already have determined with the WAV tests above that your reader is only reliable at 1x - even though the box says 2x, 4x, or 45.7x. The same thing that keeps a WAV file from sounding good will guarantee ticks and pops when you record on the fly. Therefore, you must slow down on-the-fly recording to the highest speed at which your reader will work. Of course, if that speed is faster than your CD-R can write, you can't beat the CD-R hardware by getting a faster reader. There's still no free lunch.

All the other stuff

Recording a CD is much like recording a CD-ROM. Some combinations of media, hardware, firmware and software work better than others. If gold/gold media work better for you for CD-ROM, they will probably be better for CD, too. Just because your hardware vendor's bundled software has a good interface for CD-ROM does not mean that you will find it ideal for audio (or *vice versa*). There are at least three choices of CD-R software on each major platform and each does every part of the job differently from every other.

There is no 'best' hardware, software or medium. The mailing list and newsgroups will provide you with unauthoritative, inaccurate, contradictory, incomplete and irrelevant information. Still, it's better than buying and learning all those products.

At least I think it's better.

From someone who knows: Pauses in Audio

From: Bart Lynn - blynn@eng.jvcdiscusa.com

As an Engineer for a CD manufacturer, I feel that I can try to explain the differences in the pause times on CDs. Red Book (the Compact Disc Specification for Audio) states that the Pflag must be 2 ~ 3 seconds. Therefore, since Absolute time starts at 00:00:00, the start of track 1 can be from 00:02:00 to 00:03:00. Of course, there can be silence after that point, so there can be in essence a lot of pause.

Most PC/Mac software sets the start of track 1 at Atime 00:02:00, since Yellow Book (Compact Disc Specification for normal CD-ROM discs) specifies that the pause be just 2 sec. At JVC, we start at 2 seconds Atime, then we offset the first track 1 second, prior to the start of the music (CD-ROMs start at exactly 00:02:00). Each additional track is offset 5 frames (30 SMTPE frames to 1 second). The end time of the last track is moved forward 1 second. The reason is that many old CD players (from like 1985, up to the triple beam ones) search for the start of the track by the Pflag, not the absolute time. The new players can be extremely accurate in searching, where the older ones cannot. As a manufacturer, we have to try and support all makes and models, so we have to provide a slight tolerance for them. Additionally, the record label/artist sometimes requests that we provide additional pauses, especially for classical music, or live recordings.

Bart Lynn

JVC Disc America

Extensive additional material is available at <http://www.westnet.com/~gsmith/> - but let me caution the unwary that the subject is complex. Those pages may be difficult to digest, but can be worth the effort.

Analogue Audio

Let's start with the basics. Sound is a pattern of pressure that we sense - usually thought of as sound waves in the air. The usual way to deal with sound waves in electronics is to create an electrical analogue by exposing the sound waves to a microphone. The signal from the microphone can be amplified and recorded - still in analogue form - on audio tape or on a vinyl disc. It can also pass through an Analogue-to-Digital Converter (ADC) to become a digital signal. One of the functions of a sound card is to provide a DAC to deliver a digital signal corresponding to an original pattern of pressure. Another function is to synthesize a digital signal from instructions in the form of a MIDI or similar file. On the other side, the sound card provides Digital-to-Analogue Conversion (DAC) which is then used to drive tape recorders, speakers or other analogue devices. The inputs used for the purpose may be WAV files, MIDI synthesis or other digital signals the card accepts. Finally, the sound card allows mixing of its various analogue signals on input and/or on output. So the card will allow you to mix the analogue signal provided from its CD input with a synthesized sound from a MIDI file to drive your speakers with the combination. And a "full-duplex:" card allows such an analogue signal to be digitized while being played.

There are two ways to get sound from your CD reader. First, there's a headphone output with a volume control. Run a cable from that output to line in on your sound card, and you're ready to go just as you would be if the source was a tape deck. Even better (especially on internal drives) is to connect a special, analogue cable from the reader's plug to one marked CD on your sound card. The sound is the same at the two outputs, but with the internal connection you don't have to remember to twiddle the headphone volume control and you don't have that cable hanging outside the case. Finally, most CD readers support Digital Audio Extraction - DAE - which gets a page of its own in this primer. Recording from the CD as analogue uses the DAC of the reader to generate the signal, then the ADC of the sound card to convert back to digital. Each of those processes is imperfect and can degrade the sound. However, if your DAE is not perfect, the analogue connection may sound better to you - so, use it! Only the purists will insist that you should put up with what you don't like because It's the Right Way. One more point: if you have two CD sources, such as a reader and a writer, do NOT try to connect them both to the same jack on your sound card. Either get a card with dual stereo inputs or run that cable from headphone out to line in.

Let's look more deeply into the question of recording from CD-DA. Since you are likely to have both options, should you use DAE or analogue recording? The answer already given is: use what works for you. But there is more to it than that. Assuming that you DO have the choice (good DAE and an analogue connection), when should you use analogue? First, some recordings have an extra bit set for preemphasis. That is a shift in the frequency response away from flat. Under some circumstances, it can let the producer record a LOUDER signal. If it's set, then DAE from the disc will sound too bright and lacking in bass. If all you're going to do is write that extracted file back to a CD-R, you can live with preemphasis as long as your software allows you to set the bit when you record. (Adaptec's PC programs do not.) If you want to edit the WAV file, you will definitely want to correct the preemphasis first; for that, you will find it easier to record analogue instead of ripping the track with DAE.

Another factor is error correction, which in general is beyond the scope of this page. The important thing about it is that errors recorded onto a CD-R or read from a CD-DA track will get through DAE without correction. But when the reader sends the signal to one of its analogue outputs, it passes it through filtering circuits which clean up some of those errors. The more cleanup the circuits do, the more they alter the sound quality. One reason that a copy of a CD-DA track may differ from the original is that the filters have more to do (CD-R has more audio errors than a pressed disc) and therefore change the sound more. Should you use analogue or DAE on a CD-R? Whichever sounds better to you - but be aware that they will have some differences in sound.

There's a lot of good material at <http://homepages.nildram.co.uk/~abcomp/lp-cdr.htm> with details not covered in this low-level page; stop by for more when you're ready.

Recording from an Analogue Source

Okay, folks - here we go on how to record your favorite LP's, cassettes, 45's, DAT's, 78's and so on to CD. These notes end with getting a clean WAV file. From there, you use advice elsewhere in this primer or wherever to make your CD-DA, MP2 or MP3, or???

Sources

Most analogue audio sources deliver about one volt output. One important exception is a turntable; most cartridges for vinyl or shellac put out less than a hundredth of that level. If your cartridge is crystal or ceramic (few are), you may be able to connect your turntable as though it were a tape deck. However, in general you need to preamplify the output of your cartridge in order to record it on your computer. (NOTE: the preamp also corrects the frequency response of a cartridge.) You can buy a separate preamplifier at Radio Shack or any other general-purpose electronics store or build one with a circuit from my Filespage. If you can't find one for a turntable, one made for a microphone may do. Even better is to use a receiver or a high-fidelity preamplifier. They may show up at a pawn shop or a swap meet. If you have a choice on a stereo unit (receiver or preamp), get one with two sets of tape input/output connectors. The output to use is one pair of connectors marked to go to TAPE REC. Output from your sound card goes to the corresponding TAPE MONITOR inputs to the preamp. In my setup, I have two high-level sources (open-reel decks) connected to TUNER and AUXILIARY, a cassette deck on TAPE 2 (REC and MONITOR), the sound card LINE IN connected to TAPE1 RECORD, the sound OUTPUT connected to TAPE 1 MONITOR, and the amplified speakers connected to the preamp OUTPUT. I can record from either open-reel deck or the cassette or phono to the sound card or from phono, an open-reel deck or the computer to the cassette deck. Just one warning: watch out for feedback loops when you're using the cassette deck.

For some purposes, it's sufficient and more convenient to record the turntable output to tape, then to feed the tape signal into your sound card. There are some other sources to discuss: Digital Audio Tape (DAT), MiniDisc (MD) and turntable. DAT and MD provide analogue outputs and can be connected just as tape decks are. Some also provide digital output; those can be connected digitally if your sound card has SPDIF interface and if the signals are (or can be made) compatible. But that's 'way, 'way beyond the level of a primer (which translates into: Mike doesn't know about that stuff).

Getting ready

Now that we have some sound going into the sound card, we need to look at getting it recorded. The first step is telling the card what input to read and to set its level. Unfortunately, most people are stuck with the absurd Voyetra mixer that comes free with Windows. If you have a better one, use it. (If you're a programmer who wants to do a service for humanity, how about making a freeware replacement?) Start the mixer - usually with a right-click on the speaker icon in the tray; otherwise, it's Volume Control in Accessories. Now, go to the Options menu and select Preferences. Click on Record and make sure that all the inputs are checked before you click OK. Now you have sliders for each input available to you. Pick the input you want. (CD is for the analogue connection of your reader or writer; unless you want that, you should probably select Auxiliary - which is their substitute for LINE IN.) Now, set the slider near the top of its range. You have just told the sound card to record from the selected channel at about the right level. DO NOT close the mixer - you're going to need it again.

Now the sound card is ready to go - but something has to tell it what to do with the selected signal. The Windows applet called Sound Recorder will do for starters, but it records to RAM. When it runs out of RAM (which usually doesn't take long), it's done. So you will probably want another program - such as Adaptec's Spin Doctro, one of the WAV editors (see the URLspage) or CDWAV (see the files page). Each package works differently from the others, but all are similar. Select the Record function, set mono/stereo, bit depth (usually 8 or 16) and the sampling rate (options are usually spelled out). If you are going to make a CD-DA, it's best to capture directly to redbook format:

stereo, 8 bits and 44.1 KHz. If you record with other settings, you must convert with some program, which is slow and inconvenient - and very slow if conversion is done well. Most programs default to redbook; Spin Doctor only works that way.

You're not yet ready to record, though. Your recorder s/w has three choices: it can use only memory (as the applet does), it can use a TEMP file (either in the Windows default directory or one you select in the program), or it can write directly to your output file. Check that program for the way it works. Now, you have to set the recording level. If you overrecord, the result is painful even if it only lasts for a short time. The analogue signal is fed to the ADC (Analogue to Digital Converter) on the sound card. The ADC can only provide its maximum signal - all 1's. The result is that any excess signal is clipped, hard and brutally. Don't do it! If you record at lower level, you can leave it there or you can use a WAV editor or Spin Doctor to correct the level. Before you start recording, put your recording s/w and the mixer in non-overlapping windows on your screen. Find a passage in your source which is as loud as it gets. Activate the indicator of your s/w - sometimes it can be done without starting to record; sometimes, you will have to start some form of recording (which you will later throw away). Adjust the level on the mixer so that the level indicator on your s/w is always below maximum. If it's only 80% or 90% of maximum, that will be fine; those correspond to two and one db loss, respectively, which is not significant (and which will still give you much more signal-to-noise ratio than your system needs.)

Recording and processing

Now you're ready to record. Hit the record button (specify your capture file if necessary), then start your source. I strongly recommend that if you want more than one selection from a side of a tape or an LP, you record the whole side at once. You can always split and edit the selections later with a WAV editor; for splitting a long file, CDWAV can't be beat. These operations use a lot of space on your hard drive and may take a lot of time. Most editors will require at least twice as much space as the file you're editing. If you've captured a 30-minute side, you will need a spare 700 MB or so to hold redbook-format WAV files for editing. As you will discover quickly, lots of RAM and a fast CPU will help a lot, too.

When the file needs to be cleaned, still more time is needed. Removing clicks and hiss can be handled more or less automatically by programs such as CoolEdit and DART Pro, but that work takes a lot of processing. Even light treatment can take longer than playing the file. Different programs and different settings take varying time and give varying results. I use many different programs for my sources, depending on the initial quality and the importance of the recording. For example, I like the way that DART Pro removes clicks, the way CoolEdit handles hiss and noise, and GoldWave supports manual editing. I have spent hours correcting a single, four-minute cut. (No, I'm not rational about that, either.) Along the way, some programs may complain about the headers produced by others. For safety's sake, I strongly suggest that you have StripWave (see the files page) available

I can't tell you how to process your files - that's up to you and your source material. Anything you do to the signal will degrade it; if what you do is more important than what you lose, you come out ahead. The least damage is done by manual editing, but that is excruciatingly slow and painful. (Literally painful; an hour or two of editing will teach you more than you want to know about tense arm muscles.) The tools available today can be used with care to produce minimum loss and satisfying results. Misused, they can turn a favorite recording into mush.

Spin Doctor

In many ways, the easiest program for analogue recording is Spin Doctor. Select the source, select the target, set the cleaning options, and let it go. Still, there are some things you need to take into account to get the best results. First, all the analogue sources are the same, regardless of the different lines from which you select. You still have to specify the source with the mixer. The different options give you different icons on the screen and may let Adaptec (some day) tune the cleaning operations to the kind of source being used. Next, if you select any of the options - cleaning or equalizing volume - the program will record the signal to the HD, then process it, then produce the output

WAV file. It is designed to be allowed to do its 'thing' without interruption; let it have its way. I strongly recommend that you do any processing in a separate operation. First, record the tracks to HD without processing. Then do some cleaning, balancing or whatever by running Spin Doctor from one HD file to another. Finally, burn the finished WAVs to CD-R. Spin Doctor permits you to go from analogue source directly to CD-R, but that simple step often fails. The problem is that any tiny speed error, any hitch in the process will create a coaster. That's because the source is (by definition) running at 1x - real time. The recorder needs a steady stream of data and has a limited buffer. So if there is any difference between the speeds, or if you use an erasable blank (which must be written at 2x), you will have a coaster. It may work for you; if you want to try it, feel free. But safety and flexibility (opportunity to split tracks, edit, whatever) suggest that you go through your HD. Remember, you have to cue perfectly, cannot correct a mistake and cannot adjust levels (or anything else) except where it will be heard forever after.

Cleaning Audio

The following is representative of the way that I clean up audio source files. It is intended simply as a guideline for those new to the process; many other programs and approaches can be used and different sorts of faults on the original call for different tools and techniques. Programs referred to in the text are generally available from links here.

Capturing

If the source is monaural, begin by recording it monaurally. If it has limited frequency range, do not sample at too high a rate. Many of the steps which follow are disproportionately difficult in redbook format (44.1 KHz, stereo); if you can work at 22.05 KHz monaural, life will be much easier. I usually record with CDWAV (available here). It permits me to capture as I wish and to split the tracks from a tape or LP while saving them. I also use CoolEdit for the purpose since I like its level indicator or GoldWave for quick manual editing. I rarely use Spin Doctor for poor sources because it permits recording only in redbook.

If my source does require full fidelity and stereo, then Spin Doctor lets me pre-clean the file during capture. That is usually good enough to eliminate the need for denoising and declipping the whole recording, but manual intervention is likely still to be necessary for an LP source. Because of the nature of the work I will be doing, automatic splitting is rarely useful. In addition, if a single source tape or LP has several tracks, it is usually easier and better to do the automated denoising and declipping before splitting the tracks. In that way, the general sound will not vary from one track to the next.

First pass - Outliers

The audio should be captured at a low enough level so that overload is not a threat. In general, if you are going to clean the source, it starts out poor enough so that capturing a few db low will not harm the result after normalization. If the source has a serious problem - such as a scratch - I like to remove the extreme spikes by hand before running anything else. Without question, GoldWave is the best way to manipulate the waveform by hand. Where CoolEdit requires moving each sample or very brutal treatment (snipping or zeroing), GoldWave allows you to redraw the waveform with the mouse. Another advantage of GoldWave (especially Versions 4 and above, in beta as I prepare this) is that it will open virtually any file and provide you with access to edit any header and footer which sneaks into the audio stream. Those are the usual sources for clicks at the beginning or end of a track and various programs will introduce or eliminate them. Note that even StripWave will not necessarily handle footers if they prove to be a problem. However, I use Strip Wave when one program's output will not immediately load into another - as when a CDWAV file is to be read by DART Pro.

