



*"The Perfect Fit for
Your Business"™*

ROBERT H. CARPENTER, JR.

ATTORNEY AT LAW
5912 CASTLEBAR LANE
PLANO, TEXAS 75093

—
TELEPHONE 972.473.4834

—
email: Bob.Carpenter@CarpenterLaw.net

—
www.CarpenterLaw.net

April 2005

SECRETS OF THE CODE

Source code is a constant cause of concern, and sometimes controversy, when parties are negotiating licensed technology agreements. Purchasers/users (licensees) sometimes get source code; they often request it. Vendors, or licensors, in some circumstances must deliver source code. Source code is the symbolic code in its original form that a computer translates into the object, or executable, code (*i.e.*, the zeros and ones, or binary code, that a computer can read).

Some licensors deliver source code to licensees because their applications run on operating systems that compile, or translate, source code into object code differently based upon the hardware being employed and the installed version of the operating system. While it would technically be possible to compile source code based on each licensee's hardware and operating system requirements, the burden of doing so for the numerous possible combinations and for each new release could be overwhelming to licensor and licensee. Time to market would be unnecessarily lengthened.

Why do licensees want source code? Most often their requests originate from business continuity concerns – that is, if something happens to the licensor or its support of the licensed technology, then the licensee wants to be able to continue operating the software product and updating it. Also, some licensees believe that, if they have the source code, they are free to customize the software application to run uniquely in their environment.

Proprietary vs. Open Source

Access to and use of source code is typically a function of the software vendor's business model. There are basically two – proprietary and open source. A financial institution, for example, is pretty much locked into the world of proprietary software for its data processing. A bit of background here will give context to this discussion.

Under a proprietary business model, the owner of the software application, an intellectual property asset (IP), generates revenues not only from license fees, but also from maintenance (*e.g.*, problem fixes, program improvements and new features and functionality) and custom code authoring. The user relies upon the developer's economic incentive to produce and maintain a robust, efficient and

innovative product. This contrasts with the business model of open source software (OSS) providers like Red Hat, Inc. that rely upon the efforts of the world-at-large to continue the evolution of OSS and that focus their internal revenue-generating efforts on product differentiation and feature/functionality enrichment.¹

Both environments impose obligations upon licensees. Sometimes users confuse the two. This is particularly true when a licensee obtains access to source code for its licensed technology. The OSS license not only allows, but also *encourages* licensees to evaluate source code and develop and distribute improvements. The proprietary licensor cannot do this.

Knowing The “Secrets”

A licensee’s possession of source code is the proverbial two-edged sword. “Open” access to source code, like any proprietary information, has pluses and minuses.²

Advantages

Actual possession of source code does, in some cases, guarantee business continuity in the event that the licensor ceases business or fails to maintain the licensed technology.³ However, a well-drafted and monitored escrow agreement can do as good a job, maybe even a better one, of protecting the licensee.⁴ An individual licensee may not receive all new releases or “builds” of a software application that are made available to another licensee, or may not receive them on the same schedule as others. When a master escrow agreement is used and the licensor is under an obligation to all licensees to update the escrow on the same schedule, new releases are certain to be available and administrative verification is more efficient and effective.

Of course, many licensees believe that the real advantage of having source code – sometimes unwittingly and incorrectly touted by the vendor’s sales executive – is the ability to perform custom coding or authoring. While some software allows customization through special configurations, custom authoring is a different matter. It really involves re-writing the software, or at least part of it, to suit specific requirements that cannot be handled by the original programming.

Risky Business

Licensors, particularly in the proprietary business model, consider their software an asset. the value of which must be preserved. One does this through a series of protective devices that declare ownership and limit use. These mechanisms assert control over licensed technology even in the hands of a licensee.

The OSS model suggests one potential risk from source code modifications that are made with either express or implicit approval of the licensor. OSS licenses typically provide that, if a user incorporates elements of OSS into another program and distributes that derivative work, the person

¹ Red Hat, Inc., Why Open Source?, <http://www.redhat.com/about/mission/opensource.html> (last visited Sep. 25, 2007).

² This “open” access describes distribution of an application’s source code to licensed-for-a-fee users, rather than the universal availability of source code under an OSS license.

³ This advantage does not obtain for software that runs on a mainframe in a complex operating system, but many applications do run on client servers that are affordable to a licensee and that can be locally supported.

⁴ Companies like Iron Mountain (<http://www.ironmountain.com/ipm/escrow/>) have excellent escrow arrangements that protect the licensee without risk to the licensor.

receiving the derivative work must get all the rights to the whole program that the distributor possessed.⁵ The unwitting inclusion of OSS by a licensee in modified licensed technology and its subsequent distribution could jeopardize the proprietary rights of the licensor.

Finally, some experts suggest that the widespread distribution of source code can facilitate hacking by making it easier for hackers to locate and exploit “back doors.” Some source code has been available to licensees for over 30 years. Hundreds of programmers have had access to it. Although considered mature programs with time-tested stability, some of these are written in “dead” programming languages that do not have the robust security features of the newer ones.

Licensees of such proprietary software must assume this security risk. (The licensor probably has largely, if not entirely, limited its liability for damages caused by security breaches.) Such software in an OSS environment would be receiving constant peer review to remedy security defects. In a proprietary environment, only the licensor is “minding the store.”

