

COPYLEFT HANGING:

ARE LICENSEES UNDER THE GPL THIRD PARTY BENEFICIARIES?

By: Paul H. Arne and John W. Martin IV^{1,2,3}

Introduction

What is the difference between a “license” and an “agreement”? For many in the open source community, this is a critical distinction. Some of the most frequently used open source software licenses intend that they be treated as a license to intellectual property and not an agreement.

The concept of terms and conditions⁴ that accompany open source software being a license, rather than an agreement, is an unconventional concept to most lawyers. Not surprisingly, the initial treatment of this issue by courts has been inconsistent.

This article examines the following:

- The basic conceptual differences between a license and an agreement;
- the history and rationale for treating open source terms and conditions as licenses and not agreements;
- the enforcement implications of treating open source terms and conditions as licenses, rather than agreements;
- the background and ruling in *Jacobsen v. Katzer*, related to enforcing open source terms and conditions as licenses and not agreements;
- the background and ruling in *Versata Software v. Ameriprise Financial*; and
- the different risk posture between treating open source terms and conditions as license terms or agreements.

Fact Pattern

This article uses a simple fact pattern for reference as shown in Figure 1 below:

¹ Copyright © Paul H. Arne and John W. Martin IV, 2016. All rights reserved.

² Paul is the chair of the Technology Transactions Group of Morris, Manning & Martin, L.L.P. John is an associate in the Corporate Group of Morris, Manning & Martin, L.L.P.

³ This article does not create an attorney/client relationship with you and does not provide specific legal advice to you or your company. Certain legal concepts have not been fully developed and certain legal issues have been stated as fact for which arguments can be made to the contrary, due to space constraints. It is provided for educational purposes only.

⁴ In this article, there is a need to discuss terms and conditions that describe what someone is required to do or not do, without reference to whether they are terms of a license or terms of an agreement. When the intent is to be agnostic as to license or agreement, “terms and conditions” is used.

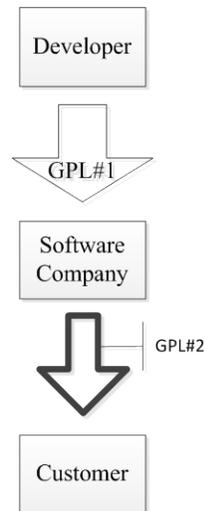


Figure 1: Licensing Fact Pattern

A “developer” creates some open source software and makes it available to the public using the GNU General Public License (“GPL”)⁵. This first use of GPL will be called “GPL#1.”

A “software company” then obtains the open source software and develops the “company’s product,” which is a derivative work of the open source software. The GPL provides that if the software company creates any derivative work of the open source software and distributes that software to a customer, then the software company must license the derivative work also under the terms of the GPL. The software company in turn licenses the company’s product to a “customer” under the terms of the GPL (“GPL#2”; the terms of GPL#1 and GPL#2 are identical) but does not include source code. If asked, the software company would refuse to provide source code to the customer.

The GPL was chosen as the license for three reasons:

- it is the most widely-used open source license;
- it contains reciprocal—or “copyleft”—provisions that are important to the analysis;⁶ and
- it is the applicable license in *Versata v. Ameriprise*.

⁵ While there are multiple versions of the GPL, the version does not make a difference in this analysis.

⁶ See *infra* note 14.

Enforcing Open Source Terms and Conditions

Using an Agreement

Imagine for a moment that a client comes to you and asks you to prepare an agreement. The client needs the agreement to be enforceable in virtually every country in the world without modification. The agreement must be binding whether or not a party physically signs, clicks “I agree,” or takes any other overt action that unequivocally manifests an intention to be bound by those terms. Is it possible to create such an agreement?

This is precisely the issue faced by many in the open source community when they seek to require compliance with certain terms and conditions in connection with the use of open source software. Generally speaking, open source software is distributed freely without regard to national borders. A developer’s challenge is to get software companies to comply with the developer’s terms and conditions regardless of location.

