

A Response To:
Why Software Professionals Should Support
The Uniform Computer Information Transactions Act
(And What Will Happen If They Don't)

-

This responds to the two papers written by Lorin Brennan and Glenn Barber in response to my paper in this Journal. (An edited version that merges these appears at <http://www.2bguide.com/docs/proucita4.doc>) I know Lorin Brennan, I respect him, and I enjoy his company. He has been a well respected participant in the UCITA (UCC 2B) drafting process. I don't know Glenn Barber, but his credentials are impressive and his partnership with Lorin makes me more, rather than less, inclined to trust him.

The Brennan/Barber responses are representative of widely shared opinions. Most of the comments they made could have been made (many have been made) by many of the other proponents of UCITA.

That said, let me say next that I flatly disagree with most of what they have said in their responses to my article. My biggest challenge in writing this rebuttal has been keeping it within a few thousand words. There's a lot more that I'd like to say than can fit within *Software Quality Professional's* (entirely reasonable) length limitations.

In this article, I refer to the two papers collectively as "The Response."

The Nature of the Opposition

The Response starts out by characterizing the opposition to UCITA.

"Not everyone supports UCITA. Some groups want to increase the liabilities and restrictions on the software profession. They want to limit on-line contracting, and impose greater risks on developers who make available innovative and alternative programs such as Java or Linux. "

There may be such groups, but I don't know them. This characterization doesn't reflect the opposition that I've seen.

- ***We don't want to increase the liabilities and restrictions on the software profession.*** We want NCCUSL to leave software law alone. The courts have been applying Uniform Commercial Code Article 2 (the law of sales) to software for the past 30 years. The software industry has done well under current law. Our proposal neither increases nor decreases liabilities or restrictions on anyone.
- ***We don't want to limit on-line contracting.*** The primary law coming out of NCCUSL for on-line contracting is the Uniform Electronic Transactions Act (UETA), not UCITA. I support the UETA (as do software publishers, I believe). What I don't support is a collection of unnecessary provisions that will make it easier for unscrupulous sellers to get away with bad practices. For example, I don't oppose click-on contracts. I oppose contracting practices that don't *allow* you to see the contract until after you've paid for the product and accepted delivery. I don't

oppose rationalizing the choice of law and forum ("forum" means where you can sue). I oppose giving the publisher virtually unrestricted, unilateral choice to resolve this.

- ***We don't want to subject independent developers to greater risks.*** My reading of UCITA is that it subjects independent developers to more risks and restrictions than current law, not fewer. My primary legal clients are independent authors, developers and testers. I have a stake in this. My reading is shared by two past presidents of the Independent Computer Consultants Association (ICCA), and by several other senior members of the software community.

Now, let's look at some specifics. All references to UCITA are to the July draft, at www.law.upenn.edu/bll/ulc/ulc.htm. Prior to this draft, UCITA was called "Article 2B" and was a proposed amendment to the Uniform Commercial Code.

Mass-Market Contracting

I start with the concept of mass-market contracts because it is basic to several other points below. According to The Response:

"Current law distinguishes between consumer and business use, treating small businesses the same as corporate behemoths. UCITA adopts a new "mass market" concept, extending consumer-like protections to small business for the first time."

The law governing sales of packaged software today is UCC Article 2. Article 2 has no special provisions for consumers. Under Article 2, customers are either merchants (specialists in transactions involving "goods of the kind") or non-merchants. Huge businesses who are not software experts are non-merchants in software contracts.

There is no distinction in UCITA between small businesses and corporate behemoths. In the UCITA meetings, we have often discussed the fact that Dow Chemical (actively represented in these meetings) will be able to take advantage of mass-market rules, such as they are.

By the way, when I say that someone "sells" software, I am speaking inclusively. The seller might be selling you a copy of the product or he might be selling you a license to use the product. There is a sale in either case, and Article 2 (law of sales of goods) has been handling transactions of both kinds for 30 years.

UCITA excludes several types of sales of packaged software (no customization) from the definition of "mass-market" (UCITA 102(a)(46)). A sale involving business or professional software use (even a tiny, one-person, home-based business) are mass-market only if it is "directed to the general public as a whole including consumers." Many small business programs, such as any kind of vertical package (such as software to run a dentist's office) will not qualify. Also, the software must be sold in a "retail transaction under terms and in a quantity consistent with an ordinary transaction in a retail market." A sale involving more than one copy of the product might not qualify and I'm not sure whether the Net is a qualifying "retail market." (Previous drafts of UCITA were roundly criticized for explicitly disqualifying sales on the Net, so now that draft has been revised and is simply ambiguous.) If there is a volume discount, the sale does not qualify because the terms have to be "substantially the same" as a consumer (one-copy) sale. A site license disqualifies the sale from being "mass-market." Many small business (and large business) purchases of off-the-shelf, packaged software will not be "mass-market."