Limited License

In a proprietary environment, a licensee should expect its licensor to be aware of and to provide for reasonable protections for the IP in which the licensor has a substantial investment. The grant language in a licensed technology agreement might read like this:

Licensor grants to Licensee, subject to the terms and conditions in this Agreement and any others set forth in a Licensed Product schedule to this Agreement, a term, nonexclusive, nontransferable, personal and internal use license to use the Licensed Program(s) identified in the product schedule only for the internal operations of Licensee and for the processing of its own data. The Licensed Program(s) include the program code (as further identified on the product schedule), documentation and training materials embodying or related to such program code and any subsequent versions or releases of the Licensed Program(s) that may be delivered to Licensee and all copies of any of the foregoing.

This license probably continues,

Licensee will not make or permit others to make copies of or reproduce any part of the Licensed Program in any form without Licensor’s prior written consent. In no event will Licensee decompile, disassemble or otherwise reverse engineer or modify the Licensed Program. If Licensee makes any changes to the code of the Licensed Program (as opposed to mere changes in user configuration that the Licensed Program may allow), Licensee agrees that such changes and modifications will be Licensor’s property, unless Licensor has given its prior written consent to the contrary; and Licensee will assign, and does hereby assign, to Licensor all right, title and interest in and to such changes and modifications.

I always thought Jimmy Buffett said it first:

*But the right word at the right time
May get me a little hug
That’s the difference between lightning
And a harmless lightning bug*

– Jimmy Buffett & Ann Lee,
“Fruitcakes” (1994)

Turns out, it was really Mark Twain:

*The difference between the almost right
word & the right word is really a large
matter—it’s the difference between the
lightning bug and the lightning.*

– Samuel L. Clemens,
Letter to George Bainton (1888)

Next month’s newsletter will address the use of “Plain English” in legal agreements. The right word at the right time gets a clear, concise and complete agreement among the parties.

⁵ See, e.g., Free Software Found., Inc., GNU Gen. Pub. License, Version 2 (Jun. 1991), <http://www.redhat.com/docs/books/max-rpm/max-rpm-html/s1-rpm-resources-gpl.html>.

License Dimensions

These provisions describe a limited license: One that has dimensions of time, place and use. The limited license is the lynchpin to the proprietary technology business model. To protect its valuable IP asset, the licensor must assert positive control over the licensed program and insist that attempts to “pirate” the program are contrary to the terms of the license.

Reasonable Accommodation

Of course, a licensee may have legitimate need to alter a program, and such changes may be so unique to the user that the licensor would not include them in a new release and would require so large an investment to implement that it would not be economical for either party. The license should provide that, in such a case, a licensee, or its pre-approved independent contractor, may change source code after notice and approval by the licensor. Further, the license will likely include the additional asset protection device that a copy of the revised source code will be delivered to the licensor. (Pursuant to the sample agreement language above, these code modifications would then become the property of the licensor.)

“Pain” Relief

Being “stuck” in the world of proprietary software simply means that a licensee must carefully evaluate how it plans to use its licensed technology and ensure that the licensed technology agreement permits the contemplated use.

Further, one should not forget the time and place dimensions of a license. Re-licensing fees can increase the cost; and these costs can be related to both time and the physical location and processing capacity of the CPU on which the program is installed.

As in any business arrangement, thinking ahead can save headaches!

ARBITRATION EPILOGUE

The March 2005 newsletter presented a “dark side” of arbitration and suggested a mediation alternative that provides a viable alternative dispute resolution (ADR) mechanism that preserves the elements that distinguish the U.S. legal system. Knowing the rules, regardless of which ADR mechanism one selects, is the most important thing. Enter the “London Arbitration Clause.” Used now in some insuring agreements, this mandatory provision presents an even more aggressive assault upon transparency, fairness and equality in dispute resolution by changing the prism through which opponents view their dispute to that of a different culture – Great Britain and its very different legal system. This London provision gives new meaning to the “know your rules” caution.

Other Recent Client Newsletters

“The Price Escalator Clause: *A Masked Bandit?*”, the February 2005 newsletter, discusses the behavior of pricing indices and how they may overstate inflation and, thus, vendor cost increases.

ePayments and how they challenge the new regulatory environment is the topic of the January 2005 newsletter “ePayments Put the Paper Check ‘Out to Pasture’.”

Client newsletters are available at <http://www.carpenterlaw.net/newsletters/archivednewsletters.html>.

© 2005, 2007 Robert H. Carpenter, Jr.

This client newsletter is for informational purposes only, and is not intended to be legal advice. Transmission of this newsletter is not intended to create, and its receipt does not establish, an attorney-client relationship. Legal advice of any nature should be sought from legal counsel.

IRS CIRCULAR 230 DISCLOSURE: Notice regarding federal tax matters: Internal Revenue Service Circular 230 requires us to state herein that any federal tax advice set forth in this communication (1) is not intended or written to be used, and cannot be used, for the purpose of avoiding penalties that may be imposed by federal tax laws, and (2) cannot be used in promoting, marketing, or recommending to another person any transaction or matter addressed herein.