;login:

THE MAGAZINE OF USENIX & SAGE

October 2002 volume 27 • number 5

inside:

THE LAW

Nicholson: Software Licensing 101

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

software licensing 101

by John Nicholson

John Nicholson is an attorney in the Technology Group of the firm of Shaw Pittman in Washington, D.C. He focuses on technology outsourcing, application development and system implementation. and other technology issues.

John.Nicholson@ShawPittman.com

We've all seen them – pages and pages of two-column text, generally in 8-point type. In real life, we might get a CD in a jewel case with lots of small text printed on it and a warning that says that opening the box means that you accept the license. When we're online, we get an easy "I Accept" box that doesn't require reading all the gibberish before letting us install the software that we want. But have you ever actually read one? What are you agreeing to? Software vendors have lawyers who spend hundreds, if not thousands, of hours drafting these licenses to protect the "rights" of the seller. And in the case of Microsoft, Adobe, AutoDesk, Borland, and several others, they also have the Business Software Alliance (*http://www.bsa.org*) to collect reports of and prosecute unauthorized software use.[1] The only thing protecting your company is your understanding of the terms of the license and your (or your lawyer's) ability to negotiate with the vendor.

The purpose of this article is to give you some insight into what a software license should cover, provide an analysis of some sample language that vendors have used, and give you some tips for negotiating with vendors.

What Is a "License"?

The word "license" is somewhat misleading. A software license is a contract. According to WhatIs.com, "An End User License Agreement (EULA) is a legal contract between a software application author or publisher and the user of that application. The EULA, often referred to as the 'software license,' is similar to a rental agreement; the user agrees to pay for the privilege of using the software, and promises the software author or publisher to comply with all restrictions stated in the EULA."[2] So, in exchange for being allowed to use the software, you agree to comply with the restrictions imposed by the vendor. If you break the rules, you have breached the contract, and there may be specific remedies written into the license or you may have even violated some laws (e.g., copyright laws,[3] the Digital Millennium Copyright Act,[4] the No Electronic Theft [or NET] Act,[5] or others).

License Structure and Analysis

A software license tells you what you can and can't do with the software and what the vendor can and can't do. What the license says (and doesn't say) is just as important to you as it is to the vendor. In general, a software license should cover:

- Software included under the license
- Scope of use for the software and any restrictions on its use, as well as any different restrictions on the use of the documentation
- Duration of the license
- Related services (e.g., consulting, enhancement, help-desk support) that will be provided, and the terms under which those services will be provided
- Pricing and payment terms
- Confidentiality provisions
- Warranties and indemnities
- Limitations on liability
- Termination of the license and/or the services
- Other legal terms that are relevant (e.g., rights of publicity, choice of law)

Frequently, software licenses drafted by vendors will try to limit software use to employees of the company purchasing the software.

If the software is going to be used internationally, there may be specific terms that need to be discussed. For example, the US government restricts certain software from being exported to certain countries or at all, and you might have to apply for a special permit; or there are certain international treaties that may come into play regarding the sale of goods.

Software Included Under the License

The first question to ask when looking at a software license is whether it covers all of the software (and all of the functionality) that you think you are getting. It's important to make sure that you understand how the software covered by the license will provide *all* of the functionality that you think you are getting, and that it will continue to provide the same functionality in the future. Although this may seem obvious, there are certain provisions related to updates and upgrades that can affect the availability of functionality down the road. (See the section on updates and upgrades, below.) When you are discussing the software covered by the license, be sure to review in detail how the software will provide the functionality that you think you are getting. If the software cannot currently provide some of the functionality that you need, see whether there is a workaround, whether the vendor plans to provide such functionality in the future, or whether you will need to develop that functionality in cooperation with the vendor, using a consultant or on your own.