Ticks and pops

Step two is to remove the clicks, ticks and pops of modest amplitude. Having tried several programs for the purpose, I have settled on DART Pro as the most effective and easiest to use. In particular, the Test option in DART Pro 32 is consistent and very informative. The similar capability in CoolEdit is available only in some of its restoration tools and is much more awkward to use. Do not overclean your track! In general, if you try to eliminate all the transient noise, you will distort the music. Any automated denoising should be adjusted to leave the music at nearly full quality even if that means some noise is left behind. In particular, count on removing some leftover ticks after a noisy track is passed through DART Pro. Again, GoldWave is the way to smooth out those residuals.

Hiss, fixed tones and equalization

My choice for these operations is CoolEdit. If there is a persistent tone in the track, spectrum analysis will identify it and a notch filter will cut it down to size. But before beginning any track-level operation, I like to normalize it - again, in CoolEdit. Many tracks are normalized to 100% so that the peak level is just 0 db - maximum signal. However, I usually normalize a highly compressed track to only 80-90% so that its average level balances well with a fully normalized

track that has a wide dynamic range. Similarly, I cut the level for a track which is quiet by its nature; I don't want a serenade and a march to play back at the same level.

After reducing fixed tones - whines, hums and the like - I listen to the track for frequency balance and for hiss. In general, I am willing to put up with more hiss than others prefer, but if I must cut it I use previews in CoolEdit to be sure I don't take out too much sound in the process. Then I judge the frequency balance of the result and decide on any general curve I want to apply. Again, I tend to change the signal as little as possible and I use the very flexible options in CoolEdit for that job. (GoldWave is similarly powerful and flexible.)

Preparation for recording

CoolEdit offers excellent tools for resampling, mixing and splitting tracks and similar manipulations. In general, I finish off the track in that program and use the Edit menu option to change the file parameters. For example, I may produce a redbook WAV for CD-DA; a 22KHz, stereo, 16-bit track for MP3; and a sample at 11 KHz, monaural and 8 bits for posting at my WWW site. GoldWave will reformat for me, but not as well and without all the flexibility of CoolEdit.

This process sounds tedious and for best results from poor sources, it is. However, it can also be quite rewarding. It may take an hour or more to extract a listenable result from a well-worn disc recorded in 1903, but it may then become a valuable resource not only for my pleasure but also for scholars and music lovers for generations to come.

It doesn't sound good!

Okay, now that you've burned a Compact Disc Digital Audio (CD-DA) - why doesn't it sound good? We have to begin with how the information is recorded to understand how it can go wrong. There is a directory of sorts on each CDDA which identifies and locates the individual tracks. A track is located by its starting block and its length. There are about 330,000 blocks of 2K each on a standard blank; each one represents somewhat more than 10 milliseconds of sound. If you start a CD without specifying a track, it begins to play at the first byte of the first block and keeps on playing through to the last. There is no information on the track about the track - for that reason, it is not a 'file'. It's just a sequence of bytes.

If you write a track to the disc that is not an integer number of blocks, there are bytes to be written which don't come from your source (WAV) file. Most modern WAV editors will fix that by making sure that the number of bytes is a multiple of the block size (2K). If yours does not, there will be a click at the track transition for the random bytes that fill the last block. Solution: break the tracks on block boundary using appropriate software. (CDWAV from Mike Looijmans is posted among the files at this site. It will split a long track into short ones on block boundaries.)

If you write a disc using Track at Once (TAO), the laser is turned off between tracks. When it is turned on, it writes a gap of two seconds before the start of the next track - that is dictated by the standard but there is software which allows you to violate the standard. You can also avoid that gap by recording in DAO - Disc at Once - where the laser burns continuously through the full set of tracks. Not all CD-R's support DAO, some do not support it well, and some which do support it are not implemented for DAO by some software. In general, if you have DAO problems, check the software from Goldenhawk; it does whatever can be done along those lines on any hardware yet manufactured. Or if you are really hard up, check the DAO page in this primer.

Snap, Crackle and Pop - noisy CD's

You have just 'backed up' your favorite CD to a CD-R. You go to play it and - it's full of noise. Snaps, pops and crackles all over the place. Must be bad software or a faulty blank, right? Wrong. There's a chance that it comes from a dirty or noisy CD; if that's the case, clean it or change it. But the most probable cause is your hardware and its Digital Audio Extraction or DAE. There's a page on the subject of DAE here. This page is devoted to how to tell whether that's your problem and what to do if it is.

The telltale symptoms of faulty DAE are that the noise is in an extracted WAV file and that the amount of noise varies over the disc (usually worse as the track number goes up). There are three ways around the problem: read at a lower speed; change the reader; or transfer through analogue. Note that a drive may extract digital audio at higher than rated speed, at rating, only at 1x or even at less than 1x, the speed at which it plays CDDA. And if your preferred reader does not do acceptable DAE (or doesn't do it at all), you should try your writer as a source.

Control of reader speed depends on the extraction software. Some programs, such as Plextor Manager, give you substantial control and let you select from 1x to the drive's maximum. Most do not and automatically select the highest possible speed. On many drives, that speed is too high for the sound to be free of noise. (Why is it that Plextor - all of whose readers except the 6x do DAE at maximum speed essentially without error - is the company that helps you use the lower speed you don't need on their drives. Unfortunately, the drives that need such a Manager can't be used with Plextor's and their manufacturers offer no equivalent.)

Replacing the reader is costly, but it is the best solution. No, you do not need to use a Plextor. There are many other good drives out there and you may be able to find one that costs less and is good enough for you. However, I have seen many posts that said: *I finally gave up and bought a Plextor; I should have done it months ago.* And in case you wondered: I get no kickback from any of the manufacturers I name. They've never offered, so I haven't even been tempted.

Finally, you can try recording a WAV with your sound card instead of DAE. Just make sure that you use the CD Player as the input by selecting it in the mixer and setting an appropriate level. (Alternative: run a cable from the player's headphone out to the sound card's line in.) If you have no appropriate s/w, you can try the Windows applet, but it will only record to memory so you may not get much music unless you have a lot of RAM on your system. Any of the WAV editors (links to several are on the URLs page) will do better for you. There are several reasons why you may be unhappy doing it this way. The most serious is that you are using digital-to-analogue conversion (DAC) in your reader and ADC (guess!) on your sound card. Each of those is imperfect and the artifacts that they create compound one another. A less important difficulty is that the sampling rate you get depends on your PC's clock, so the playing speed may be slightly higher or slightly lower than the original. However, because of the interaction of similar frequencies in non-linear processes (boy, he knows big words), resampling can produce artifacts such as subtle beats or recurrent noise. Still, it costs nothing to try this approach and if it works, don't fix it.

Skips and repeats

The CD writer needs a continuous supply of data. If something interrupts that flow for a significant time, the buffers can empty and underrun. If the interruption is shorter, a buffer - particularly that in the reader - may not cause an immediate underrun but may supply either zeroes or a repeat of the last information. In that case, you get a silence (zeroes) or a repeat lasting a fraction of a second.

Of course, the right way to fix such a problem is to ensure that the buffers do not empty. If you are writing on the fly from a reader, check the lights for regular operation. If you see a significant inactivity, then a burst you may have spun down (typically, between tracks) and if spinup is not fast enough the flow is interrupted. Another cause is a slightly damaged source disc which must be reread on the fly. If the problem arises when writing from the HD (very rare, indeed), then something is interrupting HD access and you should check for the usual villains: FastFind, anti-virus monitoring or another concurrent program.

If the problem occurs when writing on the fly and you have no remedy for the cause, extract the files to the HD and burn from there.

What's this DAE stuff, anyway?

There may be nothing about CD-Recordables more confusing than Digital Audio Extraction. So, let me try to explain what it's all about and why it is an issue. As usual, corrections are invited; I am not an expert in this field and can be accused of sharing my ignorance, not my knowledge.

In the beginning, the compact disc format was defined for digital audio - CDDA. The rules for CDDA are defined in a standard called The Red Book (not the Little Red Book - that's the one with Chairman Mao's wisdom). CDDA says that there's a small file (in the computer sense) on the disc which identifies the tracks that follow. Then there are the tracks with a precisely specified sequence of bits in which two channels of audio at 16 bits per channel and 44,100 samples per second are interleaved. Those tracks are not files - they are simply tracks with strings of bits. The only files on the disc are those little pointers which tell the laser where to go to get a track, how long it is, etc.; they could hold a bit more information, but since few CD players would use it, few publishers bother to put it in.

The difference between a track and a file is worth looking at. After all, a WAV file may be an interleaved pair of strings with the same properties as the track. But the WAV file contains information about its contents, for example, the fact that it's sampled at 44.1 KHz; it also has an associated CRC (cyclic redundancy check) which lets the retrieving software determine whether it has been read correctly. When stored on a CD-ROM, it has an extra layer of error correction; the track has a minimal level in the hardware that we'll discuss later. In short, the WAV file carries overhead, 'wasted' space, with information not in the CDDA track.

When a CD player recognizes an audio track, it routes the bit stream to its DAC - digital-to-analogue converter. There the minimal layer of error correction is applied to patch up some errors in a disc that's in good shape (it cannot do much more) and the analogue sound output is generated. Since an error is usually in a single sample and a sample lasts less than 25 microseconds, you're not supposed to notice it. However, we would like to capture that digital data stream to our system in order to record it onto a CD-R and perhaps even to edit it. That means we wish to take raw data extracted from the audio track and convert them into a file that can be stored, modified and written with a computer. To do that, the reader must feed the digital data to an output that the computer can read. If the drive will provide that information, it is giving you Digital Audio Extraction, DAE. Not all CD-ROM readers do that, so some simply cannot provide the signal that's needed. You can run any software you wish, if it won't provide the output, the output won't be there.

In some cases, the firmware of the reader can be changed to provide DAE; in others, it's a hardware problem - period. In addition, something must tell the drive that the computer wants the DAE output - that the signal should be diverted from the DAC to the digital output. It can also tell the reader how fast to try to read the CDDA stream. Another critical need when the computer is trying to convert the stream of bits into a file is that they come through on time and in sequence. Again, if there's a bit of inconsistency in the audio stream, we probably won't hear it; but the computer is not as forgiving as the human ear and it wants each sample to be meaningful, not just 90% or 99% or even 99.9%. So where a sample or two can be dropped every so often while listening to a CD, none can be lost if the data are to go into a file.

Thanks to all the error correction on digital data, a CD-ROM reader may run at 4 or 8 or 20 times the speed that a CDDA is supposed to play. But without that correction, the audio stream may not be able to run at full speed. In fact, on some supposedly high-speed CD-ROM drives, CDDA may not even extract at 1x - 150 KB/sec. In other cases, there are so many errors that the resulting file is unacceptable. Again, the problem is not the software but the reader's hardware and/or firmware. All that the software has done so far is to command that particular reader to read CDDA as digital data to the appropriate output port, receive the data stream, and slow down the reader if the data are not usable to build a file.

Finally, the software comes into play to do a bit more. It takes the received data stream of a track, converts it to a WAV file and writes it to the hard drive if that is being used as an intermediate. By writing to the HD, the DAE is allowed to be slower than would be needed to write directly to the CD-R. If you do not go to the HD, then noise or something else may slow the reader enough to underrun the buffer on the writer (it is not getting data fast enough to keep writing) and your CD-R is another coaster. Writing the data to the HD buffers the information so that varying read speed is acceptable. In addition, you can listen to the WAV file that the software records to confirm that it's good enough to write to your valuable blank. Of course, if you are sure that the reader and the CD-ROM are good, or if you are in more of a hurry to get a copy than you are in need of a good one, you are free to burn 'on the fly' - directly from the reader to the writer.

Most readers do not use even the little error correction available when performing DAE. Only the Plextor family consistently corrects the data stream on the way to the digital port. And few drives but the Plextors extract CDDA as quickly as they do digital data. Those using Plextors for DAE are not praising them because Plextor is paying them; they are simply sharing with you their best advice on saving money and time. You are free to ignore their (our) recommendations, but don't expect a whole lot of sympathy when you do. And I am sure that they/we will be happy to let you know when another vendor delivers quality signals from a quality product. Until then, we will have to restrain ourselves from saying: 'I told you so' when you find that you spent more for your 24x EIDE than you would have for a 4+ and cannot get decent DAE even at 1x.

DAO - Disc at Once

Most of us are familiar with audio CD's and the tracks of music which they contain. When mastering a CD, those tracks may either be laid down separately, with the laser turning off between each pair; or in a continuous flow. If the laser writes Track at Once (TAO), a gap of two seconds is introduced (set by the Redbook standard) to ensure proper synchronization. If the tracks are written continuously in DAO, no gap is required. There are schemes for approximating DAO by adjusting the gap in spite of the standard; the results may or may not be satisfactory depending on the player used.

DAO also has a role in data recording: masters for mass production are required to be in that format. Jerry Hartke, President of Media Sciences, Inc. <http://www.msscience.com/> has provided the following on why pressing plants require DAO masters. Note that going through the 8-mm intermediate adds to the cost of mastering and that it will also (slightly) change the size of the disc being pressed.

Mastering facilities must have perfect source data or their laser beam recorder will abort and ruin an expensive glass master. Track-at-once recording leaves "link-blocks" at the end of lead-in and at the beginning of lead-out. These are read as defects, or errors, by the LBR. Disk-at-once recording has no link blocks. Many mastering houses have learned from a rather gruesome history, and transfer information from a CD-R to 8-mm tape using methods that remove problems such as link blocks. Mastering is then conducted from the perfect image on the 8-mm tape, not from the "bad" CD-R.

Interestingly (at least to me), DAO costs a small amount of space on the finished disc. In an experiment with a single data session, a closed session adds 4 KB to the size of the files. A disc closed in a single session has the same 4 KB penalty. Each additional session takes 604 KB even if there are no data to be recorded. DAO costs 300 KB. While audio has not been tested, it may be worth noting that two seconds corresponds to about 300 KB.

Not all CD-Recorders are capable of DAO and not all software supports even those that can do it. The best information on DAO recording is provided by Jeff Arnold at <http://www.goldenhawk.com/> If you believe that you will want DAO either for mastering pressed discs or for making CDDA's without gaps, check there before you pick your recorder.

WAV Files

A WAV file consists of three elements: a header, audio data, and a footer. The header is mandatory and contains the specifications for the file - information on interpreting the audio data - and optional material including copyright. The audio data are - um - audio data in the format specified by the header. The footer is optional and, if present, contains other annotation. I like to call the combination of header and footer the 'envelope' of the data. Usually, the data in a WAV file take the form of PCM bit streams. A simple one would be 8-bit samples taken 11,025 times per second from a single (monaural) channel. Then within the envelope would be a RAW file - the sequence of those 8-bit bytes in order and ready to be fed to a sound card's Analogue to Digital Converter (ADC) to play on your speakers. There is a well-defined way to put the other formats into the RAW file, which then goes into the envelope containing the information about how to make sense of the bytes which follow. What gets written to a CD-R is very much like the RAW file, with one important modification: the order of bits in each byte is reversed. So your CD-R authoring program can take a 44.1-KHz, 16-bit, stereo WAV file (i.e., in redbook format) out of its envelope, switch the order of bits, and write it to your disc.

