Background

Neither the vendors of shrink-wrap software, nor their customers are well served by existing copyright law. Software vendors have been trying to rectify half of this problem by inventing the concept of shrink-wrap "licenses". Needless to say, these shrink-wrap licenses tend to be one-sided because the purchaser has no influence in their terms. Thankfully, they are also pretty much not enforceable (or at least, that's what the computer press has stated over and over again). To be fair, most of the ones I have read pretty much restate the fair use provisions of copyright; however, some contain some pretty onerous provisions – Like gag rules preventing one from publicly criticizing the software in question. The latest power grab is UCITA, which allows software vendors to write any kind of license they want and force purchasers to agree to the terms.

Let's look at what the various parties have at stake here.

Purchasers

Purchasers of shrink-wrap computer software range from the largest corporations to individual home users. The needs of business users and home users vary in terms of the impact and urgency, but are pretty much the same. These needs are

- Help installing the software and getting it to work (i.e., training).
- Bug fixes for defects that prevent the software from doing its job.
- The software must continue to work. Purchasers will end up spending many thousands of hours creating data with the software. If the software stops working, they will most likely lose the fruits of that labor. In the long term, purchasers become locked into using particular pieces of software. Indeed, entire industries are locked into particular software products that have become de facto standards. Examples of computer programs that are de facto standards in their respective industries are Microsoft Word, Microsoft Excel, Adobe Photoshop, Adobe Illustrator and Digidesign Pro Tools.

If the first two needs are not going to be met, it will become apparent to the purchaser before they become locked into the software product: All they need is the ability to return it if it doesn't work for them. Most vendors do allow this. Even if they don't, the purchaser may still be able to sell the software to someone who can use it or just write it off as an experiment gone bad (the cost of ditching the software is usually not prohibitive at this point in the cycle).

The last requirement is where most of the problems arise. The consumer needs

1. A reliable mechanism for backing up the software so it can be restored onto new hardware should the computer it is installed on fail. For some business users, any delay in this area can be extremely expensive and could even be fatal to the business. For example, if a top flight recording studio has a hardware failure, they are expected to be operational within a few hours tops: If they can't be, then they are expected to reimburse the clients for all costs associated with the failed session. And that may not be enough to remedy the situation: If the client was on a critical deadline they could lose the client and gain a lot of bad PR to boot. Restoring operation should at

most be a matter of wheeling in the replacement computer and restoring files onto the second computer in the exact configuration where the first one failed.

2. Updates to the software to keep it compatible with new operating systems and new hardware.
3. In many cases, the vendor and purchasers are on a collective joint venture refining the software to make it do its job better.

Vendors

Vendors have capital or sweat equity (or both) invested in the software. They deserve to make a profit on it (assuming the software really works). They also need an income to pay for providing the long-term support the customer needs to make productive use of the software.

The main need in this area is to ensure that the vendors get paid for the copies of their program that are in use (both for new users and for upgrades).

Vendors generally sell shrink-wrap software at very low prices in high volume. Given how much they get paid, they shouldn't be held liable for damages resulting from honest mistakes. As long as they allow the software to be returned, they should not be held responsible for whether the functionality of the software addresses a particular purchaser's needs: The purchaser should make this determination within some reasonable amount of time after purchasing the software.

Current State of Affairs

Currently, software is sold as a copyrighted work. The vendor's legal department may think they are selling a license, but most purchasers think they are buying a copy of the software, not a license to use the software. How well does this protect the various parties' interests?

In theory, copyright law ensures that the purchaser's need #1 is met (computer backups are clearly fair use). In practice, some vendors have implemented copy protection schemes which do not allow for reliable backup and timely restoring of their applications (especially when used in a business environment). Copyright does nothing for needs #2 and #3. If the vendor decides to stop supporting the product (for whatever reason), there is nothing the purchaser can do. This phenomenon is known as "abandonware".

The vendors are fairly well protected by existing copyright law. They can prosecute people who copy their software without authorization. They can also sue to collect monies due them. The hard part is figuring out who is doing the copying and no new law (or shrink-wrap license) is going help that.

Copy Protection

Inappropriate copying of software is the major risk vendors are exposed to; so, they obviously take steps to keep it from happening. A large part of the problem is that a lot of purchasers don't realize exactly when it's illegal to copy software (if they even know it's illegal at all).

