

**CEM KANER, J.D., Ph.D.**

Law Office of Cem Kaner  
P.O. Box 1200  
Santa Clara, CA 95052

[www.badsoftware.com](http://www.badsoftware.com)

[kaner@kaner.com](mailto:kaner@kaner.com)  
408-244-7000 (Voice)  
408-244-2181 (Fax)

**ARTICLE 2B and REVERSE ENGINEERING**

Cem Kaner. July 1998.

Article 2B validates use restrictions in software licenses, including mass-market licenses. One of the common use restrictions is a ban on reverse engineering, and 2B makes it presumptively enforceable (enforceable unless federal law preempts the restriction). Before you say, "Yes, yes, federal law will preempt this," let me remind you of Ray Nimmer's recent analysis of federal preemption of contract clauses:

**There have been no cases in which [Copyright Act] Section 301 preemption was used successfully to challenge and invalidate a term of a contract that was enforceable as a matter of general state contract law.**

(Ray Nimmer, "Breaking Barriers: The Relation Between Contract And Intellectual Property Law" *Conference on the Impact of Article 2B*, Berkeley, April 23-25, 1998.  
[www.SoftwareIndustry.org/issues/guide/docs/rncontract-new.html](http://www.SoftwareIndustry.org/issues/guide/docs/rncontract-new.html).)

Therefore, the place to consider whether reverse engineering should be allowed in the mass market (via restriction on contract terms) or banned in the mass market (via letting publishers enforce restrictive terms) appears to be Article 2B.

***Here are four examples of use of reverse engineering. All seem socially desirable to me. How do we avoid killing off these types of uses under 2B?***

1. **Education**: I learned Operating System programming by disassembling the Apple II OS. (*See charts 2-3.*)
2. **Investigation of Potential Infringement**: When I was Testing Technology Team Leader at WordStar, we reverse engineered a competitor's product in order to determine whether they had used some of our code in their product. (*Sorry, no charts are available.*)
3. **Interoperability**: When I was the Software Development Manager for Express Publisher, a desktop publishing program, we had to reverse engineer other companies' file formats in order to be able to read (and thus format) their data. (*See chart 4.*)
4. **Investigation of Consumer Fraud**: Part of the basis for claims that the capabilities of SoftRAM95 had been misrepresented in the mass-market came from reverse engineering. (*See charts 5-7.*)

## **ARTICLE 2B and REVERSE ENGINEERING**

### **Chart 2: DISASSEMBLY OF THE APPLE II O/S**

Some people say that reverse engineering is so difficult that no one really wants to (should want to) (has time to) do it. Therefore, they say, banning it is no big deal.

Reverse engineering is time consuming, painstaking, but informative work. But, just as lawyers learn much from examining briefs written by other lawyers, many programmers learn new skills by studying other peoples' programs. That's what I was up to.

The three pages on this chart are examples of the kinds of work that John Vokey and I did to analyze the Apple II Operating System in 1981.

- One page is from a tracking table that showed us how special locations in memory were used by parts of the operating system. (At this time, we only knew those parts by number. Over time, once we understood them, we assigned them names.)
- The next page shows a disassembled segment of memory and our notes about what parts of the program are stored in this segment. Once we understood the function well, we typed up the notes (see the final page).
- The third page shows our almost-completed analyses of two of the Operating System's subroutines.

**Based on these analyses, we made several enhancements to the system. Examples:**

- We discovered a serious error in the system's random number generator (RNG). We developed and published an alternative RNG that was suitable for the mathematical simulations that John Lyons and I were conducting.
- We extended the built-in programming language, to make it easier for different laboratories to create standard functions and share them. (We published the extensions in a few articles in computer magazines.)
- I rewrote or extended device control routines so that we could run certain types of experiments with total timing errors of less than 60 microseconds. These routines made my dissertation research (timing-related experiments) possible and were used by other doctoral students for their research.

**I couldn't have done this work under Article 2B.**

**CEM KANER, J.D., Ph.D.**

Law Office of Cem Kaner  
P.O. Box 1200  
Santa Clara, CA 95052

[www.badsoftware.com](http://www.badsoftware.com)

[kaner@kaner.com](mailto:kaner@kaner.com)  
408-244-7000 (Voice)  
408-244-2181 (Fax)

## ARTICLE 2B and REVERSE ENGINEERING

### Chart 3: INTEROPERABILITY

In 1990-91, I was the Software Development Manager for several releases of a desktop publishing program called *Express Publisher*. *Express Publisher* reads files that you created with a word processing program and helps you lay out the files (format them in pretty ways).

*Express Publisher* must be able to understand your files. A memo that you write with Microsoft Word is stored in a different file from the same memo, written in WordPerfect. *How can Express Publisher know how to read each of these program's data files?*

Initially, I called several software publishers and asked them for their file format specifications.

- One publisher sold us a set of documentation for about \$70.
- A second publisher said that they didn't have a full specification, but we were welcome to try to figure it out (by looking at how lots of different memos were stored -- ***this is reverse engineering***).
- One publisher told me that we could *not* (not, ***NOT, NOT!***) reverse engineer its file format because we were competitors and they didn't want us to be able to import their files into our program. ***They wanted to prevent us from doing reverse engineering for the purpose of making our program interoperable with (able to work with) their program.***

Having bought copies of each program in the mass market, we decided that we had the right to reverse engineer their file formats whether the publishers of those programs liked it or not. After a lot of work, we were able to read the main types of files that our customers would expect of us.

Eventually, *Express Publisher* became the best seller in a crowded niche (under \$100, DOS-based desktop publishers).

We wouldn't have had this success -- or any success -- if we hadn't been able to reverse engineer the file formats. No one would want to retype their documents just to work with our program.

As a general principle, no new product can enter an existing category unless it can read data files created by its leading competitors. If there is no ability to read, the cost of converting from the old format to the new will be much too high to risk on a newcomer that isn't well established. And few newcomers to the market will be able to read their competitors' files without reverse engineering the competitors' file formats.

In our case, we were not competitors with the word processing companies, but we still could not have succeeded in the market without reverse engineering other companies' data files.

*This was an example of reverse engineering to achieve interoperability. As far as I can tell, Article 2B will allow companies to ban this. What public policy is served by that ban?*

## **The Case of Synchronys SoftRAM95**

SoftRAM95 was a utility program that sold tremendously well when Windows 95 first came out. It was supposed to improve the computer's management of memory, but apparently it did nothing (or almost nothing).

Here are some articles:

- *SoftRAM 95 "False and Misleading"*--overview of the situation, with many links to other data and reports.
- PC Magazine reports that SoftRAM95 does not compress RAM under Windows 3.x either.
- Dataquest conducted a survey of SoftRAM95 users. ***80% of them were satisfied customers.*** This illustrates a key point. ***It is hard for non-experts to tell whether a program lives up to its claims.***
- Andrew Schulman reverse engineers SoftRAM
- Mark Russinovich reverse engineers SoftRAM
- Synchronys warns a magazine about an upcoming review, claiming rights under trade secrets and copyright laws. ***Under Article 2B, these claims would have had power if the analysis of SoftRAM was based on reverse engineering and the SoftRAM product came with a no-reverse-engineering license.***
- The FTC files a complaint (eventually settled--SoftRAM goes off the market.)

*This example of reverse engineering and disassembly was done to detect consumer fraud. What public policy is served by banning or discouraging this type of reverse engineering?*