Other formats can be stored in a WAV file. Apart from changing the PCM parameters, you can put a compressed stream into the envelope. Whatever is in the WAV file, your CD-R program must convert it to redbook in order to write CD-DA (Compact Disc, Digital Audio); and if the disc is not in CD-DA, it won't play on an ordinary audio CD machine. In many cases, converting from another format to CD-DA will require substantial processing. As a result, you may not be able to keep up with the writing speed of your CD-R. Since the program does not know how fast your computer is or how much processing will be needed, ECDC (at least) always converts a non-redbook WAV to redbook in a temporary file before writing it to your disc. ECDC also doesn't want to run out of disc space during that conversion, so it converts one file at a time, writes it to the CD-R, then converts the next. Writing one track at a time, it obviously cannot write DAO.

Now we get to the messy part. Not all programs follow all the rules all the time. As a result, the envelope may not be quite what it should be to make a valid WAV file. When that difference confuses the next program reading the file, it may refuse to open the WAV file or it may need to be told the format of the audio data, or it may just interpret some of the envelope information as audio data. For example, the 16-bit version of DART Pro appears to convert a valid header with annotation to an invalid one and does not recognize footers at all. If you feed it a WAV file from a program like GoldWave, which creates an envelope with valid annotation in the header and footer, it produces an output file which is almost unreadable. Fortunately, GoldWave will open it - when you tell it what format it has. Unfortunately, the envelope information DART Pro did not respect are still present - as data within the audio file. Those can be heard as initial and final clicks and can be seen in the first and last milliseconds of the file. If you snip those data out of the file, you can now Save As WAV and get a proper file again. Another option is to run StripWave. That program will strip the footer (it is unnecessary) before you start the offending program. Run it after that program has mishandled the header and it will strip out the excess audio - the initial click. NOTE: DART Pro is not the only offender and the 32-bit version appears to leave annotation where it is.

Another problem program is Plextor Manager. It also fails to put the right information into the header so that data appear in the audio stream. Again, StripWave will fix it easily - but you may not need to do so. The effect of the faulty envelope varies depending on the next program to see it. Some authoring programs are able to strip the faulty data the same way that StripWave does, so the simple error will not show up on your disc. Or, if it does, it may be lost when it is read from the disc. So once you have a bad envelope, you may or may not get an initial click or a final click. And you may get it on DAE but not when you listen to the disc, or hear it when you play the disc but not see it on DAE. If you do have the faulty file and process it before writing it to CD-R, the processing (WAV editing) program may refuse to open the file, may embed the data in the audio stream so that the click is guaranteed to show up later, or may correct the problem for you.

The bottom line is that you either set up a procedure which you know works for your software and your needs, or you figure out for yourself when and how you strip the faults from the WAV files. Note that StripWave is shareware and links to it and to other interesting tools are in the URLs at this site.

Sound Balancing

Your perception of the loudness of an audio track depends on several factors. One is the absolute loudness, another is the dynamic range, a third is the kind of material.

The maximum signal you can record on a digital system is fixed by the number of bits available. If an analogue system is overdriven, the result is a gradual increase in distortion; digitally, saturation is a wall and even a slight step beyond it is highly disturbing. (There are, of course, some kinds of 'music' which are designed to be distorted; I have nothing to contribute for them.) So the maximum level you want to set is something which does not go beyond maximum loudness.

The second factor is dynamic range - the range from the loudest to the quietest passage in the program. At a symphonic concert, that can be as high as about 100 db - for a very quiet audience and some very loud passages. CD recording at 16 bits offers 96 db range. A very good tape deck can deliver about 60 db. But in many environments, use of wide dynamic range is undesirable, so the actual dynamic range on a recording may be compressed to 40 db or less. Different recordings will vary in the amount of compression used - and the average sound level depends more on that than on the absolute maximum you have to keep to 100% (16 bits).

The third factor is the program content. I don't know about you, but I want a march to sound louder than a lullaby. So even after you set the maximum and mean levels by normalizing and compressing, you have to consider how loud you want the result to be.

All is not as bad as this sounds. In general, a given label will be pretty consistent across its recordings of a single group or even of a single kind of music. But as you try to mix a greater variety, you face the problem that the software doesn't know what compression makes sense (or how best to implement it) - and it certainly does not determine how loud you want a given track to seem on a given compilation. The automatic level setters do what they can, but they are basically limited to setting the maximum level. The rest must be done to your taste by you. A good WAV editor, such as CoolEdit Pro will let you shape the compression on a track-by-track basis as well as permitting you to adjust level and keep from hitting saturation. It will also denoise, alter channel and frequency balance, and do many wonders - with a lot of time and effort on your part. To avoid that, choose the tracks for your compilation with some care and take what you get with automatic balancing.

Correcting Pitch

In most cases, the pitch of your source is correct - that is, the recording is at the pitch and has the duration intended by the performers. However, the farther back the original was made, the more likely that it is now at the wrong pitch regardless of the form in which you find it. In the first three decades of sound recording, a '78' would be recorded (usually) somewhere between 60 and 95 rpm! Later, errors became less frequent and less extreme, but they still occurred. The best way to solve the problem for transfer to your computer (and to your CD-R) is to change the pitch at the source - with a variable-speed turntable or tape player. There are also variable-speed CD players, but they offer control only at their analogue outputs; whether you would prefer to record from a corrected analogue signal or from off-pitch DAE is up to you. If you are using a variable-speed player, then, you simply adjust it until the pitch is right to you and record through the line input of your sound card. So from here on I assume you're starting from a fixed-speed source and you have to correct pitch digitally. For convenience, all sampling rates below are in KHz.

Determining pitch

Well, this should be easy, right? Maybe so, maybe not. It depends on your source and on your ear. First, we assume that whatever pitch error exists, it is constant through the recording. That was not strictly true in the earliest days, but correction for changes during the recording is beyond the scope of this primer (i.e., I don't know how I would do it if I had to).

Step one is to record a short selection in which the pitch can easily be determined. I recommend doing that in monaural at a reduced sample rate, perhaps 22.05. It makes handling the file easier and should serve your purposes as well as working with a file four times the size. That file is loaded into your favorite WAV editor. For this purpose, I find GoldWave the best choice because its speed controls are the most comfortable for me. All you need to do in GoldWave is to adjust the playback speed until the pitch on your test file checks with your pitchpipe or other reference. Now note the error - the difference between the playback speed and that at which you recorded. For reference, a semitone is about 6% in speed; half an octave is about 40%.

Resampling

One way to correct pitch is to record at the nominal sampling rate you will be using, say 44.1, and then to resample to correct. For example, if your 22.05 test played back best at a rate of 23.3 KHz your WAV is about a semitone flat. In GoldWave, you can simply record the whole selection at 44.1 sampling, then transpose it down a semitone. The same thing is done slightly differently in other editors, but the effect is the same: approximately six samples are thrown away out of every hundred and the others are adjusted to fit. GoldWave gives you no control over how that adjustment is done and uses a simple, quick algorithm for the purpose. CoolEdit Pro and other WAV editors provide control in the form of pre- and post-filtering and relatively complex interpolation schemes. Since that interpolation is done 44,100 times per second per channel in this example, even a fast CPU will take substantial time to do a good job. Only you can determine how much time you want to spend processing and how important the errors are.

Off-Sampling

(Note: this term is my invention. If you have a better one, please let me know.)

Most audio capture programs allow you to pick from a few sampling rates. Typically, they are 48, 44.1, 32, 22.05 and lower. However, Mike Looijmans' CDWAV is not so choosy. In addition to its presets, you may type in any integer sample rate you would like. If the playback is best at 6% higher speed than recording, then you can record at a sampling rate 6% lower than your target and if you have a way to interpret the result as 44.1, you would be on pitch. With the right tools, that's straightforward.

Record in CDWAV at your shifted sample rate - here 41.5. Record the selection as a .PCM file. PCM is a raw format (other extensions are RAW and SND). Now, open that file as PCM in CoolEdit and lie to the program - tell it that the rate is 44.1. In this case, lying is good for you. All you need to do now is to clip off the click in the header (a real PCM doesn't have a header, but CDWAV recorded it with the WAV envelope, which has to go). Save the file as WAV - and you're done. No resampling, so no approximations or delays.

Sound Cards

To tell the truth, this subject is peripheral to CD recording and I've been reluctant to add it. But questions come up so often that it seems to be necessary - and for many, the performance of the sound card in the PC is essential to getting good results on CD-R.

First, I shall neglect some of the factors important to users of computer audio for other purposes. The primary interest for CD-R is the ability of the sound circuitry to encode PCM - Pulse Code Modulation - from analogue. There is a special case in which the sound card serves a purely digital function - S/PDIF; it will be dealt with at the end. In addition, the focus here is on CD-DA, Compact Disc - Digital Audio.

Pulse Code Modulation

In PCM, a collection of computer bytes corresponds to a single, specific point on an analogue waveform. The frequency with which those samples is taken is the sample rate, usually designated in samples per second but written (imprecisely) as KHz (KiloHertz). Of course, one issue is whether there are one or two channels of audio involved - monaural or stereo sound. Another is how many bits are used for each sample collected; the usual values are 8 or 16. If the encoding is set up to collect 8,192 samples per second (8 KHz) from a monaural signal and to encode at 8 bits per sample, the resulting PCM stream takes 8 K bytes per second (bps). If samples are collected to be recorded in redbook format at 44.1 KHz, stereo and 16 bits, the PCM stream takes 176.4 Kbps. That rate is labelled 1x for audio. (Because of error correction, the corresponding data rate is 150 Kbps.)

Noise Floor

One key function of the sound card is to convert the analogue input signal it receives into digital data. The quality of the result depends essentially on two factors: the noise floor and the accuracy of ADC. Noise floor means the noise introduced by the card itself and the leads which provide input when there is no signal. It determines a level of hiss due to the card as opposed to what may be received from a tape or other input. For comparison, a very good cassette tape can deliver about 60 db SNR - Signal to Noise Ratio. That means that its noise floor is 60 db below its maximum signal. Because of quantization (the resolution capability of a digital signal) the SNR for 16 bits such as from a CD is about 96 db. A sound card which puts its quality into gaming and MIDI may deliver something like 50 db - in other words, it may provide more noise than a good, Dolby B cassette. A less expensive sound card which focusses on WAV quality may deliver 75 db. Where those cards typically cost less than \$100, for \$200-500 or so, one can buy still higher quality, up to about 82 db. Beyond that, the experts use outboard converters to avoid the electrical noise inside the computer's case.

ADC - Analogue to Digital Conversion

The input to a sound card is typically an electrical analogue of a sound wave. The card's job is to convert that signal to a digital one of the proper number of bits and at the proper rate. The conversion process is more or less accurate depending on how well the card is designed and built. Not surprisingly, the cards with low noise floor also have high-quality ADC. High-end cards may encode to 20, 24 or more bits. That allows substantial editing without losing quality before the bit depth is reduced to 16 for writing to CD-R. In general, mastering s/w will create redbook audio from 8- or 16-bit samples, but not from others.

Sample Rate

Similar to but worse than bit depth, sample rate is under your control and must be chosen with care. The highest frequency which can be captured in digital form is half the sample rate: 44.1 K samples per second cannot record more than 22.05 KHz audio. However, there are significant effects at lower frequencies, so it is desirable to work at a higher sampling rate when practical. Most DAT recording is done at 48 KHz; 16 and 32 KHz are common choices for lower-fidelity sources; professional mixing is often done at 96 KHz. However, those rates are not easy to convert to 44.1 for CD-R, so they will require processing outside the mastering software before they can be burned to CD-DA. Like

adjusting for non-standard bit depth, that is a job for a WAV editor. Because the file size and the difficulty of editing increase with sample rate, processing low-fi sources at 22.05 KHz monaural is much more efficient than the same process in redbook - 44.1 KHz stereo. Fortunately, most mastering s/w will convert that to redbook without bothering you (though not on the fly).

File Formats

Rather than repeat the information elsewhere in this primer, I will simply summarize it. A PCM file may be raw or wrapped in an envelope with a specific format name. Raw files typically have extensions PCM, RAW or SND. A raw file carries no information except the stream of bytes. If you put data on that stream to specify details such as sample rate, bit depth and mono/stereo, then you must use an envelope or wrapper such as WAV, AU or AIFF. That envelope permits other information (e.g., copyright) but does not require it. Considering only the WAV file (it's all I know), the wrapper can also hold compressed files, but those are not PCM. ADPCM and MPEG Layer 3 are typical compressed formats.

Inputs and Outputs

For convenience and for mixing, the sound card will accept several different inputs. Typically, one comes from CD-ROM, one from Line In (an external connector) and one from Microphone (usually monaural). There may be additional CD-ROM connectors on the card, but unless there are separate sliders for them on the mixer, they probably are all shorted together and simply provide different types of plugs. You should not connect two signal sources to a single input of your sound card. It probably won't sound good, but it definitely will not load the sources correctly. If you want to connect both your writer and your reader to a single card, you can find a card with distinct inputs (separate sliders), install a switch for the two inputs or use a cable from an analogue output such as headphone to Line In.

In order to hear a signal from your sound card, you must enable it on your mixer. However, what you hear is not necessarily what you will record. In order to record that signal, you must enable it (and set the record level) on that same mixer. On the standard Windows mixer, you go to Options, Properties, Recording in order to reach the controls you need. The next time you boot, your settings are lost and you must go there again! Other mixers make the controls more accessible and sticky, but they are not usually available except with their corresponding sound cards. An Ensoniq mixer is available only with a Creative Ensoniq card, for example.

S/PDIF designates a direct digital connection between a device such as a MiniDisc player or a CD-ROM and your computer. It is usually implemented in a high-end sound card. Note that not all outputs of CD-ROM drives which say digital or look digital will mate correctly with an S/PDIF card. Like DAE, S/PDIF avoids DAC and ADC altogether and allows you to remain in the digital domain. In addition, many sound cards implementing S/PDIF also convert from 48 to 44.1 K samples per second on command. Another advantage is that an S/PDIF input allows you to connect that second CD-ROM safely. But by now, we have gone beyond the scope of a primer. (And anyone who suggests that that means it has exhausted everything I know on the subject has been cheating!)

Recommendations

As usual, I make no comments on equipment I do not know. I have replaced three sound cards since getting into CD-R: a Creative AWE 32, an AWE 64 and a Yamaha. (The last meant replacing the motherboard on which it was located.) In all cases, I found both the noise floor and the quality of ADC insufficient; the Yamaha also proved almost impossible to control with the standard mixer. My current systems use an Ensoniq PCIAudio (no longer available) and its first cousin, the Creative Ensoniq PCIAudio. The cards and the mixers are different (and not interchangeable), but both deliver high quality at low cost. I have not tried the Creative Live, a Turtle Beach, Card D, or any of the dozens or hundreds of alternatives.

A Note on Video

The two applications which drive many aspects of personal computing are video and games - and they are obviously closely related. With the availability of inexpensive capture devices such as IOmega's Buz, many are jumping into video and becoming frustrated in the process. This note is intended to address the beginning user of digital video.