Scope

The scope of a license can vary dramatically, depending on the restrictions that the vendor wants to place on your use of the software. The license may specify:

WHO CAN USE THE SOFTWARE

Frequently, software licenses drafted by vendors will try to limit software use to employees of the company purchasing the software. This would mean that neither your contractors nor any consultants could use the software, which could be an inconvenience for you (if you complied with the restriction) or could result in a breach of the license (if you didn't). If you are only using object code, there isn't really much reason for this restriction. If you have source code, it would be reasonable for the vendor to insist that contractors and consultants sign a Non-Disclosure Agreement before being allowed access to the source code. And there might even be a short list of direct competitors that the vendor would want to prevent from seeing the source code under any circumstances.

FOR WHOSE BENEFIT

Software vendors want to sell as many copies of their software as possible, so they try to limit the number of people who can benefit from each installed version of the software. You, on the other hand, want to buy as few copies of the software as possible, so you want to be able to use it for your own company, your subsidiaries, other affiliates, your customers, your suppliers, etc. As long as you and the vendor come to an understanding regarding how the software will be used prior to executing the license, the vendor can take the scope of your use into account in its pricing.

FOR WHAT PURPOSES

Just as vendors want to limit access to the software by other companies, vendors want to limit access across business lines, if possible. By understanding how your company will want to use the software, you can reach a reasonable agreement with the vendor in this area.

IN WHAT LOCATIONS

If you plan to use the software in multiple countries, some vendors will try to argue that they need to get approval from regional segments of their organization in order to negotiate a deal. How the vendor organizes its own accounting is not your problem, and if a vendor wants to be a global player, it should act like one. (Note: this is just an extension of the "no authority" strategy discussed in the "Negotiation Tips" section later in the article.) It is possible, however, that there may be countries in which, for some legitimate reason (generally legal or regulatory), a vendor is unable to license you to use the software.

ON WHAT EQUIPMENT

Certain licenses are restricted to a particular-sized processor. Be certain that the license you are getting is sufficient for the use you intend now and throughout the term of the license. You will not put yourself (or your successor) in a good negotiating position if your business needs to renegotiate the license in the middle of the term because you did not adequately anticipate growth levels. If the pricing for the license is based on the processor (or other equipment) that you will be using, and there is a possibility that you might need to upgrade or expand your equipment during the term, negotiate the pricing for using the software on the upgraded/expanded equipment before signing the initial deal. Remember, your negotiating leverage is at a maximum before you sign.

Certain licenses are restricted to a particular-sized processor.

If the license is based on number of users or number of machines accessing the software, you should consider whether devices like PDAs would qualify. For example, one vendor's standard license prices the software by "client," which it defines as "an application that invokes, typically via a network protocol, the software functions provided by one or more servers." Under that language, a query from an application installed on a PDA could qualify as a "client," which could dramatically increase the price of the software.

PERMITTED COPYING

Vendors frequently include language such as, "Customer shall not make any copies of the Software, except for a single archival copy solely for internal purposes." This means that, for example, you would have to separately license development, test, and production instances of the software, and that you might also need to separately license any copies to be used for training. Since you must have at least a test region in addition to your production region, the license should include copies for you to use in those additional regions without an additional charge. Furthermore, if you have multiple locations, each running the software locally, you might want to have a backup copy at each location so that you can re-install via the local network. This is also a reasonable request.

Another thing to look out for in software licenses is whether the defined term "software" includes the documentation. Frequently, it does, and the impact of that definition combined with the language quoted in the previous paragraph would mean that you could not copy any of the documentation associated with the software in order to make training materials or procedures manuals. Training materials and other documentation provided by the vendor are likely to be generic rather than customized to your particular implementation. To customize your documentation, you might want to make sure that the license includes language such as, "Customer may copy, share, distribute, modify and create derivative works from the user manuals and any related documentation solely for Customer's internal business purposes."