Using a License

Contrast the creation of a world-wide, enforceable contract with buying a book at a bookstore. Suppose that an author is a U.S. citizen who writes a book in the U.S., creating a book that is copyrighted under U.S. law. If a purchaser buys the book at retail in the U.S., the purchaser does not have the right to make multiple copies of the book and distribute those copies. The U.S. Copyright Act grants certain exclusive rights to the holder of a copyright, which includes the rights to make copies and distribute the copies.⁷ Because the Copyright Act grants these exclusive rights to the copyright holder, it is illegal for the purchaser of a book to make copies.⁸

The purchaser of the book didn’t have to enter into any agreement with the author or anyone else in order to be restricted from making or distributing copies of the book. Rights in copyright exist by statute and are not dependent on the existence of a contract for enforcement.

Worldwide Enforceability?

Take the same facts as above but change one fact. Rather than the book being purchased in the U.S., suppose instead that it is purchased in a different country. Can the purchaser of a U.S. copyrighted book, legally make copies and distribute the copies outside the U.S.? In most countries, the answer is still “no.”

The vast majority of all countries in the world are signatories to the Berne Convention (“Berne”) — a treaty that establishes a set of minimum requirements for each member countries’ copyright law. Berne was first adopted in 1886 to provide minimum protections for creators of literary and artistic works around the world.⁹ Approximately 172 countries have adopted Berne, giving it nearly world-wide force.¹⁰

⁷ 17 U.S.C. §106 (2011).

⁸ The “first sale doctrine” is ignored for the purposes of this article.

⁹ *Berne Convention for the Protection of Literary and Artistic Works*, WORLD INTELLECTUAL PROPERTY ORGANIZATION, <http://www.wipo.int/treaties/en/ip/berne/> (last visited Nov. 20, 2016). Berne protects the exclusive rights of copyright holders to reproduce their works (Article 9) and to make alterations to their works (Article 12).

Berne also establishes the concept of “local treatment.” Each member country agrees to enforce the copyrights of another member country as if the copyright was created under the first member country’s laws. Therefore, the holder of a U.S. copyright can go to another member country and enforce its copyright in that other country as if the U.S. copyright were instead the copyright of that other member country.

Copyright law may therefore provide a solution to the dilemma of getting terms and conditions enforced in most countries without having to comply with each country’s laws regarding contract formation, at least with respect to works that are subject to copyright protection. The idea is to provide software under a license, rather than an agreement.

The Rise of Open Source

Richard Stallman is a fierce proponent of programmers’ “rights.” If a programmer has access to software, whether in source code or executable code form, Stallman believes that the programmer should have the right to receive the source code, copy it, modify it, and distribute it in any way the programmer sees fit.

This principle only works as long as the software company cannot prevent the customer from doing the same things. Stallman recognized that simply dedicating software to the public domain did not propagate the “free” nature of software.¹¹ By simply releasing a work to the public domain, a developer surrenders all rights to the work and relinquishes any control over how it may be used by subsequent programmers.¹² Anyone could freely take, use, modify or incorporate the work and license it in any way they wish, effectively removing it from the public domain and defeating the aims of the open source movement.¹³

Stallman decided to use the exclusive rights in copyright law to help developers maintain the free nature of their products. Licensing software under the condition that the same open source license be extended to derivative works of that software allows developers to enforce the free nature of their software.¹⁴ By using a license, rather than an agreement, to accomplish this objective, Stallman was potentially able to achieve almost worldwide enforceability for the openness of software.

¹⁰ *Id.*

¹¹ V.K. Unni, *Fifty Years of Open Source Movement*, 40 S. ILL. L. J. 271, 279 (2016). This article offers an extensive history of the open source movement and an overview of some of the most common and influential open source licenses. Stallman’s vision of “free” is summed up in the policy statement incorporated into the GPL, “When we speak of free software, we are referring to freedom, not price.” *Id.*

¹² *Id.* at 282.

¹³ *Id.* at 282.

¹⁴ The use the exclusionary rights in copyright to enforce openness is often called “copyleft.” Those provisions of open source licenses that enforce openness are frequently known as the “reciprocal” provisions of these licenses. Some refer to these “reciprocal” provisions as “viral” provisions, including the court in *Versata v. Ameriprise*.