As to the new consumer-like protections, there are none (compared to current law). Early drafts of

UCITA extended many rights to mass-market customers. But those were whittled away over the years. What's left is the original intent (accurately described by Brennan and Barber) and a series of special rules that preserve for mass-market customers rights that currently are available to all customers. Here is a complete list:

- A mass-market "agreement" to apply UCITA to a contract involving a "tangible copy of information in print form" (like a book) cannot alter laws specifically applicable to books. (These rules currently apply to all contracts. UCITA is saying that non-mass-market contracts can contract around these rules.) (UCITA 103(e)(1))
- A mass-market "agreement" to take a software transaction out of UCITA "does not alter the applicability of unconscionability, fundamental public policy, or good faith." (These rules are supposed to apply to all contracts. UCITA is saying that non-mass-market contracts can contract around these rules.) (UCITA 103(e)(2))
- A mass-market customer has a right to a refund if the seller doesn't make the terms available until after the sale and the customer doesn't agree to them. Under current law, if the new terms are significant ("material"), the customer-favorable court decisions say that customers can reject the new terms and keep the product anyway while some other courts say that a customer who rejects the terms must return the product for a refund. UCITA rejects the first rule (see below) and applies the second rule only to mass-market customers. The much-vaunted "right of return" is not only not new. It is narrowed so that many businesses will no longer will be able to take advantage of it. (UCITA 112(b), 211(b), 211(c))
- Suppose that a customer specifically negotiates a contract with a software publisher. On installing the software, he encounters a click-wrap license. He must click "OK" to install the software. He does. Under current law, the negotiated agreement prevails over the click-wrap (*Morgan Laboratories v. Micro Data Base Systems, Inc.*, Case Number 96-3988 THE, 1997, U.S. District Court, Northern District of California, Chief Judge Thelton Henderson). Under UCITA (211(a)(2)), the negotiated agreement prevails only for mass-market sales.
- UCITA (304) says that a mass-market customer can terminate a continuing contract (such as an annual maintenance contract) if the contract says the seller can modify the terms and the seller changes a significant term in an unacceptable way. I think that large customers have a good argument to cancel such a contract under current law. They lose it under UCITA.
- UCITA (406(b)) says that a warranty disclaimer in a mass-market contract must be conspicuous. First, under Article 2, it is very hard to argue that a term that cannot be seen until after the sale is conspicuous, so all customers lose some ground on this issue. Additionally, under Article 2, the disclaimer must be conspicuous no matter who the customer is. UCITA takes this away from non-mass-market customers.
- UCITA (704(b)) is the "perfect tender rule," which allows the customer to reject a product if, on a quick inspection, the customer discovers a nonconformity between the product and the contract. The perfect tender rule is a longstanding, basic rule of contract law. The Response says that the perfect tender rule "makes no sense for custom software." Yes, sales law generally makes exceptions for custom work. But what about all those non-customized software products that are not mass-market? Complex products (like airplanes and nuclear reactor turbines) are subject to the perfect tender rule today. UCITA takes this right away from all non-mass-market software customers.

Customers who are large enough will be able to negotiate the terms of their contracts. These changes are a problem for large customers because they change the baseline, the starting point for

negotiations. This is why the Society for Information Management (which represents large software and hardware customers) oppose UCITA. But the people who are especially hurt by these changes are small businesses because they lack the bargaining power to get back all of the rights they are losing under UCITA.

Privacy Protection?

The Response says,

"What if a vendor uses your private information against your wishes, such as putting your name and address on a mailing list? Sales law will not allow consumers to make a contract that forces the vendor to take it down. UCITA will."

In 1996, there was quite a debate about privacy protection in UCITA. See my paper, "Privacy Problems in Article 2B" at www.badsoftware.com/privacy.htm, which was part of the debate. Ultimately, the drafting committee decided to leave the privacy issue alone, not to extend or limit privacy rights.

A specific term in a contract that protects a customer's privacy rights would certainly be enforceable under current law, as it would be under UCITA.

Of course, you might have a challenge getting meaningful privacy protections written into your contract, a contract written by the seller and presented to you on a take-it-or-leave-it basis. Contrast this with the privacy directive in the European Union, which gives you the right to know that a business is collecting data about you, the right to see the data, and the right to require the business to limit its distribution of that data. You don't get rights like these under UCITA.

Click-On Contracts and "Expanded" Return Rights

You can signify assent to a contract by conduct under current law, and that conduct can include clicking on a computer screen. Click-on contracts are already being enforced by the courts. There are some circumstances under which it is not clear if a click is legally equivalent to a signature, but these are well taken care of by the Uniform Electronic Transactions Act, which will almost certainly be passed by NCCUSL in July, 1999. We don't need UCITA for this.