WHETHER THE LICENSE IS TRANSFERABLE OR NOT

Frequently, a license will specify either that (1) the customer may not assign the license without the vendor's consent or (2) neither party may assign the license without the other party's consent. If your company is acquired or wishes to assign the license to another company (including a parent, affiliate, or subsidiary) for some other valid business reason, this provision could allow the vendor to refuse to grant its consent unless you satisfy conditions specified by the vendor. Language like this should be revised so that you can transfer the license to a parent, affiliate, or subsidiary without being required to get the consent of the vendor. At the least, you should get the vendor to agree that it will not unreasonably withhold any consent.

Duration/Termination of the License

The license should specify the term of the license. Large system operating software is generally priced by the month or the year. In the US, a perpetual license is legally acceptable, but outside the US you should discuss with a lawyer using a term of 20 years of less in order to avoid certain legal issues related to perpetual contracts.

In addition to knowing when the license will expire, you should examine very carefully the ways that the license can be terminated. A piece of software can be critical to your company's operations. Unless you don't pay for the license or you knowingly disclose the vendor's legitimately confidential information, the vendor should not be able to terminate the license. There are legal (i.e., financial) remedies that can compensate the vendor for virtually anything else that your company could do without threatening your company's ability to use a potentially mission-critical piece of software. On the other hand, you should be able to terminate the license and associated maintenance as long as you provide a reasonable amount of notice to the vendor.

Frequently, vendors will agree to allow you to terminate a contract early provided that you give them notice and pay a termination-for-convenience fee.[6] These fees range from a reasonable recovery of costs by the vendor (such as any investments that the vendor had to make to provide your particular services or to develop a particular function) to bordering-on-absurd attempts to include all of the revenues that the vendor would have received during the term of the contract. If you terminate the contract early, it isn't unreasonable for a vendor to recover costs that the vendor expected to recover during the term. The vendor shouldn't suffer harm because you elected to terminate a contract early. However, once you terminate the contract, the vendor won't be providing services or incurring costs, so there is no reason for the vendor to receive the total revenues that the vendor would have received for providing the services during the remainder of the term.

Nor is there much ground for the vendor to argue that it should receive the profits that it would have received during the remainder of the term. Vendors will try to use the argument that they have a right to lost profits because they shouldn't suffer harm because you decided to terminate the contract early. If the contract had never happened, the vendor would not have received the profits that they are trying to claim, so allowing the vendor to include lost profits in a termination-for-convenience charge puts the vendor in a better position than if the deal had never happened. The vendor should be allowed to recover costs, but not lost profits.

Payment, Acceptance Testing, and Warranties

In general, the vendor wants its money for the software right away (particularly these days). When negotiating, vendors look for payment on delivery (if not sooner) and in a lump sum. For example, one vendor's standard license agreement specifies, "Within 10 days after the Effective Date of this Agreement, the applicable Order Form and payment to [Vendor] of the associated license fees, Support Fees or other fees set forth in the Order Form, [Vendor] shall provide access to the [Vendor] FTP site to enable licensee to download the license Products and Documentation." This means that you have to pay for everything up front and only then will you be able to download the software and documentation. For something really simple and basic, this might not be too bad.

But for more complicated software that will require some implementation, you should be able to delay at least some of the payment until the software is successfully implemented (i.e., after acceptance testing). Vendors will argue that acceptance testing isn't necessary because (1) the product is proven in the market and (2) you're getting a warranty. However, the warranties frequently are effective for 90 days after *delivery* of the software. Most implementations take longer than 90 days, so unless the implementation is either simple or on a fast track, your warranty is effectively useless if it is based on the delivery date. Your warranty should begin running after the software has been successfully installed and tested (including integration testing), and you should be able to return the software for a full refund of the license fees if it isn't accepted.

Vendors want to be able to recognize the revenue from the sale of the software as soon as possible, and allowing you to return the software can interfere with the vendor's ability to recognize the revenue. Revenue recognition affects the vendor's earnings and the salesperson's incentive compensation. There are very specific rules regarding when revenue can be recognized.[7] If revenue recognition becomes an issue in your deal, you should have someone from your accounting/finance group who is experienced in this area working with your team.