Relying on Berne arguably eliminates the difficulties in creating a single agreement that is enforceable everywhere in the world no matter the local requirements of contract law. Quoting Eben Moglen¹⁵:

[G]iven that we are all now existing in a Berne universe, the right way of internationalising is to come as close to Berne Convention mechanisms as possible. In order to do that the one step really needed is to get out of the licence dependence on any single legal system's vocabulary or unique conceptions.¹⁶

Berne and its standardization of copyright law allows developers to protect their work with one standard license,¹⁷ making the license form, and not the agreement form, very attractive for those who wish to broadly disseminate their work and prevent subsequent contributors from making the derivative works proprietary.

The developer essentially says, "I give you permission to use my code, if you comply with my conditions." Open source software makes the conditional permission (set forth in terms and conditions) available with the software. The license is triggered when the software company desires to do something with the open source software that requires a license under copyright law. If the software company complies with the conditions, then the software company is a licensed user of the open source software. If the software company doesn't comply with the conditions, then the software company is not a licensed user and therefore is an infringer.

The software company is not required to "agree" to anything. The software company can agree or not. The software company can follow the license or not. However, if the software company doesn't follow the conditions of the license grant, then the software company is subject to an action for copyright infringement by the holder of the copyright.

What are the Developer's Obligations to the Software Company?

If, by granting a license, a developer is merely granting permission, subject to certain conditions, does the developer have any obligations to the software company? A license is in essence a covenant not to sue.¹⁸ In the absence of a contract between developer and software company, the developer would seem to have no more obligations to the software company than the author of a book would have to a person who buys the book at a bookstore.

The Free Software Foundation makes this clear in its FAQ:

¹⁵ Eben Moglen is a law professor at Columbia University. He was the principal lawyer in charge of developing version 3 of the open source software licenses of the Free Software Foundation (including the GPL), the organization founded by Richard Stallman. *Staff*, SOFTWARE FREEDOM LAW CENTER, <https://www.softwarefreedom.org/about/team/> (last visited Nov. 22, 2016).

¹⁶ Eben Moglen, Address at the Third International GPLv3 Conference (June 22, 2006) (transcript available at <http://www.fsfeurope.org/projects/gplv3/barcelona-moglen-transcript.en.html>).

¹⁷ Case law in the various countries makes this assertion premature, but that is the open source community's hope.

¹⁸ Pamela Jones, *The GPL Is a License, not a Contract*, LWN.NET, Dec. 3, 2003, <https://lwn.net/Articles/61292>.

Strictly speaking, the GPL is a license from the developer for others to use, distribute and change the program. The developer itself is not bound by it, so no matter what the developer does, this is not a “violation” of the GPL.¹⁹

What About the Customer? Who Can Enforce Openness?

If a customer is licensed to use the company’s product under the GPL#2, the rights of the customer vis à vis the software company would seem to be the same as those of the software company vis à vis the developer under GPL#1. GPL#1 and GPL#2 are the same license. The software company is not bound to the customer under GPL#2.

Now suppose that the software company does not provide the source code of the company’s product with the distribution, and the customer would like a copy of the source code.²⁰ The customer requests the source code, and the software company refuses. The customer would like to be able to enforce the terms of the license against the software company, but may not have legal standing to do so.

We have already established that the software company has no obligations to the customer under the GPL#2. However, the software company’s refusal to provide the source code means that the software company has failed to comply with one of the conditions of GPL#1, so the software company is an unlicensed user and an infringer. The developer, as the copyright holder, therefore has a claim against the software company for copyright infringement for the failure to deliver the source code. However, only the holder of the copyright, the developer, has the right to bring a claim for infringement.

In some situations, one can be a third party beneficiary of an agreement between two other parties. One way for the customer to exert his rights is to argue it is a third party beneficiary to GPL#1 between the developer and the software company. The question is whether the customer can be a beneficiary of a license, as opposed to an agreement, and therefore make a claim directly against the software company. While there are no decisions on point, allowing a third party, the customer, to bring an action would seem to defeat the legal principle that only the holder of a copyright can bring a claim for infringement.

If, however, the GPL was interpreted to be an agreement rather than a license, then the customer *might* have a claim against the software company under the GPL#1 as a third party beneficiary.