UCITA couples something else with click-on contracting--*post-sale presentation of terms*. This is the point of contention and it is the most astonishing claim made by proponents of UCITA. They say that they have created a new customer right, the "*right of return*" in the process of making post-sale terms enforceable. Not so.

Current law is split in dealing with contracts that you don't receive until after you pay for the product and take it away.

One approach treats the contract as having been made when the customer paid for the product and the seller shipped it. The terms in the box (or embedded in the software) are proposals for modification of the contract. The customer can reject "material" modifications and keep the product. *Arizona Retail Systems v. The Software Link* (*Federal Supplement*, vol. 831, p. 759, 1993, U.S. District Court, Arizona) and *Step-Saver Data Systems v. Wyse Technology* (*Federal Reporter*, 2nd Series, vol. 939, p. 91, 1991, U.S. Court of Appeal, 3rd Circuit). are software cases that make this point. I researched this issue in detail and summarized my findings in *Bad Software* (Kaner & Pels, 1998, see pages 190-218, 316-318). In my view (and in the view of many other legal scholars), this is the dominant approach within Article 2 of the Uniform Commercial Code.

Early discussions of UCITA were forthright about the intention to change the law. For example, the Reporter's Notes to the April, 1996 draft of Section 308 said, "This section reverses *Wyse Technology v. Step-Saver*."

A few recent cases have said that these terms are enforceable even though they are presented after the customer pays and takes delivery. *ProCD v. Zeidenberg* (*Federal Reporter*, 3rd Series, vol. 86, p. 1447, 1996, U.S. Court of Appeal, 7th Circuit) and *Hill v. Gateway 2000* (*Federal Reporter*, 3rd Series, vol. 105, p. 1147, 1997, U.S. Court of Appeal, 7th Circuit) are the two primary cases. Cases in this line say that these contracts are enforceable *because* the customer has the right to accept the contract and keep the product or reject the contract and send the product back for a refund.

UCITA adopts this second approach. It is not new and customer advocates consider it unfavorable. The drafting committee that is working on the latest round of revisions to Article 2 has refused to adopt it.

Additionally, as noted above, UCITA grants the right of return only to mass-market customers. Under UCITA, if the publisher puts a notice on the box that says "License terms inside" the non-mass-market customer who sees the terms after the sale and doesn't agree to them has no right to return the product for a refund.

Enhanced Choice?

The Response says that it is important to preserve choice in support options. I agree. But to make an intelligent choice, you need some information. Like, what is the support policy of the software vendor? Is there a warranty? How long? Does the vendor charge for support? Starting when? How much? How do you choose between support options if you can't find out what option you've chosen until after you pay for a product and accept delivery?

Electronic commerce should make it especially easy for sellers to make contract terms available to customers. When the customer is ordering a product, the seller can display a hyperlink to the contract for that product. Customers who want to do comparison shopping will click the link to read the terms.

Software publishers argued against this proposal, and the UCITA drafting committee chose not to adopt it.

Increased Developer Risk?

The Response says that we "want to . . . impose greater risks on developers who make available innovative and alternative programs such as Java or Linux." Not so.

The business model of the free software movement is a service contract model. Give the customer the software for free and sell development, installation, and support services. Under current law, those services carry a warranty of "workmanlike performance" (you promise to do good work).

Article 2 doesn't cover service contracts. UCITA does. Like Article 2, UCITA writes a warranty of merchantability into every contract and a warranty of fitness for use into many contracts. The warranty of merchantability promises that the product is reasonably fit for ordinary use. The warranty of fitness promises that the product is specifically fit for this customer's stated use.

If you act as a consultant for someone, and provide and install Linux on their computer, then under UCITA, you are accountable to the customer for all of the defects in Linux and you will be liable for all of the customer's incidental expenses and consequential losses, unless your contract specifically disclaims the warranties and excludes the damages. Products like Linux come with warranty disclaimers, but those protect the developers of Linux, not you. Under UCITA, you have

to provide your own disclaimer or you became a guarantor on your own.

Publishers don't mind UCITA's warranty rules because they can disclaim the warranties in contracts that they don't let the customer see until after she has paid for the product and accepted delivery. But service contracts for small amounts (\$25,000 or less) are typically handshake deals or short form contracts that name a service and a price. Long form contracts that disclaim liability create mistrust and often blow the deal or result in extensive (and expensive) negotiation.