Like audio, most video begins with an analogue signal. If your camera generates a digital signal directly, most of this page does not apply. For the rest of us, some capture device is required which converts the analogue information to digital as your sound card does with audio. In general, the capture is to a lightly compressed format. (Without compression, you would have to record at around 30 MB/sec. That would both load your system very heavily and rapidly use up all available space on a hard drive.) The lightly compressed format is usually compressed further offline with a codec (encoder/decoder) that limits the quality of the resulting file. One popular format is whitebook MPEG, which is used to generate VCD. In that form, the data rate is about 1x in CD-ROM terms, meaning that the signal has been compressed 200:1. In that format, a CD-R can hold about an hour of video. At 50:1 compression, a CD-ROM can hold only about 15 minutes of video. Obviously, the more compression used, the lower the resulting quality of both video and audio. In order to play back a video, the playing computer must have the same codec installed as was used to create it. Many such are available and whenever you install one to encode, you will be able to play it back. However, the person to whom you send it may not have that codec - and in that case will need to find it and to install it before she can watch your product. Intel has several Indeo codecs and there are many others with modest advantages from one to the next. Those codecs usually generate AVI files on a PC (MOV is more common on Mac). AVI and MOV are different envelopes wrapped around the same raw data stream - as WAV and AU are envelopes for audio data. Because of the variation in AVI codecs, it may be most convenient to compress your video to MPEG, which is consistent across platforms and has a standard codec. You can produce MPEG with other products, but many find the Xing encoder most convenient and reasonably priced.

A key factor is the size of the file you generate. Since your original recording will be an AVI even if you are going to compress it to MPEG, you are limited by the specification of AVI. The definition of that format limits it to 2.1 GB total size. While that seems immense, in video terms it is quite limiting. If your original capture is compressed 50:1 (which is very high), an individual capture must be less than 45 minutes. Compression to MPEG is similar to compression to MP3 in audio - and similarly makes editing in that format almost impossible. In order to edit your video, you will need to operate with an AVI and use a tool such as Adobe Premiere or ULead MediaStudio. Editing is very demanding on your system, so be prepared to add large, fast HD's, plenty of RAM and powerful processors if you want to deal with ten minutes' worth at one time. However, it is important to recognize that a high-performance capture card is of little value for home video. If you have a very high quality digital signal, you may throw away 90% of it in compression. If you start with a half-frame source, you will 'only' throw away 60%. The results will be indistinguishable. (Obviously, the rules and the systems are very different for broadcast quality.)

No one can tell you what picture quality is good enough for your purposes. But once you know what you want, do not overbuy to achieve it. High quality in capture means major system revision; modest quality will let you know how much you are willing to invest and give you satisfying results.

MPEG - Picture and Sound

There is a set of standards for video labelled MPEG. Some apply to video; others apply to audio. This note will cover only the basic features; for details see the usual sites as linked under URLs. Like all lossy compression schemes, the quality of the output of MPEG encoding is less than that of the original video or audio source. In general, the user has enough control to select a balance in compression between loss of quality and reduction of file size, but one should not hope for miracles. Experimentation is needed to determine what looks and sounds right for you.

Video

MPEG video was to be available in three flavors. MPEG 1 can be produced and played back (encoded and decoded) in software alone and on a fast system can provide reasonable quality at 352x240 pixels or about half of the usual minimum screen size. MPEG 2 is more efficient, supports higher resolution and requires specialized (and costly) hardware to encode and moderately priced hardware to decode; it is currently used for DVD playback. MPEG 3 is still higher quality and was intended for high-definition television (HDTV), but it has been suspended until requirements can be fully defined.

MPEG 1 is readily produced on a PC and is suitable for recording from and to VHS videotape. 'Readily' must be taken with a large grain of salt, however, since software encoding takes many times real-time. The MPEG codec is quite efficient in storage requirements and provides a high-quality picture at reasonable file sizes by video standards. A properly encoded MPEG 1 file can be turned into VCD (Video CD) with Easy CD Creator and played on a suitable set-top player. More useful for many is that it can be played on a personal computer and even accessed by a WWW browser with a suitable helper application. Many other codecs (coder/decoder) are available for video; their products are usually in the form of avi or mov files depending on whether they are packaged in Windows AVI or Mac/Windows QuickTime envelopes.

Audio

MPEG Layer 1 is a moderately compressed audio format which is used in some video to reduce decompression load. Layer 2 is usefully compressed and can be generated easily in software encoders. The usual extension for MPEG Layer 2 is .mp2. Layer 3 is still more highly compressed; the shareware program l3enc (with its companion l3dec) appears to be the most popular choice for that process. Other mp3 encoders are available or on their way.

MP2 and MP3 playback require an appropriate application; usually both are provided in a single program. One with several advantages is the inexpensive shareware WinAmp (and the companion MacAmp) from Nullsoft (<http://www.nullsoft.com/>) It offers a great deal of flexibility so that good results can be obtained on a slower computer; a graphic equalizer with some compression of dynamic range; and even the ability to export an MPEG file in decompressed form in the wav format. Since CD-DA is uncompressed, an MPEG file cannot be the starting point for recording an audio CD-R. An MPEG can be stored on a disc, but to make it useful in a CD player, it must be decompressed to wav before burning.

Streaming

Streaming video and audio refers to the ability to play the picture and sound as it is being received. In general, that requires special capability on the server side and is not available from a CD-R. The formats used for streaming include RealAudio, RealVideo and TrueVoice. The RealMedia capabilities are very inexpensive, but require that the user either download or buy the player program. Efficiency and quality for video are less than MPEG, but more than some video codecs; RealAudio options are quite attractive, but no converter is available to produce wav files from ra. In addition, the author of a RealMedia file may mark it so that from a streaming server it cannot be saved to disc by the user.

MP3 and the Perfectionist

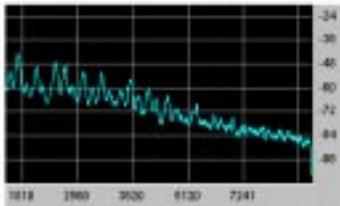
To begin at the beginning, MPEG stands for Moving Picture Experts Group. That body establishes standards for digital video and audio. We are concerned here with the standards for the audio layer in the MPEG1 format. MPEG2 is in use today, but is not related to the MP3 files; they are MPEG1 Layer III audio. Simplifying the situation, different layers impose different loads on the decoding software - the program which converts the MPx file to uncompressed Pulse Code Modulation (PCM) audio to drive the reproducer. Layer III - MP3 - is consistent with modern low-cost dedicated packages and with Pentium-class CPU's. Layer 2 - MP2 - is less demanding; its performance is not adequate for it to be considered quality reproduction. In theory and usually in practice, system and software reproducing a given Layer will handle any lower Layer.

Simplifying again, the MPEG standard for a Layer specifies the playback of a file encoded for that Layer and leaves encoding to the developer. MPEG audio employs perceptual encoding and is lossy. That is, it compresses the data stream by throwing away information which the encoding algorithm 'believes' will affect the listener least. The decode side of the codec (code/decode algorithm) is usually pretty simply implemented from the standard; there are some differences which will be discussed below, but in general the playback algorithm is not an issue.

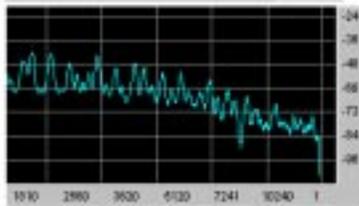
The encoder is a different matter altogether. There are three parameters input to any encoder to control the way it process the file: channels, sample rate and bitrate. Channels simply means monaural or stereo; in general, the encoder will provide 'stereo' channels (identical content) from a monaural original, combine two original channels into one, or leave the count unchanged. Sample rate is simply the number of samples of the data per second. Bitrate dictates the size of the encoded file. Those factors are interrelated.

Rates in common use for digital processing include 44100, 48000 and 96000 samples per second. Even though it violates convention, those are usually shortened to 44.1, 48 and 96 Ksps - which will be done here. (The standard for CD encoding is published in a book with a red cover; uncompressed, 44.1 Ksps, 16-bit, stereo signals are conveniently referred to as 'redbook' in its honor.) The more serious problem, representing Ksps as KHz, will not be accepted here. As is well known, the maximum frequency which can be encoded digitally is one half the sample rate, providing an incentive to take more samples to preserve high frequencies. In the octave below that limiting frequency, phase shift can substantially alter waveforms even though amplitude loss may be acceptable. For various reasons, it may be desirable to have the sample rate in an MP3 file different from that of the original. Many MP3 encoders will resample for the user. Some are limited to commensurate rates such as downsampling 96 Ksps to 48. Others will handle incommensurate rates and can accomplish the more difficult task of resampling 48 Ksps to 44.1. The quality of resampling may be significant here. Simply put, samples are created or destroyed in resampling and that may be done with slow, careful algorithms or quick, simple ones. Needless to say, the effects are audible if one is willing to listen.

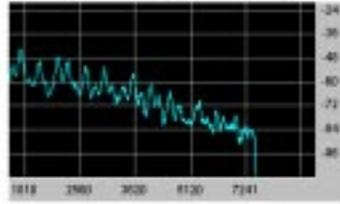
Bitrate dictates the size of the finished MP3 file per minute of audio. CD-quality audio is defined in the standard and requires about 175 Kilobytes per second (175 KBps). MP3 bitrates are specified in Kilobits per second (Kbps - note the lower-case 'b'). The rate most often used for CD audio is 128 Kbps - which corresponds to 16 KBps or about one eleventh of the redbook rate. Other rates are frequently used, usually 64 or 256 Kbps though lower compression (higher bitrates), but 128 Kbps is usually assumed. As you would expect, throwing away 80-95% of the original information will impact the sound of the MP3. There are many encoding algorithms on the market. Some, such as BLADE, are optimized for lower compression than usual and will be neglected. Of the others, both subjective and objective tests indicate that the patented algorithm developed by the Fraunhofer Institute is the most pleasing. That is unfortunate, since it is by far the most costly to license - \$300 versus \$15 or less.



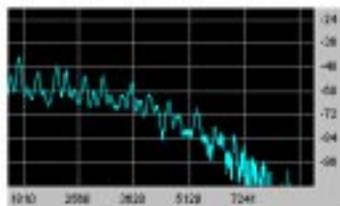
**Figure 1:
Original Wave**



**Figure 2:
Fraunhofer**



**Figure 3a:
Encoderx**



**Figure 3b:
Encodery**

Quality is a lot easier to hear than to describe. Absent an objective measure, there are two parameters of primary interest: preservation of audio spectrum and avoiding artifacts. In practice, they go together and codecs which maintain spectrum best tend to introduce the fewest audio disturbances. The most common form of artifact is a metallic tone which includes some narrow resonances; it's one of those things you'll know when you hear it but it defies description. It appears to originate when signals near the maximum frequency are encoded differently from those near them in pitch. The Fraunhofer codec gives very nearly flat response up to a frequency at which it cuts off abruptly. Another codec may nominally extend the response another half octave, but in the process substantially distort the response curve. Figure 1 shows the spectrum of a 22.05 Ksps monaural original file. Figure 2 shows the same file after encoding with the Fraunhofer high-quality option and decoding back to PCM. Figures 3 are the same with other algorithms. Note that the spectra are quite similar up to about 1.5 octaves before the cutoff at 11.025 KHz. The Fraunhofer remains nearly flat to about 8 KHz, then falls off a cliff. Another algorithm provides some output in that half-octave, but is 12 db down where the Fraunhofer is within about 1 db up to its cutoff. In short, the Fraunhofer provides good

encoding within its frequency range where the alternative generates artifacts amid some signal on significant overtones. Needless to say, the perception of the sound of the samples is very different even to an untrained listener.

If artifacts are audible in the encoded file, the only solution with a given algorithm is to limit the frequency response of the original. Needless to say, that limiting must be more severe, perhaps by a full octave, with an inferior encoder. Another solution is to increase the bitrate or, in the case of variable bitrate, to increase the maximum bitrate, but either of those would increase the file size. There are listeners who find even an inferior codec's 128 Ksps file identical to a redbook source. With any audio acuity at all, a listener should be able to recognize loss in the Fraunhofer at that rate in A/B comparison with the source; she will be significantly disturbed by the artifacts of a poor encoder, though she may be unable to describe the faults in the sound.

One point in all of this is that evaluating an MP3 player as a high-fidelity device is not as simple as evaluating CD players. It is necessary to know not only the sound of the source - often, an audio CD - but also the encoding algorithm that was used. Many of us are familiar with the influence of engineering when transferring a master analogue tape to CD; the same original can sound quite different depending on the equipment used and the engineer's judgement. Fair comparison of two reproducers requires that they be loaded with the same MP3 file, not simply compared using whatever file it may have against the original. There are MP3 playback products which 'compensate' for the faults of the encoder they are intended to follow. That is a kind of inversion of the deservedly deprecated Dynagroove process in which the original was distorted to correct for faulty reproduction. As with any high-fidelity product - video, audio or otherwise - the ideal is to maximize quality of each element in the chain, not to correct in one for expected faults in another.

Video CD

VCD is a well-defined and specific format, much like CD-DA for audio. Like audio, it requires a precise implementation of file structure so that it will play back correctly on a VCD or DVD machine which does not plug into a computer.

The VCD has one or more MPEG-compressed data streams recorded without the usual ECC of files. As with CD-DA, it is assumed that any errors will be unnoticed in the result. Unlike CD-DA, the data stream looks to be a true file and can be read (as a DAT) to your computer. However, since it does not have ECC, it can be too large to write back to a disc as a conventional file - and would not be recognizable in a VCD player if it were. Many MPEG players will play back DAT files as well as MPG's, even though they are slightly different internally.

To create a VCD, two steps are required. First, a whitebook MPG must be generated, then it must be recorded with software designed for that purpose. There may be boards or devices which create whitebook MPG's directly, but since those files are very difficult to edit, the preferred starting point is an editable AVI. The AVI from a capture device is most often edited with ULead's Media Studio or Adobe's Premiere; the latter is the choice of most serious users. The edited file can be saved with your choice of codec or those using Premiere and the Xing plugin can render directly as a fully compliant MPG. ULead's format is not fully compliant but is acceptable to VCDC from version 3.5b. Both hardware and software encoders for MPEG are available. The Xing software implementation is relatively fast and consistently generates true whitebook code when the VCD option is selected.

The VCD format requires specific folders and files and the DAT itself must be recorded without ECC. Therefore, a dedicated program such as VCD Creator (a component of Easy CD Creator Deluxe) is required. VCDC will not record a non-compliant MPG file, so you will not waste a blank by discovering non-whitebook format after the fact. Note that relatively few VCD or DVD players will handle CD-R. The Sony models are reported to accept them without complaint. Similarly, first-generation DVD-ROM players do not read VCD's recorded on CD-R. Second-generation DVD-ROM is MultiRead and should handle either CD-R or CD-RW. That is particularly advantageous in the VCD format because of the time and effort required in preparing and editing material. With audio, one can simply DAE the source to WAV files, then rearrange them and burn them to CD-DA. There is no equivalent compilation capability in VCD at this writing; one may either generate one's own files from AVI or copy a disc in its entirety. If you are creating a VCD, it makes sense to burn an RW for evaluation before making one or more CD-R's for use.

Correcting an MPEG for VCD

One question raised frequently is how to put MPEG files from various sources onto a VCD. Frequently, the answer is: you cannot - at least, not practically. That usually results in cries of protest: someone must have a simple, freeware program to 'fix' the file. Not so.

VCD is a strictly defined format - just as CD-DA audio is. If your original audio file is sampled at 44 Ksps instead of 44.1 Ksps, the only way you can make it into an audio disc is to resample it to 44.1. Similarly, NTSC video for VCD must be 29.97 fps. Other rates will play as MPG without a problem, but will not make a VCD. Rates that are frequently encountered on the Internet are 30 and 15 fps; 29.97 is a rare choice.

If the original MPG does not conform, it must be opened in a suitable editor and recompressed in exactly the right format. First, there are not many editors which will do that; the best editors begin with AVI, not with MPEG. However, ULead Media Studio 5.2 will do it. Unfortunately, compression is a lossy process. In general, you begin with a file nearly conforming to the standard and end with one which has double the losses. While it is fully conforming, it looks bad. That's not because the compression process is done badly but because double compression is bad.