Copy ProtectionTerminology and Background

Removing copy protection mechanisms is called "cracking". Programs which have had the copy protection removed are called "cracks" and described as being "cracked". Programmers who remove

copy protection are called “crackers”. I do not know anyone who does this, but hearsay and anecdotal evidence indicates that they do this because they believe the copy protection is taking unfair advantage of purchasers and/or because cracking the copy protection presents a technical challenge.

Making a copy of the contents of a computer’s hard drive(s) is known as backing up the computer. The resulting copies are called backups. Backups have two functions: To preserve the information should the hardware fail and to allow for retrieval of information that is accidentally deleted. Retrieving information from a backup is known as restoring. A good backup scheme should include making multiple backup copies and keeping at least one of those copies off-site. One makes multiple copies because the backup media itself is not 100% reliable. Keeping copies off-site is protection from losing all your data in the event of a major disaster (such as a fire). A good backup scheme is necessary for computer systems that are used for commercial purposes.

Types of Copy Protection

Copy protection schemes range all the way from simple reminder mechanisms to external hardware devices that must be present before the software will run:

The Friendly Reminder

This is the simplest kind of copy protection. The software will periodically ask for the user to enter a password or serial number. The serial number usually must be obtained by paying the vendor. The software will operate forever without the serial number, but will keep reminding you to pay. Very small companies that let you download the software over the Internet usually use this scheme. The price is usually very low (relative to the price of similar products that are heavily marketed). These schemes are generally programmed in the most straightforward manner and could be removed by a cracker with a trivial amount of work: They usually don’t get cracked (it’s either not enough of a challenge or the products are not considered “rip-offs” by the crackers – I can only guess). One vendor who distributes this way says he checked all the places where cracked software usually shows up and his didn’t show up there. This scheme is obviously user friendly. I have not had any problems with this mechanism.

One Time Password or Serial Number

This kind of protection is similar to the friendly reminder except the program will not run until you enter the serial number. It is sometimes set up so that you have to re-enter the password if the program has been moved to a new computer, but the same password can be used on the new computer. Large vendors (Adobe, Microsoft, etc.) usually use this scheme. The password is almost always included in the retail package, but not printed on the CD case (it’ll either be in other documentation or in the manual). This scheme is also purchaser friendly as long as they are together enough not to lose the password. My experience with this mechanism is that it is trouble free.

Install Disk Validation

This scheme has been turning up very recently. It involves using a specially manufactured CD-ROM (“magic CD-ROM”) that presumably can’t be copied using CD ROM burners (or at least not easily). You need to insert the CD-ROM to install the program on a new computer (even if you copy the program file(s) using some other mechanism, such as restoring a backup). One vender I buy from requires the CD-ROM to be inserted every 6 months. At first glance, this seems pretty good. The purchaser can restore onto a new machine and get going without much extra effort. It will take several

minutes per application, though (whatever the "magic" is, it takes a long time to check the disk). The thing I worry about is the "magic". They used to use this technique with floppy disks and the "magic" involved doing things that compromised the data integrity of the disks -- it turned out to be unreliable every time a new generation of floppy disk hardware came out. We may end up finding out that these new "magic CD-ROMs" don't work with DVD drives, for example. If this happens, the vendor will have to manufacture new CD-ROMs. They will, of course, pass this expense along to the purchasers. It's also possible that the magic may affect the long-term survival rate of the CD-ROM. This scheme requires that the CD-ROM be stored near the computer: That makes it subject to theft and damage in a business environment. I haven't experienced any problems with this scheme, yet, but it's only been around about 6 months.

Key Disk Authorization

With key disk authorization, the purchaser is provided with a floppy disk. The floppy disk will have one or more authorization tokens that can be transferred to a hard disk on the purchaser's computer. The software will run as long as the token remains on the hard disk (the token can also be transferred back onto the floppy disk). This scheme suffers from many problems. For starters, if the hard drive fails, you lose your token -- And the hard drive is the least reliable component in a computer. Some vendors provide multiple tokens to give you time to get a replacement key disk, but others do not. To make matters worse, the floppy disks also have the "magic" mentioned in the preceding section applied to them: They have historically suffered from hardware compatibility problems -- The most recent one being that most new computers don't come with floppy drives and the magic only works with two specific USB floppy drives – One of these is out of production and the company that manufactures the other is out of business.

This type of copy protection has also been unreliable due to software bugs. I don't know if this is because it's unusually difficult to write or because the company that supplies it is incompetent (most vendors buy copy protection from one of a handful of suppliers). In my experience, the failure rate of this type of copy protection has been significantly higher than the failure rate of the software it protects. Another drawback of this scheme is that software updates for copy protection bugs must involve distributing new key disks. Not only does this expose the purchasers to the bug for a long time, but also the vendors have to buy new disks from their copy protection supplier in order to fix the supplier's bug (and of course, they pass the cost along). One of the most effective ways to improve the reliability of software in the field is to get bug fixes out to customers as quickly as possible: This scheme fails miserably in that regard.