Starting in 1996, I repeatedly proposed that sellers and service providers (publishers and consultants) should not have to reimburse customers for consequential losses (these are usually the biggest losses) that were caused by a defect that was unknown at time of sale/service or that was documented in a way that a reasonable customer could understand. This gives independents real protection, a way to manage risk that they don't have to negotiate for. The IEEE, ACM, ICCA and others have also proposed that sellers should be liable for consequential damages if but only if the product has a known, undocumented defect (or, in some proposals, a defect caused by gross negligence). The UCITA drafting committee has repeatedly rejected this.

In practice, UCITA's new rules will force independent developers to seek legal advice more often and will subject them to much more liability risk. I see UCITA as a threat to the free software movement.

Reverse Engineering

Another innovation-related issue is the contractual use restriction. This is a restriction, placed by the software publisher on your use of the product. A clause forbidding reverse engineering is such a restriction. UCITA makes contractual use restrictions broadly enforceable, so on its face, it allows publisher to enforce bans on reverse engineering.

Large publishers have argued in favor of enforcement of reverse engineering bans, arguing that reverse engineering creates opportunities for unfair competition. But we do reverse engineering to discover and work around defects, to develop interoperable products, to discover theft of copyrighted work, and for many other reasons that have nothing to do with supposedly unfair competition. (See my paper, "The Problem of Reverse Engineering" at www.badsoftware.com/reveng.htm for examples and references). The ACM, IEEE-USA, ICCA and many individual developers have protested against UCITA's apparent enforcement of a ban on reverse engineering.

Reverse engineering restrictions are fully enforceable in traditional, signed license contracts. But courts throw them out in the sale of mass market products. This includes cases involving computer software, such as *Sega Enterprises Ltd. v. Accolade, Inc.*, (1992), *Atari Games Corp. v. Nintendo of America, Inc.* (1992) and *Vault Corp. v. Quaid Software Ltd.* (1988).

The Response says that "UCITA expressly defers to preemptive federal law." Of course it does. Under the Supremacy Clause of the United States Constitution, a state law (like UCITA) is unenforceable when it is preempted by federal law. The Response also says that "the case law has addressed reverse engineering as a question of copyright infringement under federal law. Congress has already adopted specific rules on the issue, including in the 1998 Digital Millennium Copyright Act. UCITA leaves this issue to Congress where it belongs." Those of us who oppose UCITA wish that UCITA *would* leave reverse engineering to federal law, where it does belong.

The problem is that UCITA creates a new argument for enforceability of these clauses, by creating a mechanism for turning every software contract into a signed license contract. When you click "OK" while installing the software, you have "signed" a contract that declares itself to be a license, and which declares that you agree to terms which include fewer rights than you would obtain if you had bought a copy of the software under the Copyright Act.

So, does this take a mass-market software product outside of the scope of cases like *Sega* and *Vault* or not?

There have certainly been attempts in the UCITA drafting process to clarify this.

Two years ago, the American Law Institute (which was then a co-authoring organization for UCITA) passed a resolution that contractual use restrictions (such as those on reverse engineering) should be treated the same way whether the product is sold or licensed under a mass-market license. The UCITA drafting committee rejected it

Then, last year, NCCUSL passed the following amendment to UCITA:

"If a court as a matter of law finds the contract or any term of the contract to have been unconscionable or contrary to public policies relating to innovation, competition, and free expression at the time it was made, the court may refuse to enforce the contract or it may enforce the remainder of the contract without the impermissible term as to avoid any unconscionable or otherwise impermissible result."

That would have given the courts plenty of room to protect legitimate reverse engineering activity while protecting trade secret or other intellectual property rights. Unfortunately, the UCITA drafting committee rewrote this (Section 105(b)) as follows:

"If a term of a contract violates a fundamental public policy, the court may refuse to enforce the contract . . . or so limit the application of the impermissible term as to avoid any result contrary to public policy, in each case, to the extent that the interest in enforcement is clearly outweighed by a public policy against enforcement of the term."

Note that the specific protection of "innovation, competition, and free expression" is gone. Also, the boldfacing highlights issues that will have to be proved in court before a judge can throw out a term under UCITA. My sense (and the sense of some law professors who teach courses on contract law) is that, compared to current law, this restricts a judge's power to throw out terms against public policy.

No one knows whether a UCITA-based ban on reverse engineering in a mass-market software product will be enforceable or not. I think (or at least hope) that these restrictions will eventually be tossed out, but it will take a lot of expensive lawsuits to settle the matter. In the meantime, independent developers are at risk, and people who can't afford to fight a lawsuit will be advised (as I will advise my clients) that a clause banning reverse engineering appears enforceable under UCITA and should therefore be taken seriously.

Conclusion

I'd like a law that unifies software contracting too, but as I read it, UCITA poses a threat to our customers, to the standards of our profession, and to the long term health of the industry. That's why I've asked that UCITA be rejected, leaving current law intact.