Returning to the audio analogy, if you download an MP3 sampled at 32 Ksps and want to convert it to CD-DA, you must first decompress it. You may do that by opening it in a WAV editor which accepts that input, such as GoldWave, or by opening it in a separate program, such as WinAmp, and saving it as a WAV. Once decompressed, the file must be resampled, typically in a WAV editor. The resampled WAV is then compressed again to MP3. Decompression is lossless. Resampling introduces problems depending on how well it is done; quick resampling can produce audible artifacts while precise conversion takes a long time. Finally, the resampled file is compressed to MP3. Whether the three steps are done in three, two or only one program, there are three stages of loss: the original compression, resampling and the second compression.

If I were forced to make a nonconforming MPEG conform, I would use two programs - in part because I prefer Xing's compression to ULead's. I would open the file in Media Studio and save it as an AVI, assuming it was not larger than 2 GB in that format. Then I would compress it with Xing. (**Note:** Compression is *very* demanding of the computer. I usually let my 400 MHz Pentium II run overnight on a 10-minute clip. That's another reason for using Xing - it's not as slow as Media Studio. The third reason is that Xing offers VCD settings which are absolutely compliant, where you have to set your own parameters right in Media Studio.

The conclusion I suggest from all of this is that the payoff for forcing conformance is too great to be worth the costs: software, learning curve to use it, time to process and quality loss. If your initial MPEG is not in whitebook format, you probably do not want to convert it to a VCD.

Compression and Decompression

Confusion in a Bottle

In large measure, this page is a recap of primer material which has proved hard to decipher. In part, that is a natural result of poor naming and confusing concepts. Regardless, let's try again.

The purpose of compression is to use computing power to give the brain the impression of more signal than is really there. The effectiveness of compression is measured by the amount by which the rate is reduced and the loss of quality in the result. The codec (the algorithm which compresses and decompresses the signal) uses a model of human perception to deliver a given data rate from a specific input stream.

Two kinds of video compression have been standardized; another has been dropped and a fourth is on its way. MPEG 1 demands a moderately powerful computer or low-cost dedicated hardware and delivers a moderately compressed picture with acceptable sound. It is used in many home systems and fits well with CD-ROM capabilities. MPEG 2 requires enough more from the computer so that it is not practical without hardware playback support; however, it produces high-quality video, typically the best available in the home. MPEG 3 was to have been even better, but it was dropped when digital High-Definition TeleVision (HDTV) was defined. MPEG 4 is under development now and will probably emerge with digital HDTV in the next few years.

Three levels of compressed audio are defined to support compressed video. They are MPEG Level 1, Level 2 and Level 3 (abbreviated by their extensions, MP1, MP2 and MP3) - and must not be confused with MPEG 1 and so on. In particular, MP1 is used for low-fidelity sources, MP2 for moderate fidelity and MP3 for high fidelity. Although audio compression level may be chosen independently of video compression, MPEG 1 usually uses either MP1 or MP2 depending on the video data rate to be achieved. If the video is being compressed a lot, then audio quality will be lowered to keep the data rate down and the advantages of higher Levels are lost. In general, a video encoder will choose the 'best' amount of audio compression for the target data rate of video compression. In some cases, there are specific standards (such as those in the White Book for MPEG for VCD). In others, you may adjust the balance for yourself. MP3 is the level used for DVD and other MPEG 2 applications.

Finally, let me note that compression is much more demanding than decompression. Typically, an MPEG 1 video designed to play back on a Pentium 200 without hardware assist in real time, will encode many times slower even with an efficient software encoder. Encoding MPEG 2 requires substantial hardware to be practical at all. At this writing, MPEG 2 encoding hardware is more expensive than the computer with which it works.

Short File Names Long

One of the rules for using data in a computer is that fully qualified names must be unique. That is, there can be at most one file with the name D:\d1\d2\d3\filename.xyz.abc and so on. If there were more than one, when you asked for a specific file the system would not know which to give you. That's true whether it's a Short File Name (SFN - maximum of eight legal characters, a period, and maximum of three legal characters) or a Long File Name (LFN - and with too many variations in legality to provide here). "Fully qualified" means including the full path from the drive letter through all the intervening directories/files to the file name. For these purposes, note that a directory or folder looks just like a file when viewed from the folder or directory at the next higher level.

In various places on your system, you probably have several files named read.me - or readme.txt or whatever. Suppose you wanted to put two read.me files into one directory. The OS would, quite properly, refuse. It can't put the second one there with the first because the fully qualified names would be identical. That would also be true with longfilename.txt - the fully qualified LFNs would be identical. The major problem arises when one has different fully qualified LFNs but one needs to have SFNs which are visible in and accessible on a system using 8.3 names. MS decided to write the SFN logically, as 6~k.3 where k is a single character counting 1 to 9 to a to z (and perhaps beyond, I haven't checked). Now, there is an error in that process - which you will discover if you try to retrieve file longfi~c.txt, but that's not the point here. The point is that the fully qualified SFN must be unique in any system.

The SFN is not a property attached to the file. When a file is written to a directory (folder), an SFN is created. For an 8.3 name which is unique, the SFN matches the LFN. For a non-unique 8.3, you get the usual question: Do you want to overwrite? The same is true when you have unique and non-unique LFN's. The problem comes when you have a unique LFN but there is already a file in the directory/folder with the SFN that that LFN would generate.

In folder D1, you have longfilename4.txt with SFN longfi~7.txt - don't ask why, you just do. In folder D2, you have longfilename6.txt with SFN longfi~7.txt - perfectly feasible. Now you want to put those two files into otherwise empty folder D3. If the SFN's were preserved, the OS would have to refuse to put the second file into the folder - its fully qualified SFN would be the same as the first one's. However, you would have no way to repair the problem since you can't alter the SFN without renaming the LFN. So MS did the only logical thing: they did not preserve SFN's. When you write a file to a directory, if the LFN's are unique, a new SFN is generated in which the character after the tilde is invented. It's really the only way to do it.

If you have bothered to read this far, then play with the process to see how it works. One interesting variation is to write into an empty folder longfilename.txt - and then to try to write in longfi~1.txt which would have the same SFN (since the SFN of an 8.3 file is the 8.3 name). Also notice what happens when you write the LFN's in different sequences. It's all perfectly logical - but also very inconvenient for users who want to use SFN's in an LFN environment. I would blame Microsoft for doing it wrong if I could figure out how to do it better.

Firmware

Firmware is the program that's on a chip in most devices you use with your computer. It is definitely in your CD-R. Originally, such firmware was stored in Read-Only Memory (ROM) of a flavor called Programmable ROM (PROM). The information is hard-wired into the chip and can be changed only by changing the chip itself. In a later version, the chip could be burned (actually burned in the sense that metallic links could be blown out) by relatively high currents. That made it a lot cheaper to upgrade the information. Of course, by now we were writing to read-only memory, but that's computerese, not language. It got worse when people figured out how to avoid actually burning the metal and developed EPROM (Erasable PROM using ultraviolet light) and EEPROM (Electrically Erasable PROM). Still, it is a way of storing information into your hardware on how the device is to listen to and to respond to the commands it receives. It is, in a word, the sole storehouse in the device of information on what to do with inputs from its host.

The PROM in your CD-R holds firmware. If the manufacturer supports doing so, it may be written from your computer instead of needing to be done by a service center. To do it, the computer must tell the adapter how to supply the signals needed and the card must have the higher-current supplies ready to drive it. Some cards do and some don't; some manufacturers can and will let you do it, others don't. The key is that if you burn the wrong information into your device, it will lose the ability to listen to the computer to get the right information in. There is no way around it but to have it serviced and if you have burned really bad stuff, the manufacturer and service people may simply say: Tough luck - we told you not to do it and you ignored us. You voided your warranty and any implied contract, so it's your problem. Next time, follow the rules.

One brief note: there are two components to the upload you will try to use. One is the binary data which go to the device - your CD-R. The other is the loader which is the program that tells the adapter what to do and how to do it in order to get the information into the device. When the loader runs, you will see some strange stuff on your screen: a version number that has nothing to do with the firmware version, apparent error messages which are not errors, etc. None of that matters. When flashing is complete and you reboot, you will see the new firmware revision as the SCSI buss is polled. At that point, you are free to sigh with relief.

Nothing else that you do with a CD-R has the potential to do as much harm as burning firmware. If you use a program from an unsupported site, you are taking the risk. If you are very lucky, it will work. If you are reasonably lucky, it will do no harm. If you are typically lucky, you will still be able to burn the right firmware when it comes out. If your star is not in the ascendant, you may have made an expensive paperweight.

Software from an unauthorized site is the stuff of Alice in Wonderland. Eat it or drink it and something dramatic will happen. Sometimes, it isn't really bad at all.

Storage on a CD-R

Some of the following is adapted from a post to AdaptecCDR by Dick Langley, dick_langley@ndex.com. My thanks to him for permission to modify and to use his material. And since I modified it substantially, don't blame him for the mistakes - they're mine!

Let's start with an audio CD, just as the developers of CD-ROM did. CD-DA tracks are made up of blocks. There are 75 blocks per second of music. There are 588 samples per block; each has 16 bits for each of the two channels, so it takes 4 bytes for each sample or 2352 bytes per block. Now, the accuracy of reading from a CD is not nearly as high as from a hard drive or RAM. For audio, that's not much of a problem, but when the disc is used for data (as a CD-ROM), drop a bit or two and there goes the whole program. To reduce the rate of data errors and to prove that we're computer types and think in powers of two, they decided to use the same block to hold 2K (2048) bytes of data and to take the remaining 304 bytes for error correction. So a block on a CD-R is 2352 bytes long and can hold either 1/75th of a second of redbook (CD-DA) audio or 2K of data.

Audio

In documentation of audio and video processing, the word 'sector' is used for a chunk of information. By default, that chunk occupies 1/75th of a second and fits into one block. Because sector comes up later, I'll stick to blocks, thank you - and you can translate when you read about audio editors and such. And you are also now prepared to do your own arithmetic to see why 74 minutes of audio fills the same blank that holds 650 MB of data, or to understand why, when you extract a CD-DA to your hard drive, it won't fit into 650 MB. And all that neglects the cost of the header on a WAV file. On a CD-DA, there is no header. The information is in a continuous stream of bytes from one block to the next. There's a TOC to tell the player at which block each track begins, but there's no audio file on a CD-DA. (The .CDA you see with Explorer is a fiction.) To create a WAV file from the data stream, a drive capable of Digital Audio Extraction (DAE) supplies the byte stream to your software. That software converts the stream into WAV format, complete with header. DAE is done by dedicated programs (see theFiles and URL pages at this site) or by your CD-R authoring program. Converting from WAV for CD-R is done by the authoring program.

If the information in a wave file is not an exact multiple of 588 samples, software pads to a complete block with nulls - silence. If you're recording a concert (for instance) and you want your selections to be addressable as different tracks via the TOC, those tracks should be multiples of a whole block. As you slam two tracks together, the sound (crowd noise for instance) should be constant when transitioning from one track to the next. If you don't have a true multiple of sectors, the last sector of a track will be padded with nulls or no sound. (In some older editors, the padding is not done with zeroes and whatever bytes were lying around will be read onto your disc). As you play your new CD and you transition from one track to the next, you may hear a click (where the sound you hear goes to nothing or to random bytes, then back again). It becomes annoying to hear these clicks between tracks. To solve this situation, software offers quantization. When you're cuttin', pastin', and slicin' tracks apart or together, it will do it in multiples of 588 samples to give you whole blocks. So if you have that concert that you've broken up in tracks, it sounds like one continuous track when played. Mike Looijmans' CDWAV (on theFiles page) was designed to cut on block boundaries. As you look for software that manipulates wave files, you should look for this "feature". Of course, that means your split may be moved. If you can hear the difference when it's shifted no more than 1/150th of a second, try analogue recording.

Data

Now for data blocks. Again, there is a problem stemming from the fact that there's no File Allocation Table on a CD-ROM. There is a TOC and most of what a FAT needs is contained in the TOC. So OS extensions are used to translate the TOC to a FAT - on a PC, to a FAT which looks like the one you would have on a 650 MB hard drive. Information on a FAT16 hard drive is stored in up to 64K sectors; the size of each sector is the total size of the drive divided by 64K, then bumped up as necessary to the nearest power of two. So a 650-MB hard drive would have sectors of 16K bytes each. Since each file must begin on a sector boundary, it will waste whatever space is left unused in the last 16K sector.

Ah, but things are better on a CD-ROM with its 2K blocks, right? No - that would be too simple. Since the OS extension must address the CD-ROM as though it had a FAT, it must look for the file to begin on a 16K-sector boundary. A 4K file wastes 12K on a 650-MB hard drive or a CD-ROM. On average, each CD-ROM file wastes half a sector or 8K. When you add up the file sizes for a disc you are about to burn, you must remember to allow for that wasted space. If you are writing from a drive which has 16K sectors - such as a Jaz or a FAT16 hard drive with at least 512 MB but less than 1 GB - you can use Explorer and Properties to determine exactly how much of your blank will be needed. If your source drive has a different sector size, you will have to do your own arithmetic or approximate using 8K wasted per file - and a little more just in case. *There's more on this subject in the page on Packet writing.*

Packet Writing

As usual - but perhaps more so given my ignorance - this material is unofficial. In the hope that it's helpful, buckle up for a hard ride. Also as usual, it is aimed at Windows users; Mac users should be aware that much of it needs to be adapted to their/your needs.

ISO 9660, Level 1/Level 3

The International Standards Organization controls standards generated by industry. The standard for CD-R is numbered 9660 and covers the agreed rules for recording all sorts of CD's. Not all manufacturers of hardware and software look on those standards the same way: some violate them freely, some implement them in part, some try to adhere to them strictly. When a publisher like Adaptec tries to implement them fully and a hardware vendor does not (names omitted to avoid assigning guilt), problems arise. The user complains that Adaptec does not do something the drive manufacturer claims to do and everyone is likely to be both right and wrong. Adaptec says: the h/w does not implement the standard; the h/w maker says: it does what it needs to do; the user says: it's not my fault. There are many solutions available. Another software package may be used which is less stringent about the rules; a different device may be used which follows them more closely; the user may give up on the capability.

ISO 9660 Level 1 covers the formats we have known for a while. They are CD-DA, CD-ROM Mode 1, CD-ROM Mode 2, Mixed and CD-Xtra. In the first three, the disc may be written Disc At Once (DAO) if the hardware and software support the capability. CD-DA is covered in other pages of this primer under Audio. CD-ROM Mode 1 was designed for single-session recording; Mode 2 is slightly less efficient and was designed for multisession. In fact, either may be used for either single- or multi-session - the difference is that while only very old drives balk at multisession Mode 2 discs, some moderately old ones won't take multisession Mode 1. Both Mixed and CD-Xtra modes combine CD-DA and CD-ROM on a single disc. Simply put, CD-Xtra permits playing the audio on a conventional CD machine since the audio tracks come first.

On a conventional CD of any flavor, the first information the reader sees is the Table Of Contents (TOC). In CD-DA, the .CDA 'file' specifying each track is a fiction that a file manager (like Explorer) uses. In a single-session, closed disc, the TOC contains information which is interpreted by a DOS extension as being a directory. The rules for converting TOC information (including the fact that everything on the disc is read-only) are embedded in the extension, such as DOS's MSCDEX. On a multisession disc, the TOC for the first session tells the system where to find the next one - or whether the current one is the last, or whether the disc is now closed. DAO is necessarily single-session (which is why one cannot write DAO tracks in Mixed or CD-Xtra formats).