For ongoing services or installment payments, you should look at the payment terms specified in the license. For example, one vendor's standard terms states:

> All billed charges are due ten (10) days from the invoice date. Licensee agrees that a copy of an invoice received by facsimile machine shall be binding on Licensee and have the same effect as an original. All balances ten (10) days past due will be subject to a ten (10) percent annual finance charge, and [Vendor] may elect to suspend technical support, software updates and enhancements, withhold shipment of computer supplies, and/or activate time sensitive devices . . . until [Vendor] receives said past due payments.

Not only is "Net 10" a short period of time for your company to make its payments, but the multiple remedies to this vendor of being able to charge interest, withhold support or other services, *and* potentially disable the software for failure to pay a bill within 20 days of the date the invoice was printed (not received by your company) is unreasonable. Because the license specifies an interest charge for late payments, the vendor is already being compensated for a late payment.

Remote Deactivation of Software

As mentioned in the language quoted in the previous section, vendors sometimes include in licenses the right to switch off the software or deny you access to date-sensitive activation codes if you haven't paid for something or if the vendor thinks that you are in breach of the license. For example, one vendor's standard form license states:

> Licensee acknowledges that some or all of the [software] may contain time sensitive devices which may be activated automatically, by [Vendor] or otherwise upon a material breach of this Agreement by Licensee, including without limit Licensee's breach of the payment terms . . . and Licensee's breach of the confidentiality provisions . . . or the expiration or termination for any reason of this Agreement. Upon activation, such time sensitive devices may alter or prevent the functionality of the [software]. Licensee acknowledges and agrees that such time sensitive devices are necessary to protect [Vendor's] intellectual property rights, that [Vendor] shall have no liability whatsoever for any outcomes caused by activation of such time sensitive devices and that Licensee shall be liable for all costs associated with the activation of such time sensitive devices, as well as costs associated with resuming normal use of the [software].

You should rarely, if ever, accept a license that allows the vendor to deactivate any part of your software with time-sensitive disabling devices. If the vendor can deactivate your software, you will likely give in to the vendor's demands rather than lose your functionality. Under the license, and under the law in general (at least in the US), the vendor will have legal remedies for any breach; the ability to remotely disable the software represents a self-help measure that gives the vendor unreasonable leverage over your company.

You should rarely, if ever, accept a license that allows the vendor to deactivate any part of your software with time-sensitive disabling devices.

Maintenance and Support

General

The standard vendor position regarding maintenance and support is that it begins after the warranty period (if any), is payable in advance, is non-refundable, and is subject to annual increases that are not capped in any way. From your perspective, you want the maintenance and support to be optional, since you might not need it or might be able to get it from someone else; you want to be able to pay in installments; and you want the price to be fixed during the term of the license, or if not fixed, then at least have the increases specified in advance and capped, to ensure that you understand the maximum amount of costs that you will have to pay.

New Releases, Updates, and Upgrades

Frequently, vendors will include new "releases," "updates," or "maintenance patches" in the cost of maintenance but will expect you to pay for "upgrades" to new "versions." Each vendor uses these terms differently and, frequently, in confusing ways. You should understand how the vendor numbers its software, and agree in advance to what constitutes something covered by maintenance (e.g., going from version 3.1.1 to 3.1.2) versus something for which you'll have to pay (e.g., going from version 3 to 4). It's reasonable for you to pay for significant new functionality, assuming that you want it and can use it, but it seems unfair for a vendor to require you to pay for the privilege of going through an upgrade to receive new functionality that you may not want or, in the alternative, risk losing your maintenance.