Courts could possibly interpret open source licenses in several ways: as licenses, as agreements, or perhaps as a hybrid of the two.²¹ Whether a customer has the legal right to force the software company

¹⁹ *Frequently Asked Questions about the GNU Licenses: Is the developer of a GPL-covered program bound by the GPL? Could the developer's actions ever be a violation of the GPL*, GNU.ORG, <https://www.gnu.org/licenses/gpl-faq.en.html#DeveloperViolate> (last visited Nov. 22, 2016).

²⁰ The GPL requires a software company to provide source code upon request, the source code is not required to be provided in the initial distribution. GNU General Public License, version 2, § 6, (available at <https://www.gnu.org/licenses/gpl-2.0.html>) [hereinafter GPLv2].

²¹ While this article does not address which terms and conditions of a typical software license agreement are interpreted by the courts as license conditions or covenants, most software license agreements are hybrids of both. *See infra* note 49.

to provide the source code can be a very important issue when advising clients on open source software matters.

Evaluating Risk in Open Source

Lawyers who provide legal services related to open source software will periodically be faced with a software company that has violated the open source software's terms and conditions. Sometimes the client will ask the lawyer what the consequences are of refusing to distribute source code when required to do so under an open source license.

A frequent fact scenario is the following:

- A software company has licensed certain software from a developer under the GPL and incorporated it into the company's product.
- The software company then licensed the company's product to a customer, which triggers the obligations to license the company's product under the GPL and to provide the source code of the company's product to the customer upon request, in accordance with the terms and conditions of the GPL.
- The software company does not want to provide, or even offer to provide, the source code to the customer.

If the GPL terms and conditions are legally treated as an agreement, then the customer may be able to argue that it is an intended third-party beneficiary of the GPL#1 between the developer and the software company. This would allow the customer — the entity that actually wants the source code — to legally assert a claim directly against the software company to receive the source code.

On the other hand, if the GPL terms and conditions are a license, then arguably only the developer, being the holder of the copyright, would have a claim against the software company. The claim would be for a copyright infringement, rather than a breach of contract. Any redress for the customer would be in the hands of the developer rather than the customer itself.

How Courts Have Handled License vs. Agreement in Open Source Licenses

*Jacobsen v. Katzer*²²

Jacobsen v. Katzer and Kamind Associates, Inc. (d/b/a KAM Industries) is significant because it was the first (and until *Versata*, only) case to decide whether terms and conditions of an open source license were in the nature of a license or an agreement.²³ Jacobsen and Katzer/Kamind were the developers of competing software applications used to program the decoder chips that control model trains. Jacobsen and others, through an open source group known as JMRI, developed software called

²² Unless otherwise indicated, the facts of *Jacobsen* are from the Federal Circuit opinion, *infra* note 23.

²³ 535 F.3d 1373 (Fed. Cir. 2008).

DecoderPro. DecoderPro was licensed under an open source license: the Artistic License.²⁴ Katzer/Kamind's software, Decoder Commander, was presumably developed and licensed under a proprietary license. Katzer obtained patents on certain inventions and then asserted that DecoderPro infringed. In response, Jacobsen filed suit, seeking a declaratory judgment that the patents were unenforceable or not applicable to DecoderPro.²⁵

Earlier, Katzer/Kamind had downloaded DecoderPro and incorporated portions of it into Decoder Commander.²⁶ Despite using some of the DecoderPro code in its product, Katzer/Kamind did not follow the requirements of the Artistic License. Among other things, Jacobsen alleged that Katzer/Kamind did not include (a) the authors' names, (b) copyright notices for JMRI, (c) references to the file that contains the text of the Artistic License, (d) an identification of JMRI or SourceForge (a web site on which much open source software is managed and made available) as the source of the code, and (e) a description of how the files had been changed by the user, all of which were required by the Artistic License.

As a result of these failures, Jacobsen also brought a claim for copyright infringement against Katzer/Kamind. Jacobsen sought an injunction against Katzer/Kamind, seeking to prevent Katzer/Kamind from using any part of DecoderPro. The propriety of granting an injunction was the issue presented on appeal.

At the time of the District Court opinion, irreparable harm could be presumed in a copyright infringement, while no such presumption existed in a breach of contract context. The question presented was whether the unmet conditions of the Artistic License were (i) conditions of the license grant itself (i.e., if the conditions were violated Katzer/Kamind's use of DecoderPro would be unlicensed and therefore a copyright infringement) or (ii) contractual covenants for which normal breach of contract principles would apply.