In order to support UDF, the TOC - first information on the disc - had to change. As a result, MSCDEX and other extensions cannot read packet-written discs even after they are closed to ISO 9660 format - the information that the extension expects to see is simply not visible. The solution was to invent Level 3. Functionally, it looks the same as Level 1 to the user, but it requires a different extension. That extension is available in Windows 95, Windows NT, and Apple System 8, so that they can read Level 3 discs. It is not in DOS, Windows 3.x or other systems. This has nothing to do with your hardware or application software. Someone may write a new extension, but Microsoft may not be interested and there's not much money to be made from the effort to inspire a company to do the job. If you write one, you will be much loved by many users, but probably not well paid for it.

Universal Data Format (UDF)

Now comes packet writing - which has two distinct flavors. The idea behind writing packets is that if one does not worry about the TOC, one can write data to a CD-R in blocks or packets. The information that goes into each packet needs to be accessible by the writer, but not necessarily by an ordinary reader. By writing an appropriate set of

functions into the heart of the OS - approximately at the level of the extensions to DOS - a vendor can provide packet capability for writers with suitable hardware and firmware. However, there is no TOC in the Level 1 or Level 3 sense; that information is on the disc in other ways. That takes extra space on the disc and means that those data must be interpreted when a UDF disc is inserted into the drive.

At one time, Sony had its own format for packet writing and provided a driver to permit a UDF disc to be used in a reader; that appears to have been dropped in favor of the standard discussed below. CeQuadrat also has a form of packet writing in PacketCD; its capabilities have not been reported and I cannot test it because it runs only in Win95. Adaptec's DirectCD implements the UDF standard and works in Win95, Win98 (so far) and NT 4. For the following, I will speak only of DCD and its implementation of UDF. If someone can provide information on other realizations, I'll be glad to append it.

In the first version(s) of UDF, only conventional, non-erasable media were considered. For them, a very efficient system could be devised. At the head of each packet is the information on the file it represents including its length (which, of course, implies where the next packet begins). Each packet contains an integral number of blocks; the only space wasted is for that header and for the leftover bytes of the last 2K block. A variable-length packet can be converted to ISO 9660 Level 3 format because all the bytes of the file are in complete, consecutive blocks. When the session is closed to Level 3, the TOC is written in a more or less conventional fashion, but data interpretation requires that the OS ignore the old header information. In other words, the TOC is similar to that of Level 1, but both the TOC and the data are different enough to require changes that make the old extension useless. (Note that the Level 3 TOC may be written either to close a session or to close the disc. If the session is closed, another UDF session can be started and finalized; when a disc is closed, the TOC indicates it and will not permit reopening.)

A note on capacity: Variable-length packet writing is very efficient. Because of the small amount of space wasted, it is possible to write more blocks than can be accommodated by a FAT16 for a 650-MB disc. If you fill a variable-length packet disc with many files of the right size, it may not be possible to close it to Level 3. That means you won't be able to read it except in your writer - the session is still open and a CD-ROM cannot read an open session.

When UDF is applied on an erasable blank, a new option is available. In addition to being able to write variable-length packets and to erase the whole disc, you can write, read and erase individual fixed-length packets. Now the CD-RW begins to resemble a conventional floppy, where the packets are the sectors, at least from the point of view of the user. That capability is reflected in DCD 2.x. However, one problem of erasable blanks is that they support a limited number of erasures - nominally, a thousand. Another problem is that when one rewrites on any drive, data may be written over previous entries. Overwriting results in 'scrubbing' or disproportionate reuse of a single region of the disc; it also leads to fragmentation of files so that their constituent blocks are no longer necessarily contiguous. For fixed-length packets, the consequences include high overhead to keep track of all the blocks and inability to convert the disc to any Level of ISO 9660. So an erasable blank formatted for UDF offers only 494 MB of storage, slow access (about the equivalent of 1x) and inability to make the disc readable in a conventional player. Neither Microsoft nor Apple will be reading fixed-length packets in their promised OS's, so those discs are and will remain for a while useful only in the writer or in a MultiRead reader with a suitable driver (such as the one now available from Adaptec's WWW site).

If you think about it for a moment, that sounds much like the case for a very large floppy - which is a good analogy to keep in mind. Both a floppy and fixed-length packets waste about 30% of raw capacity on formatting; neither can be read in a device which cannot write it; each is relatively slow; and if you have a catastrophic failure during writing, you may lose the whole disc. Of course, the UDF disc is about 350 times bigger than a floppy, but that's a (rather important) detail.

Adaptec has not chosen to offer the user the option to format an erasable for variable-length packets. Rightly or wrongly, they find little sense in using them on CD-RW, but therefore face a real potential for user confusion. As long as they are the only contenders in the general multi-OS marketplace, we can only ask them to re-evaluate. If they get a serious competitor, the market may make them change their minds.

How UDF is implemented - and when it's safe

When a CD-R is formatted for variable-length packets, space is committed for the TOC, but no TOC is written. Nothing else of importance is done, so the process is quick. When a file is written to the disc, it becomes a packet which is headed by the information corresponding to what would be in its FAT entry, including its length so that the location of the next packet can be determined. At the same time, DCD holds in memory a virtual FAT (VFAT) which gives access to that file just as though it were on a conventional disc. Write another file and the VFAT is updated appropriately. When the disc is ejected, the VFAT is removed from memory and nothing is written to the disc.

If the session is closed when the disc is ejected, the information for the TOC is written to the disc from the VFAT; the packets themselves are not altered. The TOC is then accessed by Win 95 or NT 4 through the usual extension. Until the disc is closed, there is no usable TOC information on it - which is why it cannot be read conventionally. A disc with a closed session has a TOC in Level 3 format; that TOC may point to the place another TOC will be written or may say: this disc is closed. During the time that a packet is being prepared and written, a catastrophic system failure would cause the write to be incomplete and might make the disc useless; at all other times, even if the system is powered down with the disc in the drive, the information is safe. The laser is off, everything needed is already written and nothing can happen. Apply power, return to your OS, and DCD reads the packets and forms the VFAT.

With fixed-length packets, the process differs in some significant ways. The key fact is that when the disc is ejected the VFAT is written to the disc so that the location of the packets forming each file can be traced. Logically, one would ask: Why not write the VFAT each time the disc is written. The answer is, simply, that that would increase the number of writes to the space that it occupies and would kill the disc more quickly. The VFAT in memory becomes a FAT on the disc. Since the runout track was written, a suitable driver then gives access to the files. That driver is provided by the UDF Reader from Adaptec. Since a fixed-length-packet disc is always RW, that driver is only useful in a MultiRead drive or a writer capable of reading erasable media. A fixed-length-packet disc still in the drive after having been written must be ejected (and have the VFAT written) to give access to the new files. If power fails or the system is reset before ejecting the disc, the new information will be lost and the disc may be unreadable. With luck, a utility to recover what can be recovered will become available, but at this writing the disc is vulnerable to such a failure.

Why Packets?

Packet writing is new, flaky and ill-supported. Why in the world would anyone need it?

In fact, practically no one does need it. But it may make your life easier if you use it well. It will certainly make it harder if you use it badly.

Variable-Length Packets

With variable-length packets (yes, there should be a better term), you can write arbitrarily large files to a CD-R as they become ready to write. Each day, you download from a mainframe all of your correspondence, transaction history, presentation drafts and other information to get you started fresh the next day. That information updates your HD - but drop a copy onto a CD-R for permanent record and to be sure you can start at full speed tomorrow even if the network goes down. A few megabytes - or even tens of megabytes - need to get easy, non-volatile storage in chunks. A perfect application for variable-length packets. Or you have fifty pages to scan and OCR tonight and a batch-job program which will do it and will save both images and text as it goes. Let those pages be output to a UDF CD-R; when everything looks good the next day, finalize the disc to ISO 9660 Level 3 and file the disc. I update one WWW site each week with one to five MB of files. Each week, I archive the site to UDF so I can return to my mistakes later and apologize appropriately.

Variable-length UDF is rarely an alternative to conventional CD-R. Basically, it's the tool for what we used to do with floppies and now might do with Zip discs - if they did not cost a dime per megabyte. The price of the CD-R is nearly a hundred times less, so we can feel happier about saving history with it. But when you have truly critical data or very large blocks to deal with at once, packet writing is not your best choice for primary storage. In the example of HD update, primary storage is on the HD and the UDF disc is backup. In the case of OCR, the final output on your HD is the primary product; the UDF is a way to verify what was done and to repeat the OCR even if the originals are not to be scanned again. Once closed to Level 3, the UDF disc is as reliable as any CD-R and almost as versatile, so 'wasting' the 15 MB or so to finalize should be routine.

Fixed-Length Packets

Fixed-length packets are written only to erasable blanks and provide a very different sort of environment and very different potential. Think of a fixed-length UDF disc as a slow drive like a very large floppy - and try to think of its use as we used floppies before HD's became common. (That was not so long ago for Apple users.) You can install a program to such a disc and run it from there even as it updates its files. If the program is designed to be used that way, it will be CPU-intensive, not disc-intensive, and will manage DLL's and overlays differently from the way it would on a fast HD. There are few programs today written for that sort of thing, but three possibilities should be considered for CD-R use: cleaning and compressing audio and compressing video. Those are programs of modest size which write relatively little information after a lot of processing. On most computers, MP3 compression of a redbook WAV takes about as long as it does to play the WAV. The file can be read from a conventional CD-ROM (at about 150 kilobytes/sec) and execution and output can use fixed-length packets, writing 15 KB/sec. That format is written at 2x, so the delay for slow writing does not matter much.

Using fixed-length packets where variable-length makes sense (e.g., for backups) is quite possible, but is usually a false economy. You can work that out for yourself, considering initial cost, loss of space to formatting, slow write and - most important for many - inability to finalize. Fixed- and variable-length packets are good solutions for some situations, but can be frustrating or costly if used incorrectly.

Packet Writing - Intermediate

Despite appearances, the following is a simplified account of storage on a DirectCD disc. For the full story, check ECMA 119 and 167 at <http://www.ecma.ch> - or the equivalent from ISO if you can afford their fees. I am not the source of this information, but its author is fully knowledgeable and has done an admirable job of making it understandable. If you doubt that, try the ECMA documents.

In order for data to be readable on a standard CD-ROM, the disc must contain three things: a lead-in (containing the table of contents or TOC), the data area (containing the tracks), and a lead-out (containing nothing). The TOC is NOT the same as it would be used for a file system; it only describes what is in the data area - like track start and stop positions. An "open" disc or session contains only the data area since the ultimate track layout is not known (new tracks can be added) until the session is closed. These data are stored in a temporary location called the PMA. When the session is closed, the drive gets the temporary information from the PMA and writes it in the lead-in as the TOC. Again, this has nothing to do with the actual contents of the data area. Move that disc to a CD-ROM drive, and the CD-ROM will find the TOC and tracks and is able to perform reads or to play the associated audio. In a multi-session disc, you would have multiple lead-in, data, lead-out sequences with each lead-in except the last linking via some pointers to the next. The CD-ROM drive starts at the first lead-in and follows it through to the last one to locate the current data set. A CD-R/RW drive does all this, but is able to read the PMA as well so it can find the open session data. Now, when using incremental, or packet writing, the link, run-in and run-out blocks are added whenever a packet is written; they are required due to the way that data are encoded onto the disc. It also has nothing to do with the lead-in, data, lead-out business either. Do these link blocks cause problems for CD-ROM's? Some, but MultiRead is supposed to mandate the ability to deal with them. So how does this all apply? When using CD-R, DirectCD will write into the data area of the open session, appending packets as needed. In this state there is no lead-in or lead-out so a CD-ROM cannot read the disc. When the disc is converted to ISO-9660 format, we write the ISO image to the disc and close the session. We could simply close the session and still be readable on a CD-ROM because all a CD-ROM cares about is the lead-in, data, and lead-out. Nobody could get anything useful from it without a reader, but the drive would be happy. ISO 9660 is useful because of the wide availability of ISO readers. In the case of CD-RW, we actually format for all the data and close the session right in the beginning. That is why it is readable (in theory anyway) from the start. With both CD-R and CD-RW, the file system is written into the data area of the disc, but with RW the session containing the data is closed and therefore readable on a MultiRead CD-ROM.

With DirectCD, data are written in discrete units (packets) that may be as large as the recording drive's buffer. Because of the way data are encoded onto a CD, each packet is preceded by a series of blocks (link and run-in) and followed by a couple of blocks (run-out). One way of visualizing this is to think of the packet being encased in some "insulation". The net result is that there are some additional blocks between consecutive packets of data. For example, if we have to write a 3-megabyte file on a drive with a 1-megabyte buffer, the data will likely end up written in three discrete pieces or packets, each packet being one megabyte in size and each pair being separated by the linking blocks. In order to read the data correctly, the reading software must be able to skip over the linking blocks that separate each packet. MSCDEX cannot do this. It assumes that all files will be written in one continuous extent. When it encounters files written in multiple pieces it tries to report them as multiple files. The Win95 and WinNT readers are able to skip ahead as needed. This is really the distinction between ISO level 1 and level 3. In level 3, the files can be in different pieces (multi-extent) while in level 1 the files must be single extent.

Another wrinkle added by a file system implementation is that we are able to handle multiple threads of execution writing at the same time. This means that we may have data interleaved on the disc as a result of receiving interleaved data from the operating system. This would also require multiple extents. Finally, the fixed packet size used for RW is 64K or 32 blocks (blocksize = 2K). Including the link blocks and the total amount written becomes 39 blocks (1 link, 4 run-in, 32 data, 2 run-out) or 78K. This is also why there is so much overhead for direct overwrite on RW media, approximately 18% waste!

All Things SCSI

Neal Gibbons, neal_gibbons@europe.adaptec.com posted this material to the CD-R list. With his permission, I am sharing it here.

I felt the need to post this after reading some of the threads on this list recently and also due to a number of direct emails I have received. What follows is not gospel but it sure helps if you believe in it!!

- 1) AHA2940UW. You must not use all three connectors at the same time. If you do so you will violate the SCSI spec. by turning your SCSI bus into a 'T'. Only use two at any one time.
- 2) Termination. Only terminate the ends of the SCSI bus. Not the middle or only one end but both ends of the bus. Do this either by enabling terminators on the device or by terminating the cable/external connector.
- 3) SCSI ID's. Try to leave 0 and 1 for disk drives. It is not strictly necessary to do this but traditionally 0 and 1 have been reserved for disk drives and it won't do you any harm if you continue this tradition.
- 4) Wide/Narrow SCSI. You can mix wide and narrow devices on the same bus but you may need to use special converters/terminators like the ones from WWW.TMCSCSI.COM. Try to convert from Wide to Narrow, that way is easier to implement and the Wide devices can run, well.... Wide. Keep the SCSI ID of the controller at ID 7, moving it above ID 7 will result in the Narrow devices being unable to 'see' the controller. Adaptec and other controllers negotiate for Wide SCSI transfers on a per device basis, the Wide part only being used during actual data transfer.
- 5) Disconnect/Reconnect. This feature is normally enabled, it allows multiple devices to 'overlap' transfers on the SCSI bus and therefore maximise the available SCSI bus bandwidth, it's what makes SCSI suitable for multitasking environments. However, it can also cause a good run in coaster manufacture, try disabling it as part of your normal 'hunt for the cause of coasters' trouble shooting. This setting can also be set on a per device basis as well depending on you SCSI controller.
- 6) Synchronous Transfers. If this were a perfect world then every device ever produced would handle the request for Synchronous Transfers correctly, however it isn't. If you are trouble shooting again, then try disabling this option within the SCSI cards BIOS. CD-ROM's, CD-R's, Tapes and Scanners have all been known to handle the controllers request for Sync. Transfers in a poor way. Today's hard drives are immune to this but....well just beware. Again this option is enabled on a per device basis so this allows devices to run at different speeds on the SCSI bus.
- 7) Ultra SCSI. Another option that can (only just now with the 2940UW) be enabled on a per device basis. Apart from increasing the speed of the SCSI BUS it has also increased the number of head aches experienced around the world by users and Tech. Support people alike! Theory goes that you can mix Ultra and non-Ultra devices on the same bus, many people do and it all runs OK for the lucky ones. Because of the cable limitations imposed by just one device running at ultra speeds my suggestion if you do have issues is to a) first make sure your cable is A1 OK, b) make it as short as possible or c) separate the Ultra devices from the non-Ultra by installing another card.
- 8) Cables. Not all cables are created equal. Ultra/Wide and Fast/Wide SCSI cables are different. They have different specifications. Make sure yours comply with whatever you are running. External cables and connectors can cause you issues if they are not 'up to spec'. you only get what you pay for! It is asking to much when you spend \$100's on the latest all singing/dancing PCI controller and expect it to drive a device via a cheap cable.
- 9) Buss Mastering. Buss mastering PCI cards greatly 'ease the worry/pain' of CD-R work. Sure they are not necessary and many people do use ISA cards and non-buss-mastering controllers, like the 2920 but at the end of the day they are affordable, like the 2910 / 2930 and give you that 'good piece of mind' feeling.
- 10) Fixes. If it 'aint broke then don't fix it. If anybody has anything to add to this, like more info. and constructive suggestions then let me know, and I will re. post it.