To minimize their costs of providing support, vendors try to limit the number of releases that they have to support at any one time. They do this by requiring you to upgrade to within a certain release level (e.g., "n–1") or by specifying that support for a previous version will be discontinued some period of time after the next version is released. On the other hand, upgrades are unpleasant and expensive for you. The more frequently you have to upgrade, particularly where an upgrade provides some change in the user experience, the less happy your users will be.

It's not unreasonable to insist on a maximum number of required upgrades in a given year.

One vendor's standard license states:

> [Vendor] periodically issues software updates that may require additional processing capacity (i.e., CPU memory or additional disk capacity) for Licensee's equipment. Licensee understands this requirement and agrees to purchase equipment as needed to remain current with [Vendor's] software releases. Licensee agrees to install each update and enhancement as soon as reasonably possible but in no event later than ninety (90) days after receipt. In the event that Licensee fails to so install any update or enhancement, then any warranty or obligation of [Vendor] with respect to the affected Program shall be invalidated.

The impact of the quoted language means that if the vendor releases an upgrade that required more processing capacity than your existing hardware had available, then within 90 days of the new release being made available, you *must* (1) upgrade your hardware to the specifications for the upgrade and (2) install the upgrade. Think of the impact that this could have on your budget and your upgrade schedule for your hardware.

Vendors frequently will not commit to a specific number of releases or the frequency with which they will produce new releases. It's not unreasonable to insist on a maximum number of required upgrades in a given year, nor is it unreasonable to expect vendors to keep up with industry-standard hardware/OS upgrades and industry-specific regulations.

The Legal Stuff You Usually Ignore

There are whole sections of contracts that business people regularly ignore because they feel like those sections cover stuff that is only interesting to lawyers. Although the lawyers are usually the only ones who argue about these issues, the business people really ought to pay more attention to them. Things like confidentiality, warranties, limitations of liability, and other terms that are always at the back of the contract can have a significant operational and financial impact; frequently, the vendors put things back there assuming that the business people won't be paying close attention.

Confidentiality Provisions

Some vendors' standard licenses define every aspect of their software and documentation to be confidential information. While it is not unreasonable for a vendor's source code or detailed technical documentation to be confidential, making the object code or user manuals confidential could create a problem. For example, one vendor's standard form contract states:

> The parties agree . . . that they shall not use, except as otherwise expressly permitted hereunder, or disclose to any third person, including any of its affiliates, or to any employee of the receiving party without a need to know, either during or after the term of this Agreement, any Confidential Information.

If the definition of "Confidential Information" were to include the object code or user documentation (which this vendor's license does), this would mean that except for employees of your company who actually have a need to see the software, no one else could even see the screen while the software was being used without it being a violation of this term.

Some vendors' standard agreements also have confidentiality provisions that are "one-way": they protect the vendor's information but not yours. If your company is hiring the vendor to implement the software or provide any other services, the vendor will be learning a great deal about how your company operates and would be under no obligation to keep any of it a secret.

Warranties and Indemnities

High on the list of things that non-lawyers traditionally ignore are the warranties and indemnities. In addition to the warranty that there are no disabling devices included in the software, the big one from your perspective is that the vendor should warrant that the software and documentation do not violate any patents or otherwise infringe on any other intellectual property rights. If someone claims that the software or documentation does violate their intellectual property rights, the vendor should indemnify your company for all costs and expenses associated with that claim.

Limitation of Liability

Next time you drop film off to be developed, look at the disclaimer on the envelope that says that if the developer screws up your pictures or even just completely loses the film, all they have to do is give you a new roll of film. This is a limitation of liability. A limitation of liability provision is a contractual term that specifies the maximum amount of damages for which a party can be liable under the contract.

In most standard vendor licenses, the limitation of liability generally only limits the liability of the vendor and limits it to an amount equal to something like the cost of the software or one year's maintenance. This means, in general, that no matter how badly the vendor screws up or how much it costs your company to fix the problem, the vendor only has to pay your company a maximum of what is specified in this clause.