Decision, Appeal, Decision

The District Court denied Jacobsen's motion for preliminary injunction, holding that the violated covenants of the Artistic License were not limitations on the grant of the license itself, but instead may be a breach of contract.²⁷ This decision was appealed to the Federal Circuit. The Federal Circuit reversed, holding that the violated provisions were conditions to the authorized use of DecoderPro, the violation of which resulted Katzer/Kamind's use of DecoderPro being an unlicensed use. Therefore, such use was a copyright infringement.

The conditions of the Artistic License take us a little further than the rights to exclude in copyright law. They require certain affirmative actions on the part of the user, as opposed to requiring the user not to do certain things.²⁸ Among other things, as a condition of distribution, the user is required to describe how the files have been changed and identify JMRI or SourceForge as the author of the code.

²⁴ The text of the Artistic License can be found at <https://opensource.org/licenses/Artistic-1.0>.

²⁵ It is the patent aspects of this case that gave rise to the Federal Circuit's jurisdiction, even though the ruling did not address patent issues.

²⁶ Apparently, portions of DecoderPro had become somewhat of a standard in this industry.

²⁷ *Jacobsen v. Katzer*, No. C 06-01905; 2006 U.S. Dist. LEXIS 79030 (N.D. Cal.).

²⁸ See Artistic License *supra* note 24.

Can the requiring of an affirmative action, where the absence of such action in and of itself would not result in one of the exclusive rights in copyright being violated, be treated as a condition that, if violated, would create a copyright infringement?

The Federal Circuit held that these requirements are conditions to the grant of the copyright license. Therefore, any use of DecoderPro while not complying with these affirmative conditions is a copyright infringement. This decision makes clear that some affirmative acts can be required as a condition of the right to use a copyrighted work.

The Federal Circuit drew a distinction based on the language used in the Artistic License — between conditions and covenants. A “covenant” is language of agreement and thus claims and remedies are those found within contract law. A “condition” is language of license, and thus claims and remedies are those of copyright law. The *Jacobsen* court held that some affirmative acts required by the Artistic License, such as placing attribution and notices in further distributed copies, are conditions, not covenants.

Jacobsen therefore stands for the proposition that terms and conditions in an open source license that are not directly related to the exclusive rights in copyright can still be conditions of a license, the failure to comply with such conditions resulting in an infringement claim, rather than a breach of contract claim.

Many software licenses contain limitations or requirements related to the “use” of the software, despite “use” not being one of the exclusive rights granted under copyright law. Most practitioners would argue, and these authors believe that most courts would agree, that the failure to comply with a use limitation in a license agreement is the failure of a license condition, rather than the breach of a covenant — at least in most software licenses.

Versata v. Ameriprise

In essence, the District Court in *Versata Software, Inc. v. Ameriprise Financial, Inc.*²⁹ reached the opposite conclusion, although the elements of the license involved were different from those in *Jacobsen*. The *Versata* controversy originally stemmed from a dispute between Versata Software, Inc. (“Versata”) and Ameriprise Financial, Inc. (“Ameriprise”). Versata licensed software to Ameriprise under the terms of a Master License Agreement (“MLA”).³⁰ As part of the arrangement, Versata authorized a third party contractor, Infosys Technologies, Ltd (“Infosys”) to make modifications to Versata code on behalf of Ameriprise.³¹ Versata eventually accused both Ameriprise and Infosys of breaching the terms of the

²⁹ No. A-14-CA-12-SS, 2014 WL 950065, at *1 (W.D. Tex. Mar. 11, 2014).

³⁰ *Id.* The Master License Agreement is a proprietary license requiring strict confidentiality and non-disclosure of code. *Id.*

³¹ Mark Radcliffe, *GPLv2 Goes to Court: More Decisions from the Versata Tar Pit*, OPENSOURCE.COM, (Dec. 15, 2014), <https://opensource.com/law/14/12/gplv2-court-decisions-versata>.