This scheme also requires you to insert the key disks to "validate" the token at essentially unpredictable times. This means that the key disks must remain near the computer where they are subject to theft and damage.

I have had significant operation problems caused by this scheme. In one case, I was doing a remote recording and the software mysteriously asked to validate the key disks. This was before I understood I needed to keep them with the computer at all times (It's not mentioned in the documentation); so, the key disks were stored in a safe place that was an hour drive from the recording location. I was lucky that I was working with a friend that day and we were doing overdubs. An event like that could have been a major disaster for my recording business. In another case, it took more than a month to get a key disk replaced: Some of the companies have trained their customer support people to assume that

anybody calling for a key disk replacement is a thief and they often either flat out refuse to replace the disks or play games to get you off the phone. Both times I have had to get a disk replaced it has involved multiple contacts with the company and fighting my way past front line support people to talk to management. In the first case, the key disk was defective when it arrived. The second time the key disk needed to be replaced because of a software bug in the copy protection software that the company knew about!

I know that I am not the only person having these kinds of problems. One vendor who dropped this form of copy protection (Arboretum) stated publicly that their customer support calls were cut in half when they dropped key disk copy protection. Given that the copy protection software is a tiny part of the overall software package, this is an abysmal reliability record.

This type of copy protection uses all kinds of encryption and code obfuscation techniques to try to make it hard to crack. In spite of this, one of the vendors who uses it has stated that it really irritates them to see their new release show up cracked on the web within 24 hours of it being shipped. In other words, even the companies who use it admit it isn't very effective at preventing illegal copying. There is also a product out called Key Disk Terminator that removes this type of copy protection. Key Disk Terminator has survived court challenges (because it can be used by legitimate owners for backups) and the DMCA (because the Library of Congress has exempted this type of copy protection from the DMCA). I haven't been willing to try Key Disk Terminator myself, because if the copy protection detects any trace of Key Disk Terminator on your hard drives it destroys all authorization tokens on your machine AND any floppy disks you put into the machine. I believe this behavior of the copy protection software is actually a violation of the Computer Fraud and Abuse Act (it would be considered a Trojan Horse), but I guess nobody has been willing to press charges.

Challenge/Response Authorization

Challenge/Response Authorization works by generating a challenge (generally printable gibberish) that is based on some hardware in your system. You send the challenge to the vendor. The vendor will send you back a response (more gibberish). Once the response has been entered into the software, the software will run. Most vendors also allow the software to run for at least a few days after it is first installed without the challenge/response. This is to cover the hard drive failure scenario and also allows for a demo period. The biggest problem with this is that the challenge is usually generated from a hard drive serial number and is thus dependent on the least reliable part of the system. It also depends on quick response from the vendor: Some are really good and have a web site implemented to give instant responses to registered users – They can, after all, straighten out any misunderstandings after the fact. Others tend to treat anybody who calls more than once as a criminal or flat out refuse to give an authorization when the purchaser bought a new computer. And then there is the one vendor who habitually takes two weeks to answer e-mail.

Another problem with this scheme is that the primary company supplying it is the same one that supplies the key disk authorization scheme. Their compatibility track record over OS upgrades has been less than poor.

There is supposedly a web site on the Internet that will provide the response for any program; so, it's pretty clear this scheme has been cracked as well. I believe Key Disk Terminator is also capable of removing it.

I haven't had any bad experiences with this form of copy protection; however, I also haven't upgraded my OS or hardware since the first product came with it. Several people in an Internet discussion group for Mac based audio have complained about a few particular manufacturers refusing to provide responses. In one case, the purchaser was offended by the whole process and started out being rather undiplomatic; therefore, one has to chalk that case up as a personality conflict, not a policy. In the other case, the vendor refused to supply the response because the software was several years old and they weren't supporting it any more. I don't feel that was appropriate.

Hardware Dongle

A hardware dongle attaches to one of the I/O ports on the computer. The software will run if the dongle is there and either won't run or runs in demo mode if the dongle is not there. The obvious disadvantage here is that the software depends on a piece of hardware and if the hardware fails, the software won't work. In practice, this hardware is very simple and doesn't fail very often. The real problem with dongles is that large numbers of them on one machine run into trouble (usually with power consumption, but they can encounter software and/or hardware limits, too). The advent of USB has alleviated those problems to a large degree. There are also occasional software problems, but they are easily fixed with an Internet download.