Whether the cost of the software or the price of a year's maintenance is a reasonable limitation on damages is something that you and your business people can evaluate and negotiate. This provision should also be mutual, so that your company's damages are also capped.

There are certain standard exclusions to this limitation that are generally included in each contract without too much argument from either side. If you're really interested, your general counsel can tell you more.

Other Legal Terms

There are frequently a few other sections at the back of any agreement covering things like "choice of law" and "right of publicity" and other terms that sound trivial, and sometimes they are. However, if you are not careful, your agreement could be governed, for example, by the law of Virginia, which has passed a very vendor-friendly version of the Uniform Computer Information Transactions Act (UCITA).[8] New York is a state that has not passed a version of UCITA and is generally considered to have a well-developed body of commercial law.

Vendors like to publicize their sales. Some companies like to keep very close control over how their name and trademarks are used. Frequently, vendor licenses include the right for the vendor to issue a press release announcing that you are a customer and to use your company's name in their customer list. Your company may have a policy regarding publicity, and if you don't pay attention to the provisions at the end of the license, you could end up agreeing to a contract that violates your company's policy.

Modifications and Customizations

What happens if the software you are buying doesn't do *exactly* what you need it to? The first question is whether you're going to do something about it or just live with the lack of functionality. If you want to do something about it, then you will either need to modify or customize the software. "Modifications" to the software are modules developed by you or a third party that use pre-defined APIs in the software but do not require any changes to the underlying source code. "Customizations" are changes to the underlying source code, made by you, the vendor, or a third party.

Some vendors will encourage you to make modifications to their software, or they may offer to either assist you with developing them or develop them for you. Be careful, though, because some vendors' licenses give them the rights to any modifications you develop.

If you plan to develop customizations, you will need access to the source code for the software. This is not usually provided as part of a standard license. Some companies provide access to source code as part of a "developer's license" that is more expensive than the standard object code license.

Customizations can create risks, though, and the biggest is that a customization will take you out of the upgrade path (this is sometimes true with modifications, too). The best way to make sure that a customization or modification will not lock you into a particular version is to have the vendor develop the customization for you, but negotiating a development and implementation deal is a subject for another time.

Negotiation Tips

The most important thing in any technology procurement is to understand what business need the technology solution must satisfy and what that is worth to the company. Once you understand why you are looking for in a technology solution, the negotiation can begin.

As with any deal, negotiating software licenses is all about who has leverage, and, until you sign the deal, you have the most leverage you are going to have (in some cases, this is still very little, but once you sign the deal you have even less). Frequently, customers are reluctant to negotiate, believing that if they don't make things difficult for the vendor at the negotiating table, the vendor will take that into account during the relationship. It's a nice thought, but this almost never happens. By that logic, you would agree to pay the sticker price for a new car in the hopes that you would get better service from the dealer in the long run. It's important to realize that, in general, a vendor's standard initial position is a position that is very favorable to the vendor. It's simply the opening position in the negotiation. One of the most important aspects of a negotiation is to know what is reasonable and why. Hopefully, this article has given you some insight into that.

Both comparison shopping and getting references are reasonable things to do in any procurement. If there are multiple packages that can provide the functionality that you need, a competitive procurement increases the likelihood that you will get a better deal from whichever vendor you ultimately pick. Also, discussing your needs and a vendor's offering with one vendor will give you ideas for additional questions to ask the other vendor(s). You should also ask the vendor for references, including at least one customer that has stopped using the vendor's software recently. If the vendor has a users group, contact them, as well, and ask them for references similar to your company's profile and for a former user of the software. References can give you valuable information about issues with the vendor's software, how well the vendor supports the software, and the vendor's plans for the future.

When it comes time to negotiate, most vendors will try to send someone who does not have negotiation authority. He or she will agree to take your positions back to the vendor, particularly to the vendor's legal department, for review. It's the same principle that car dealers use – the salesman actually has no authority and acts like he is on your side, and it's just his manager who's being inflexible and unreasonable. Try to make sure that the person you are dealing with has the authority to agree to changes in the terms of the license. If they don't, then you're wasting your time negotiating with them.