The ASPI Layer

In the early days of CD-ROM, every manufacturer had his own interface and connections to the host. A standard interface (ASPI) was devised and manufacturers of SCSI devices either conformed or wandered out of the market. The situation with IDE devices was similar, but the standard that was developed for them (ATAPI) made them look to the system like SCSI and allowed them to work through ASPI. For that reason, your computer's IDE adapter and devices show up under the SCSI controls.

The ASPI Layer is a collection of programs to realize the ASPI interface. Those DLLs work together and must be consistent and complete to avoid problems in communicating and interpreting commands. Some vendors package a subset of the DLLs with their products - scanners are notorious for this. Often, they will grab an old file (one that's good enough for their needs) and package that with their own software - but with the then-current date. It will then install even though a later version is on your system and it will not be removed readily, again because the date is misleading. (Okay, it's a lie!)

ASPICLK from <http://www.adaptec.com> will analyze your system and report the consistency and version numbers of your ASPI components. If it reports a problem, you may try Adaptec's repair program - but that runs into the date problem and may not resolve inconsistency. At this writing, the latest ASPI layer (1014) appears to be solid; it is installed with Easy CD Creator 3.5b and may be reason enough to update your software if you have an earlier version. Unfortunately, the solution using Steve Gibson's ASPI_ME is no longer available.

The variety of CD-R failures due to a faulty ASPI layer is remarkable and defies detailing. Any repeating error message dealing with communications to or from your writer should trigger a verification with ASPICLK. It sometimes appears that programs that really do not need the ASPI routines install them (badly) just to be sure. With ASPICLK and the Adaptec files, ensuring a working ASPI layer is painless. To steal from an American commercial: Just do it!

Oversize Discs

The capacity of a 'normal' CD-R, like that of a normal CD or CD-ROM, is 650 MB of data or 74 minutes of audio. Sometimes, one runs across a source disc that seems to be larger or a CD-R that claims to have more recording capacity. Those phenomena may be real or illusory and they can be handled in various ways. This note is intended to provide guidance in copying the uncopy-able and similar issues. One suggestion I do make is: Don't break the law. Backing up a disc you own is apparently legal. Saving your own files onto CD-R is certainly OK. Otherwise, please be careful; I don't want to be an accessory to theft.

Estimating size

First, capacity of a disc is measured in megabytes - but how large is a megabyte? Is it a million bytes? a thousand kilobytes? or 2^{20} bytes? A thousand kilobytes is 1,024,000 bytes or about 2.5% more than a million bytes; a true megabyte (1,048,576 bytes) is about 5% more than a million bytes. Since there is no enforceable standard for the word 'megabyte', manufacturers of CD-R's get away with claiming almost 5% more capacity than is 'really' there. It's an accounting trick that will not let you store one extra character on the disc.

There are CD-R's with more than 74 minutes' recording capacity. The maximum possible is a shade under 80 minutes. There are also 63-minute discs, but while they are said to be more reliable than 74's, they are again hard to find and no less expensive than the longer ones. Even the 80-mm blanks with about 15 minutes' capacity may reappear. (There are also recordable minidisks, but this is complicated enough already!) The result is that under all reasonable circumstances, you will be recording on a 74-minute, 650-megabyte CD-R. If you need an 80-minute blank, you will need to hunt it down.

Where did the megabytes go?

A data CD-R gives the impression that it is written with a FAT and structure just like that of a hard drive, but it isn't. As a result, there is information stored on the CD-R that takes away capacity in hidden ways. The amount 'wasted' depends on the type of recording being made. More to the point, CD-R software does not usually estimate the capacity of the disc precisely. A similar phenomenon occurs with audio CD-R's. As a result, you may be 'cheated' out of some tens of megabytes. Even worse, you may not know about it until you have burned the disc - and find that it cannot be closed.

The size of the Table of Contents (TOC) which consumes CD-R capacity depends on parameters you set when recording. To minimize the wasted space, record DAO. If that doesn't solve the problem, try writing an image to your HD. That image is a byte-level preview of what will be written to the CD-R, so the software need not estimate whether or not it will fit but can actually count bytes and sectors to prove it. Another test (not a perfect one) is to do a test write. Most cases of trying to write more than the disc will hold will be caught in the process and all you will have wasted is the time for the test. You may even run a test on an ISO - just to be certain. As usual, different writing software has different ways to do these things, so check your manual.

When you are preparing to write a CD-R, your software may estimate that you do not have enough room. Some programs simply refuse to proceed. Others inform you and blithely try to do the impossible when you tell them to. If you are near capacity, write the image, then test writing it to the disc you want to use. If that works, cross your fingers over a rabbit's foot and burn your copy. And remember what happened to the rabbit.

Disc length and Overburning

Although standard discs are marked 650 MB and 74 minutes, in fact they vary significantly in actual length. The true size - in blocks or another measure - can be found with appropriate software such as the Disc Information of Easy CD

Creator. I provide a table with some values reported by others and a link to another such table. 80-minute discs have returned to the market and may serve your needs at modestly higher cost.

Overburning refers to stealing some of the space used for a runout or leadout track to hold data. That track is standardized at about 13 MB, but not all readers need the full length. With some hardware and software, you may use that space for data or audio - but you should be aware that that may leave the disc unreadable in some drives. This procedure is risky because it leads to apparent success and works often, then can let you down when the disc is most critical to you. Even with overburning, you cannot exceed the address space of an 80-minute blank, so I suggest that you look into other choices before overburning even if your system has the capability. Still, if you are in a critical situation and *must* find an extra couple of megabytes, it may be worth trying.

Oversize source

There are two distinct cases here: a source CD or CD-ROM which is too big to copy, and a set of files under your control which exceeds the capacity of the CD-R. In fact, there's a middle ground of considerable interest.

First, some CD-ROMs are effectively copy protected and cannot be copied at all. (More precisely, some can be copied but the copy won't run.) The mechanisms include physically altering the medium as well as creating software conditions that some or all CD-R software will not handle. If you wish to backup such a CD-ROM and if you have reason to believe that the protection is effected only in the data (not by the physical medium), try other writing software to see if it can do what your preferred program won't. That may cost you some coasters, but you must expect to pay a price for circumventing the vendor's efforts. (Incidentally, there may be information in a newsgroup or mailing list that will be of assistance.)

Next, there are tricks that may be played by the CD-ROM publisher with files and directories that make the amount stored appear to be larger than it is - sometimes by a factor of two or more. If you make an image of the source disc and find that it fits onto your CD-R, you're home free. Incidentally, those tricks may be used for purposes other than to inhibit copying, so paranoia may not be appropriate.

In the category of files under your control, the solution is obvious: Control the files. Save some for another disc. If you are archiving, consider zipping text, database and other low-density types. Resize images or use a bit more compression. For audio, drop a verse or a chorus or repitch the selection. 10 MB is about 1.5% of the capacity of a CD-R. If everything you did were 98.5% efficient, you wouldn't be likely to complain. So, don't do it here, either.

Finally, if you have a disc that is stubbornly just a bit too big to copy, don't waste your time trying to figure out how the publisher did it. Retrieve all the files to your HD and remove the read-only attribute on the data. Now you have control and can apply the same tricks you would if you had created the files in the beginning. Perhaps there's a directory full of information for a platform or OS you don't use; kill it. There may be 500 graphics files you don't care about; scan for any worth having and dump the rest. Use your imagination and your good judgement. (One useful trick if you have a Jaz drive or a blank HD partition: create your reduced disc on that drive and run it. If it works, burn it; if not, make the appropriate adjustments.)

A Note on Microsoft's Way

It should not be a surprise to you that Microsoft knows their operating systems - and how to break the rules that they impose.

Much of MS's software is cross-platform: it runs on different systems with only moderate changes depending on the host and its configuration. An easy way to develop that code is to have the common elements in one set of folders developed by one group and have the parts particular to a specific platform developed by a group of specialists in that platform. For ease of control, the platform developers are probably given read-only access to the common code in

folders that appear to belong to them. In effect, the developer for a DEC system builds her own CD-ROM using the common code and her own interface to the Alpha. She can then create a CD-R for the Alpha and test it. However, before release the Alpha and Intel (I386) packages need to be combined onto a single CD-ROM for distribution.

It is easiest to keep all the references to common code in the platform-dependent parts unchanged. To accomplish that, MS can and apparently does fake the references in the TOC. That is, the common code appears to be in both the Alpha and the I386 folders. It is actually present only once, but the table which references it gives the appearance of providing two copies. In many cases, that results in a disc which seems to have much more than 650 MB on it. Note that this illusion is developed as a programming convenience, not to provide 'copy protection'. You can probably back up a MS disc which appears to be oversize either by dropping unneeded platforms from your copy or by making an image and burning from there.

Spindown

The one essential requirement for a successful CD-R is that the laser receives data continuously. The buffer of the writer itself is supplemented with that of the host computer to provide that continuous flow. If the data are not there when the laser is ready to write, the burn is interrupted and a 'coaster' - a useless disc - is the almost inevitable result. One reason for the data not being ready when the writer needs them is that the source has spun down - slowed or stopped its rotation. In general, hard drives do not do that, though they may if a 'sleep' mode is provided (especially on a laptop). However, other sources which use a rotating disc may spin down to low speed or even to a full stop. That is done to protect the source itself from wear or burnout on its bearings. Typically, a high-speed CD-ROM reader and a removable drive will spin down if inactive for a long enough period. Some drives (IOmega Jaz, Plextor readers) provide control of the delay before spindown; others do not.

If the source is being read continuously or nearly so, spindown is unlikely to be a problem. However, suppose that you are writing a mixed-mode disc from data on the hard drive and audio tracks from a CD-DA. While the data are being written, the CD-ROM is idle and may spin down. At the end of the data session, the buffer is emptied, the TOC is written and the reader is interrogated for data. It must be up to speed before the audio tracks can be copied; if that takes long enough, the laser will be starved and a coaster may result.

There are several ways to avoid the problem. One is to read all inputs from the hard drive or from the CD-ROM. Another is to set the spindown time long enough so that the source is still at full speed when interrogated. The easiest solution may be to keep an eye on the susceptible source; if it spins down while the other source is being read, spin it up manually. That may be done by accessing it with a program such as Explorer or by ejecting and reinserting the medium.

Note that there's nothing that the writing software can do to eliminate this problem. If the source isn't there (i.e., it has spun down) when the laser needs the information, the software and the reader are helpless.

Labelling

Now that you have produced your shiny silvery or golden disc, how do you mark it? By this time, you are probably not surprised to learn that that is not a simple question. Before going into the specifics, some words of warning are needed. DO NOT use ball point, Post-It notes, or any conventional glue or tape.

Construction

The bottom of the disc seems intuitively to be the sensitive part, but 'tain't so. The reflective layer just below the upper surface is far more delicate than the thick plastic on the bottom. Your reader will make quick work of a scratch or dust speck on the bottom, but anything which disturbs the metal foil will make the disc useless.

In the least expensive discs, the upper layer is little more than a plastic varnish over the foil. Solvents used to suspend ink and hard tips used to lay the ink down can wreak havoc. Some discs have a more durable coating applied to the top - various names are used to suggest that they are less vulnerable to damage. They are still fragile, but not to the extent of the bare discs. In general, a printed surface suggests a stronger one and one designed for you to write on the disc (underlined spaces for date, etc.) is a pretty sure bet. Still, a soft nib and water-soluble ink are desirable. Finally, there are discs made for ink-jet printers. They have thick, usually painted surfaces and can be written much more freely.

Pens

By now, this is easy, right? You want a soft-tipped pen with water-soluble ink for complete safety. The TDK version is available from several suppliers of bulk discs for about \$3. An art supply store can probably provide you with a range of alternatives in different colors and line widths.

It's inevitable that people ask about the excellent Sanford Sharpie line. Simply put, there's been no report of failure from the use of Sharpies, but the ink is not water-soluble so there's a chance that it's a problem waiting to happen. I use the standard Sharpie all the time on coated discs, but even though the maker of my inexpensive discs assures me that the Sharpie will work, I haven't chosen to try it there. I figure that the TDK costs me a tenth of a cent or so per disc and at that rate I can afford to splurge. <G>

I will offer one warning on the Sharpie and others: be aware of the tip and its potential for problems. Sanford's Ultra Fine Point is surrounded by a metal collar. If I want a fine line, I use the TDK with its semi-rigid tip. Perhaps I'm being overcautious, but I see no point in risking data loss for something as simple as this.

Labels

There are several manufacturers of labels and of positioning devices for CD-R. Avery's 5824 are unique in two respects: they have an ingenious positioning scheme which means you have high reliability without an additional hardware; and you can get them from any competent stationer. They are expensive so I recommend them for those who only want to use labels on rare occasion. They are also a good starting point for beginners since Avery provides templates for many programs and supports the 5824 with their Label Pro software.

There are also branded labels from Neato, Stomper and others plus unbranded ones from independent sources. There are two essential issues in selecting such labels - but only one can be determined easily. That is whether the printing surface is suitable for your printer and purposes. If you want to print photographic quality with an inkjet, you may have real problems finding suitable stock. The other problem is the adhesive. You would think you can use some sort of super glue - but the adhesive, like ink, must not contain any solvent which would attack the plastic or seep through to damage the reflective layer.

The adhesive bonds the paper label to the disc's upper surface - the one that's so vulnerable to damage. If it does not hold tightly enough, the label can begin to peel. You cannot remove it (that would pull off some of the lacquer layer and probably destroy the reflector and the disc); you cannot press or glue it back down again; and you cannot use the disc with the label pulling off. The best choice is to try to stick it down long enough to make a copy.

If the label is not very well centered and reasonably flat, it will unbalance the disc - particularly in a high-speed reader. You get one and only one chance to position the label and if your eye or your hand is off, you may turn a perfectly good recorded disc into a coaster. For that reason, the major makers of labels sell positioning devices. Once you have graduated from Avery and have selected the brand of label stock you want to use, the positioner is a wise choice.

Direct printing

There are several vendors with printers adapted from low-cost inkjet models. I have not used any of them, but the reports are consistent: they work well for a while, then critical parts wear out from the heavy load of a CD-R where the design is intended for paper.