MLA.³² Ameriprise denied the breach and accused Versata of using the threat of a lawsuit to force Ameriprise to implement its newer, and more expensive, replacement for the software.³³

During discovery, Ameriprise found evidence that the Versata software contained source code developed by Ximpleware Corporation (“Ximpleware”).³⁴ Ximpleware originally licensed this code under the GPLv2.³⁵ Ameriprise argued that because Versata had used code subject to GPLv2, the MLA was void and that only the terms of the GPLv2 applied to its use of Versata software.³⁶ Further, it claimed that Versata was obligated to release its source code to them, free of charge and subject only to the GPLv2.

The original suit by Versata against Ameriprise was brought in state court as a breach of contract claim based on Ameriprise’s supposed breach of the MLA.³⁷ Ameriprise counterclaimed for breach of contract based on Versata’s alleged breach of the GPLv2.³⁸ Versata raised copyright pre-emption as an affirmative defense to its breach of the GPLv2 and Ameriprise did likewise. Ameriprise then removed the case to Federal court to resolve the copyright pre-emption issue.³⁹

A copyright pre-emption defense raises the federal question of whether the breach of the applicable terms and conditions of the GPLv2 results in a federal copyright infringement claim.⁴⁰ In Versata, the court therefore had to decide whether the failure to comply with the reciprocal provisions of the GPLv2 would result in a breach of contract or copyright infringement claim.

The court applied the Fifth Circuit’s two pronged test to determine if the contract claims were pre-empted by copyright law.⁴¹ The first prong requires a simple determination as to “whether [the claim] falls within the subject matter of copyright.” The second prong, “the ‘extra element’ test, . . . asks whether the asserted cause of action requires proof of ‘one or more qualitatively different elements’ than a copyright infringement claim.”⁴² If the claim is within the subject matter of copyright and there are no “extra elements,” then the state contract claim is pre-empted.⁴³ Even where the first prong is satisfied, any additional right or “extra element” identified by the court may sustain the state court contract claim.

³² *Id.*

³³ Aaron Williamson, *Software Litigation Opens Pandora’s Box of Key Open Source Issues*, 7 No. 3 *Landslide* 53, 54 (2015).

³⁴ *Versata*, 2014 WL 950065, at *1.

³⁵ *Id.*

³⁶ Radcliffe, *supra* note 31.

³⁷ *Versata*, 2014 WL 950065, at *1.

³⁸ *Id.*

³⁹ Williamson, *supra* note 33, at 54.

⁴⁰ *Id.*

⁴¹ *Versata*, 2014 WL 950065, at *3.

⁴² *Id.*

⁴³ *Id.*

The court found that the GPLv2 contained an extra element, the “viral component.”⁴⁴ The court held that Ameriprise claim that Versata failed to disclose the source code of its product was the breach of a contract term:

*The “viral” component of the GPL is separate and distinct from any copyright obligation. Copyright law imposes no open source obligations. ... Ameriprise has sued based on Versata’s breach of an additional obligation: an affirmative promise to make its derivative work open source because it incorporated an open source program into its software. Ameriprise’s claim therefore requires an “extra element” in addition to reproduction or distribution: a failure to disclose the source code of the derivative software.*⁴⁵

In its ruling, the court may have been answering the wrong question. Instead of determining whether the Copyright Act contains open source provisions, the more relevant questions may be:

- Can compliance with a requirement that does not address itself to the exclusionary rights in copyright law be a condition of a license, the failure to comply with such requirement resulting in an unlicensed use?
- What was the intention of the drafters of the GPL as it relates to whether the copyleft provisions of the GPL were to be treated as a condition or covenant?
- What is the probative effect of the intent of the author of a license agreement where the author is not a party to the license agreement?

By virtue of this holding, the court found it unnecessary to rule on whether Ameriprise should be considered a third-party beneficiary of the GPLv2 between XimpleWare and Versata. “Having found no basis for federal jurisdiction over this claim, the Court need not determine whether Ameriprise has standing to enforce the GPL as a third-party beneficiary.”⁴⁶

Squaring Versata and Jacobsen (Or Not): Conditions versus Covenants

The *Versata* court held that the obligation to release code to downstream customers was an “extra element,” squarely placing enforcement within the realm of state contract law. This leaves open the possibility that a customer may be able to compel disclosure of source code directly against the software company on a theory that it is a third party beneficiary of the agreement between the developer and software company, as shown in Figure 2, below.