To the extent possible, you should avoid using the vendor's standard form license. Any document can be written any number of ways, and software vendors have had lots of practice at drafting agreements that are interpreted in their favor. If you do have to use the vendor's form, have an experienced attorney look at it to identify areas of concern. Vendors are very good at writing provisions that sound very reasonable but that can have serious consequences to the consumer. You may not be able to do anything about them in negotiations, but at least you will understand the risks of a particular deal.

The common arguments that vendors will make when resisting changes to their form are that the changes will adversely affect their pricing model (i.e., they'll have to charge you more), that they will affect the vendor's ability to recognize the revenue, or that the vendor will have to make special accommodations to manage your contract if you get terms that are different from everyone else. The only one of these that holds any water from the customer perspective is that some provisions of a software license allocate risk (for example, a warranty pushes the risk of a failure onto the vendor), and if you change the allocation of the risk, you may actually legitimately change the assumptions underlying the vendor's pricing. However, vendors tend to use this as an argument much more often than it is merited. If the vendor argues that a change will affect the vendor's ability to recognize the revenue from the sale of the software, ask to see the opinion of the vendor's accountant.

Conclusion

If all of the above wasn't enough to make you want to look through your software licenses with a pair of lawyers and a fine-tooth comb, here's a recent quote from a Microsoft EULA for the patch that fixed a security problem in Windows Media Player:

> You agree that in order to protect the integrity of content and software protected by digital rights management ("Secure Content"), Microsoft may provide security related updates to the OS Components that will be automatically downloaded onto your computer. These security related updates may disable your ability to copy and/or play Secure Content and use other software on your computer. If we provide such a security update, we will use reasonable efforts to post notices on a web site explaining the update.[9]

By downloading and installing that security patch, which you really needed to do, you would give Microsoft the authority to automatically dump software onto your machine, and the only thing they would have to do would be to make a reasonable effort to post a notice about it somewhere on a Web site.[10]

Your software is letting your company do its business, and the licenses are what control how you use that software. Understanding the reasons why your company wants to use a particular software and the terms and conditions governing its use are critical issues if you are going to be involved in negotiating or administering software licenses. At the very least, read the licenses of the software for which you are responsible so that you can have a clear understanding of what you are and aren't allowed to do.

Notes

1. This article provides general information and represents the author's views. It does not constitute legal advice and should not be used or taken as legal advice relating to any specific situation.

2. *http://whatis.techtarget.com/* under EULA *(*as of Sept, 3, 2002).

3. US Code Title 17.

4. PL 105-34 (1998).

5. PL 106-113, 113 Stat 1501, 1501A-521 (1999).

6. Note that there is a difference between "termination for convenience" and "termination for cause." Termination for cause is when the vendor has breached the contract in some way and you are firing the vendor. Termination for convenience means that you have simply decided that you don't want to use the software or the services any more.

7. American Institute of Certified Public Accountants Statement of Position 97-2.

8. VA Code §§ 59.1-501.1 through 59.1-502.1. Maryland is the only other state so far that has passed a version of UCITA, but it is being considered in a number of other states. Three states, Iowa, North Carolina, and West Virginia, have enacted "bomb shelter" anti-UCITA statutes to protect their citizens from the effects of UCITA provisions in shrink-wrap or click-wrap licenses.

9. As quoted in Thomas C. Greene, "MS Security Patch EULA Gives Billg Admin Privileges on Your Box," posted at *http://www.theregister.co.uk/content/4/25956.html*, as of 7/01/02.

10. For a humorous take on this, see J.D. "Illiad" Frazer's comic strip, "User Friendly," for July 6 and 8, 2002, at *http://www.userfriendly.org*.