This system has also been cracked (by Key Disk Terminator according to their ads and I am told that illegal cracks are to be had on the Internet). The biggest problem I have heard of with dongles is theft.

I have several products using dongles and have had no problems with them.

Online Verification

One of my consulting clients provides customer support at a level where they monitor the customers computers for problems. They also monitor for use of applications the customer hasn't paid for and address the issue discreetly after the fact (it usually turns out the salesperson has told the customer it was OK to try out the software prior to purchase and didn't bother to tell the support people). At the moment, this level of customer support is too expensive for shrink wrap software and I don't think most people would particularly go for allowing a vendor to scan their computer without them receiving a lot of benefit for it.

Apple Computer has rolled out software that will automatically update your computer software for you. If they extended that, they could probably offer to distribute vendor upgrades at the same time; so, maybe in time we will see a different form of copyright compliance program that customers get a benefit from at the same time.

Copy Protection Conclusions

There are two basic facts about copy protection:

- No matter how complicated the copy protection is, it doesn't appear to do much beyond making the purchaser aware they shouldn't be copying the material. One can't state this as an absolute fact, because it is impossible to prove one way or the other.
- The more complicated schemes damage legitimate purchasers.

UCITA

From what I can tell from reading UCITA, it attempts to address both problems with contracting for custom software and also tries to legitimize the totally one-sided shrink-wrap licenses vendors have been trying to pretend apply to purchasers. Custom software procurement is extremely complicated; so, I'm not going to deal with that. Without getting into details, I see several major problems with the shrink-wrap provisions of UCITA (I'm using shrink-wrap generically to include "click to agree" things on web sites and other similar mechanisms).

- These agreements are totally one sided. In rare circumstances, the purchaser may actually be able to decline to purchase the software and purchase from a competitor. But in most cases, the customer is locked in because they are upgrading [Remember that UCITA lets the vendors change the terms at any time after the software is purchased.] or because the program in question is an industry standard.
- The purchaser does not generally have legal council present when they are asked to "agree" to the terms.
- Almost all of these agreements that I have bothered to look at are long complicated legal documents. I would not be able to properly understand them without help from council; although, they do appear similar to each other. Furthermore, I seem to encounter several of them on a daily basis (I own around a hundred software products; so, it seems I am always installing upgrades). My computer IP lawyer charges $280 per hour. Assuming it would only take them 1/4 hour per agreement and the average is two agreements per day, it comes out to $50,000 per year, which is a bout 60% of my gross income -- I would define that as an unfair burden. And I'm not even counting how much of my own time would be lost because I had to stop what I was doing and send a contract to my lawyer before I could proceed.

What is Needed

I think what is needed are two standard contracts that apply to all shrink-wrap software purchases unless a signed contract is executed before the software is purchased [in an online environment, signing a document must mean exchanging digital certificates, not just clicking on a button that says "download"]. One standard contract would apply to software for home use and one would apply for business use. The vendor should state which applies (and could reasonably be expected to charge more for a business use version of the same product). There are three over-riding principles I am suggesting here:

- Copyright is a good basis for the standard contracts. In that light, I will state what I think most people interpret fair use to mean for home and business use (it has never been totally clear).
- Purchasers need more protection from software becoming obsolete.
- Copy protection does nothing for the purchaser and probably does nothing for the vendor; therefore, purchasers shouldn't have to pay for it.

Common Features for Home and Business Use

The purchaser should be able to make one backup copy of the original distribution media. [I actually think this is of dubious utility unless the media is functioning as a copy protection key, but it seems to be important to other people.]

If copy protection prevents copying the media using normal procedures, the vendor should provide two copies immediately and should be required to replace those copies at no cost to the purchaser **any time** they should become defective in the future. Defective in this case must include both media failure and software incompatibilities. [In this case, copy protection is preventing the user from making an adequate backup copy of the software; so, the vendor should be responsible for providing and maintaining the viability of the backup.] Releasing a non-copy protected copy at a later date should release a vendor from this obligation.