There are at least three quality direct printers. Fargo and Rimage have been in the business for some time and Marcam has recently appeared as a contender. However, these devices are costly (\$1300-\$2000 list) to buy. They print onto special discs for which a premium of the order of ten cents is typical; that means that they are desirable for both cost and quality when volume is high enough. With resolution ranging from 600 to 1200 dpi, they produce professional output and the manufacturers of bulk CD-R systems seem to prefer the Fargo Signature for high-volume applications. Note that the silk screening used on pressed discs offers only about 150 dpi; 300 dpi may be marginal for a letter, but it looks good on a CD-R.

If you are patient and watch carefully, you may be able to get a good deal on a CD-R printer. For example, Fargo recently upgraded their Signature product and offers refurbished units at a substantial discount from their list prices. Nevertheless, it takes a substantial printing load to justify \$1,000 or more to replace a \$3 pen or a package of labels.

Readin', Writin', Speedin'

Before we get to matters of reading and writing speed, there's a bit of housecleaning to take care of. First, if you do not understand spindown then, please read about it first. Second, some words about hard drives.

In the old days - when a big hard drive offered 250 MB in a full-height 5.25" bay - there was a problem holding tolerances as the drive heated up with use. Those drives occasionally took time out for thermal recalibration. Drives which did continuous recalibration were able to support continuous data streams, so were designated A/V (audio/visual). Modern drives have much tighter tolerances, so they all recalibrate continuously and don't stop working just for that. Unfortunately, the term 'A/V' has been retained to designate very high-speed drives. But what we care about is that you do not need an A/V drive for CD-R. However, remember that if you let your hard drive go to sleep, it will have to spin up again. Now, that's out of the way.

Writing

Okay, how fast should you write? For most users, it is not important. If you are in production, turning out copies of your own Greatest Hits album, then you care. But since modern mastering software uses only a small amount of a modern system's power and permits you to do almost anything while burning, it is no longer a big deal. If you are going to burn a half dozen discs a day or fewer, you will feel less pain waiting for a 2x burner than you will paying for an 8x. It's also a lot easier to find media which write well at 2x than those that behave at 4x or 8x. (They tend to be less costly, too.)

Reading

This primer is about CD-R. If you want a high-speed reader for some other reason, fine - but that is not of interest here. You want a CD-ROM reader to read discs for writing. In order to write "on the fly" - directly from reader to writer through mastering software - your source must supply a steady stream of data as fast as your writer will write them. The hardware provides some buffering of those data, but that is only enough to handle a short transient, typically 6-7 seconds. Modern mastering software provides additional buffering from spare system RAM. Still, eventually the buffer will be drained if the source dries up long enough.

To be sure that your source does not lag behind the demand, it must be able to supply data continuously at the needed speed. A rule of thumb is that the source should be twice as fast as the writer. In general, a CD-ROM drive rated at more than 8x does not deliver constant speed. The speed on the innermost part of the disc may be less than half that on the outermost. As a result, a drive rated at 24x may not be able to provide data fast enough to keep a high-speed writer happy. Complicating the matter further is the fact that audio extraction is often substantially slower than data transfer. Some drives do not do DAE at all; others may be rated 24x but deliver digital audio at 2x, 1x or slower.

Surprisingly, there can also be a problem with a reader which is too fast. A very fast reader may fill the buffer very quickly, then sit in idle while the buffer is draining. If that takes long enough, the reader may spin down. When the buffer needs to be refilled, the source starts to spin up again. If it takes too long to supply the data, the laser can be starved and the dreaded buffer underrun may appear. There are many ways to address that problem if it occurs for you: reduce the software buffer, slow the reader or increase its spindown delay. In fact, it may also be solved by slowing the write; that would give more time for the buffer to be reloaded. This problem will only occur when writing on the fly, it is detectable during test, and it can be monitored by watching buffer utilization if your software displays that information.

Tips for Windows

Adaptec provides valuable information consistent with Jeff's - and freeware to make things work. For example, you can and should pick up ASPICheck from them. It determines the modules in your ASPI layer and reports whether or not they need to be updated. If you need to update, you can do it with freeware from the same site.

Where possible, put your pagefile (swap file) on a different physical drive from the one that you are using to store your data.

Defragment your HD before you read data or build an image from a CD.

Defrag the HD from which you will be writing before starting the burn.

Auto Insert Notification

Again, this takes editing the registry.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Cdrom

Enable AIN by setting the value of Autorun to 0x00000001 (1)

NOTE: This in itself does not enable Autorun (or Autoplay). For that purpose, I suggest using TweakUI from Microsoft's Power Toys.

Windows 95

Jeff Arnold offers:

The following tip can greatly improve CD-R performance under Windows 95. This could prevent buffer underruns and other common problems from occurring. By default, Windows 95 does excessive file caching. This is a waste of CPU time, memory, and possibly disk space, since you will never go back and use anything in the cache. To fix this problem, do the following...

1) Open the file SYSTEM.INI with a text editor (this file will usually be found in C:\WINDOWS or C:\WIN95).

2) Find the section of the file called "[vcache]".

3) Add the following lines after the "[vcache]" line.

minfilecache = 512

maxfilecache = 4096

4) Save the file changes.

5) Reboot your PC.

NOTE: If you have any problems with this setup, you can just remove the lines and reboot your system again.

A friend reports:

It appears that Win95 is not shy about putting duplicate drivers here and there. You may only see this in "safe" mode, so bring Win95 up in Safe Mode, go to System Properties, and look at Device Manager. If redundant devices appear that weren't visible in regular mode, deleted *all* the hard drive devices, floppy drive devices, and redundant devices ports, etc. Then reboot several times until Win95 finds all the hardware again and installs only *one* copy of each.

NT Notes

Windows NT 4 provides a fine environment for CD-R. Operation is quite similar to that in Win95, but a few comments are worth noting.

NT 4 Server *Provided by Glenn, glennb@algonet.se*

To use NT Server for burning CDR's successfully you need to use regedit and change the following registry setting: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management
Find a DWORD entry "LargeSystemCache=0". If the value is set to "1", file caching has higher priority than the application code in the fight for real memory. This has the nasty effect that when you handle large files, like a cd image, or a stream of files to be burnt, NT will page out the cdr-software executable (and everything else, except the kitchen sink) to the hard disk. Problem is that in the next instant NT finds that it needs to page the executable back in as it is actually running and handling these large files NT made room for... So the system will keep bouncing the program code in and out of memory, completely annihilating the system throughput, until you end up with a coaster. If I remember right, is this value set to 0 (as desired) as default on NTWS, while it is set to 1 (coaster city) as default on NTS. It is quite possible that the install (or use) of the Server resource kit caused this change to the "standard server value". This tip might be useful for anyone wanting to use NT Server when making cdrs. I run Server myself, and have no problems burning provided I do this single change to the registry.

A 650-MB WWW Site

Let me begin by saying that my primary use for CD-R is not that of most others. Before I bought my first writer, I defined what I wanted to do with it and how. The result has been (through May, 1998) seven CD-ROMs pressed for me and distributed through the Internet to a small but appreciative 'market'. This note traces the logic leading to the selected format(s) and suggests tools for following that lead.

Format: Why and How HTML

My objective has been to make files in various formats easily accessible to users with limited computer skills. I quickly reduced the options to two: Adobe Acrobat's PDF and HyperText Markup Language - HTML, the language of the World Wide Web. Acrobat offers several distinct advantages: well-defined formatting with esthetic values such as multicolumn text, explicit placement of graphics and full-text search. It shares with HTML the ability to link elements and to operate freely across platforms. HTML won out for me for several reasons, notably the ability of the user to run a browser she already knew and the flexibility to adapt that browser to her particular needs. If she is running a monochrome display, monochrome options are at her fingertips; if she needs larger type, she may simply set her defaults. In addition, creating and editing HTML is easy with tools of moderate cost and ready availability and, perhaps most telling of all, one can easily embed a PDF into an HTML file, but not HTML into PDF. Finally, PDF would have required the user to acquire a reader and to install it onto the computer; I wanted a self-contained, machine-independent configuration.

Because I want my discs to read on as wide a range of platforms as possible and with the least limitation on personal configuration, I use only the most primitive forms of HTML. Fonts are set with H1-H6 rather than by specifying them. There are no frames and very few tables; in fact, inline and background graphics are used to supplement text instead of to replace it. As a result, a user can browse any site or CD-ROM I create with virtually any browser and retrieve from it whatever her tools support. Another plus for HTML is that if a format begins to be supported, I can add it to my 'site' and the user can get it on her own for her platform. In a word, I can deliver the product we both want without requiring the user to adapt to my means of delivery.

To create the HTML, I use HoTMetaL Pro, a powerful language processor which implements far more than I need. One of its virtues is controllable enforcement of rulesets. I can pick the flavor of HTML I want to use and have the program enforce it as I write or after the fact. You will not be surprised to find that I use the most limited ruleset available and keep it on at all times. If you prefer a less costly approach, such as the shareware HotDog, that's fine, too. However, I do not recommend that you use HTML export or authoring from programs not dedicated to the purpose. They tend to produce code which looks just right to you - and fails miserably when seen on another system. In fact, they throw away many of the advantages which attracted me to HTML in the first place. They often implement the specific font choices you made when you wrote instead of leaving options open for the user. I want to create an open shop with its wares laid out for all visitors to see and to use; those programs tend to organize the shop into the specific pattern you see on your screen and to force visitors to see it your way.

My Recipe - Flavor as you wish

Unlike a site on the WWW where a few megabytes are stored, the CD-ROM offers 650 MB and needs careful organization. My discs are organized with a home page that provides direct access at the top to the major components, technical and legal information as required, and advice at the bottom on special needs for configuration and helpers. Because I work on a PC and most of my customers are on PC's, I develop first for that machine and then through beta test verify operation on other hosts (both Mac and Unix). However, I do nothing that I can avoid which would keep a user out. On my latest discs (read about them at my other site, <http://mrichter.simplenet.com/>) I do require MP3, which is not built into all browsers. In fact, thanks to lack of imagination by programmers, only a few configurations work on the Mac and the recipe for success on that host is not convenient.

I include on each disc everything needed to get up and running. For a novice on a PC (running Win 95 with Autorun ON), the disc is inserted into the reader and autoruns. I bought a royalty-free license for I-View, a compact and flexible browser designed for use off-line. I obtained permission from the authors to include freeware not only for WinAmp but also for a suitable Mac browser and helper. A Mac user can run an existing browser (or may look at a MACREAD.TXT file in the root) to install the included browser and helper on his own machine. Of course, if the user prefers another browser and is willing to add the MP3 or other helpers required, that option is also open. Again, everything is kept as simple as possible; elementary menus are used instead of frames or tables; the user is given the greatest possible flexibility.

With HTML, cross-referencing is my job before I create the disc, but that access is easily supported. Suppose I want a record of my family's history and have video clips, still photos and audio recordings prepared for inclusion. I can supply a page of Uncle John with a portrait and text and with links to sound and video clips and to scenes in which he figures prominently. I can link to that page from a page with the family tree or from each page of narrative text in which his name is mentioned. I can even take a family picture and provide a link from his area of the photo to 'his' page. Note that the same page or the same file may be linked from each point where it seems fitting to me.

Special cases and methodology

On three of my CD-ROMs, I incorporate the history of my primary WWW site. That is easily done by taking the individual pages that have been posted and linking them either chronologically or by subject. The result is a large, instantly accessible site on the CD-ROM. Note that I have to be careful because of the way most browsers operate. For example, I use playlists of MP3 files, but neither MSIE nor NN will play them from the disc. I-View has no trouble doing so because it is designed for the purpose; to make a conventional browser work, it would have to know where to find the CD-ROM. The usual browsers load the playlist into the TEMP folder, then look for the referenced files there when they are actually on the CD-ROM. On a PC, they want the letter of the drive (which, of course, I do not know on the user's machine); on a Mac, they could use the volume label, but for some reason they do not. Only by testing can those cases be found and even there you need to be sure to use a variety of hosts and of configurations to be confident. (I goofed on one disc and some files are only playable on machines with more than 32 MB of RAM; I made a change which imposed that requirement after testing was complete and have been embarrassed by the awkward workaround for the finished discs.)

My way of building the disc relies on my own system hardware and preferences; yours may vary. I start with an outline of what I want to do on that disc and an overall map of its organization. I break that down into pages and groups of pages to be written. As I prepare material for each page, I write the supporting HTML and check it on that level. I store the pages on a Jaz and periodically make a CD-R copy for backup. (Typically, I have several projects in development and dedicate a Jaz cartridge to each. Then I can work on that project simply by selecting the appropriate cartridge.) My home page for the disc is built in outline form; fleshing it out is the last step and exploits what I have from previous efforts.

Periodically, I run HTML Power Tools to verify my linkages. I skipped that step on my prototype; as a result, it has both duplicate files and broken links. When I have a viable version of a new project, I send it out to testers who are knowledgeable in the computers they use and the subject matter. While they evaluate the work, I write the home page and work on other projects. As the testers feed back, I return to the job with a fresh eye and make changes as appropriate. That is also a time when I may add another cross index or additional text, but not additional files unless I'm prepared to ask for retest. I also contact the person who will press the disc and prepare the graphics and other materials needed for production. When the process is complete, I again run HTML Power Analyzer to verify the layout, then burn a couple of CD-Rs to check yet again on my PCs. Finally, I drop the discs into the mail to the pressing plant and erase the Jaz. I interact with my broker, Digital Bim, by e-mail and by telephone. About a month after mailing the masters, the finished product is delivered and ready for distribution. For your information, the cost of having discs pressed has been consistent for a couple of years: roughly \$800 for 500 copies, \$1000 for 1000 copies.

A New Medium - New Ideas

CD-R technology can allow us to realize familiar things in familiar ways. We can back up CDs and CD-ROMs, we can save archives to a nearly permanent medium easily and economically. We can transcribe tapes and LPs to CD's. We can even back up our computer systems. Yet there are potentials here which are not yet being exploited - at least, not extensively - that offer new opportunities.

Disc Jockey Time

We all know the virtues and the limitations of the CD: great sound limited to less than 80 minutes; continuous or selective play with the sequence predetermined on the disc or requiring programming into the player. But there is a viable alternative allowing more music to be stored on a disc and accessed in whatever order is appropriate for the job. Suppose that the tracks are recorded on a CD-ROM, not in CD-DA format, but in WAV or MP3. Then playlists can be constructed which preselect track sequences for specific purposes: an hour of slow dancing to end an evening, twenty minutes of increasingly upbeat tracks leading to the climax of the evening, three hours of Glenn Miller for the 'mature' audience, and so on. If special needs arise, one list can be modified in real time or a new one can be constructed while another is playing. A single CD-ROM can then hold ten hours or so of music to fill a full gig.

Playback? Instead of carrying a stack of CD's and CD-R's to the party and hoping that the host's CD machine is good enough, carry one disc and a laptop you know. Connect its line out to the local sound system and you have full control right from your computer. With a program such as WinAmp from <http://www.nullsoft.com/> you can call up a graphic equalizer to tone things down or to tune the sound to the local system. WinAmp and other programs for WAV and MPn also support simple scripting and both sequential and quasi-random playback.

Footnote: Two at a time!

So, does it make sense to put both of those ideas together? You bet it does! When I'm working on a disc, I write the HTML and the files - usually, MP3 and JPEG - at about the same time. I use only relative addressing and provide a folder for each logical collection with all required files. Then I can check things out with browsers on a folder level. When I'm ready, I add a home page linking to the HTML in the folders, add I-View, WinAmp and whatever else is appropriate, run some checks with analysis tools and play with it a bit - and then burn away.