⁴⁴ *Id.*

⁴⁵ *Id.* at *5 (emphasis supplied).

⁴⁶ *Id.*

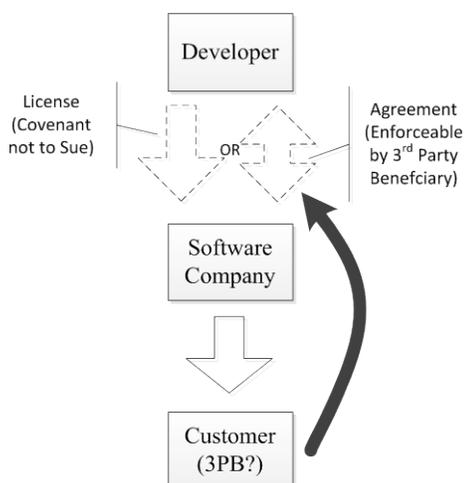


Figure 2: Third Party Enforcement

The case settled before the state court had a chance to rule on whether disclosure could be compelled on this theory. However, the mere fact that the federal court did not find copyright preemption opens software company users of GPL-licensed code to a greater risk of litigation and compelled disclosure.

In *Jacobsen*, the court found that affirmative obligations to affix copyright notice and attribution were conditions to a license. In *Versata*, the court found that an affirmative obligation to distribute source code along with a program is an “extra element” enforceable under contract law. These rulings, while not exactly contradictory, blur the line between contract enforcement and copyright enforcement.

Caveat: Not all Open Source Licenses are Licenses; Some are Agreements.

While the Free Software Foundation licenses — GPL, GNU Lesser General Public License, and GNU Affero General Public License — have always been couched as licenses and only enforceable through copyright law, some open source “licenses” are intended as true agreements. A primary example of a license of this nature is the Common Public License and its progeny, the Eclipse Public License (“EPL”).⁴⁷

The preamble to the EPL states: “The accompanying program is provided under the terms of this Eclipse Public License (“Agreement”). Any use, reproduction, or distribution of the program constitutes recipient’s acceptance of this Agreement.”⁴⁸ While the term “license” is used to title the body of text, it is clear that the author intended it at least be thought of and referred to as an agreement.⁴⁹

⁴⁷ For purposes of simplicity, we will refer only to the Eclipse Public License.

⁴⁸ Eclipse Public License, version 1.0 (available at <http://www.eclipse.org/legal/epl-v10.html>).

⁴⁹ Raymond Nimmer, *Legal Issues in Open Source and Free Software Distribution*, THE LAW OF COMPUTER TECHNOLOGY (1997, 2005 Supp.). A more detailed analysis of various licenses and whether their language indicates agreement or license can be found in this source.

The EPL contains further evidence that it is intended to be construed under the law of contracts. The EPL contains “boilerplate” language typically found in contracts. Perhaps most tellingly, the EPL contains a choice of state law clause. Not all open source licenses are licenses; careful examination of the license is required to determine how it might be enforced.

Conclusion

Suppose that you represent a potential purchaser of the software company in our fact pattern. If a software company refuses to disclose the source code of the company’s product to a customer, then it seems clear from our fact pattern that the software company would be in violation of GPL#1.

Following the rationale in *Jacobsen*, in such a situation, only the developer would have the right to bring an action against the software company. The developer may not care nearly as much about the customer’s problems as the customer itself does. Even if the customer is willing to pay the developer for the costs and expenses of bringing an action against the software company, the developer may not be interested in being a party to a lawsuit. Further, the remedies in such a case, damages for copyright infringement and injunctive relief, do not necessarily result in the customer receiving the source code.

Following the rationale in *Versata*, the customer may have the right to bring its own action, alleging that it is a third-party beneficiary of the agreement portions of GPL#1.

When representing the potential purchaser of the software company or the software company itself, there is a substantial difference in risk between a customer having a direct claim and only the developer having the claim.

To the open source community, which relies on the terms and conditions of an open source license to be enforced in accordance with copyright law, rather than the law of contracts, the world-wide enforceability of these licenses may be at stake.

The current status of this area of law is anything but settled. Of course, the best advice is proscriptive: “don’t put yourself in this position.” Once the proverbial horse is out of the barn, it is very difficult to give definitive advice with any certainty.