The purchaser should be able to make any number of backups of the software as part of the backup of the entire computer the software happens to reside on. [Most shrink-wrap agreements allow for one backup copy of the software without specifying whether they are referring to a copy of the original media or general computer backups. Given that computer backup programs don't allow you to skip saving applications and even if they did, skipping the software when making the backups would not allow for timely restoring in the event of a hardware failure. I don't think this is an intentional omission from shrink-wrap agreements. Is defining what a "backup copy" means difficult legally?] Copy protection should not interfere with backups and should not increase restore time by more than one minute per application. Any scheme more invasive than that should be considered damaging.

The vendor should maintain copy protection support forever at no cost to the purchaser. Releasing a non-copy protected copy at a later date should release a vendor from this obligation. Beyond copy protection support, the vendor should only be required to provide the customer support advertised at the time the software was purchased.

The purchaser or their agents should be able to reverse-engineer any part of the vendor's product to the extent necessary to ensure interoperability with other software and hardware products and to debug software problems. In this case, purchasers explicitly include vendors of other software products. The vendor should not be legally responsible for supporting the reverse-engineering process or the resulting software interface. Any information regarding interfacing between software products should be free from any sort of trade secret restrictions (whether released by the vendor or discovered by third parties using reverse engineering techniques). Reverse engineering should not be authorized [by any standard shrink-wrap license] beyond what is needed for interoperability.

The purchaser should be responsible for determining whether the software is fit for the job they need done. To that end, the vendor should either allow returns for some period of time or allow for some kind of demo. [One usually can't tell what a software package can really do without actually using it for a while.]

Vendors who cease supporting a product should be required to either sell it to somebody who will support it or release the source code to the public domain to allow its purchasers to fend for themselves. This should apply even if the vendor applies for bankruptcy protection; i.e., the users of a

software product should be considered stakeholders in any bankruptcy proceedings and this consideration should include not curtailing software support during any lengthy legal proceedings.

Purchasers should be able to sell the software to a third party. If they do this, they must destroy all copies of the software on their computer, including older or newer versions. In other words, if the purchaser upgrades to version 4.0, the purchaser can't sell version 3.0 to a third party and keep using version 4.0. The vendor should be able to charge a reasonable fee (10% or the list purchase price, perhaps) for transferring the ownership.

All data created by the purchaser using the software is the purchaser's property.

Business Use

For business purposes, the fair use concept should be refined to include:

Single Machine Copy: The purchaser can use a copy of the software on one computer that is shared by multiple persons OR the software may only be used by a single person (who might keep copies of the software on multiple machines).

Network Copy: The purchaser can install the software on as many machines as they want, but may only use N copies at any given time. [This kind of scheme works using a network authorization scheme. This might be outside the scope of a standard contract.]

Fair use does not include the right to distribute copies of the software on any mass media (including the Internet).

The vendor should not be liable for consequential damages unless those damages resulted from a copy protection failure. In this case, a copy protection failure should be defined as anything involving copy protection that stops the software from working properly after the software has been installed and become operational. It should also include any damage caused by the copy protection in the event that the copy protection mistakenly decided it detected software the purchaser had not purchased. In the event that the copy protection failure was proceeded by other failures, the damages should only include damages attributable to the copy protection failure.

For legal actions concerning damage caused by copy protection, choice of venue should go to the purchaser. In all other cases, choice of venue location should go to the party that is not instigating the legal action.

Home Use

For home use, fair use should be refined to mean that the purchaser and family members living with them may use the software on every computer in their home except computers used for a home business.

Fair use does not include the right to distribute copies of the software on any mass media (including the Internet).

The vendor should not be liable for consequential damages.

Venue location choice for legal disputes should go to the purchaser. It would be reasonable to require arbitration as the first step in resolving any legal dispute.

Abuses which should not be in a shrink-wrap Contract

Gag rules [Some real examples: "The customer shall not disclose the results of any benchmark test", "The customer will not publish reviews of the product…"]

Choice of legal venue that automatically gives a huge advantage to the vendor.

Many shrink-wrap licenses have the following "This License Agreement will not be governed by the United Nations Convention of Contracts for the International Sale of Goods, the application of which is hereby expressly excluded." I don't know what this is about, but it has sleaze written all over it…

Claims to any data created by the software.

My source for most of these is:

http://www.cptech.org/ecom/ucita/

One more anecdote from a Slashdot user (unedited quote):

Most of the retail software I see has a seal on the box (and CD) saying "By opening this pakage, you agree to the terms and conditions on the license agreement contained inside". OK, You open the box, not knowing what you agreed to; You read the license, say "no way in heck!", put the box back together and take it back to the store. You guessed it! "You cannot return opened software"

Several people on that forum posted that they tried to return software because they disagreed with the shrink-wrap license and were not able to.