

Creating an Open Source Policy for Your Company

Stephen Mutkoski
Intellectual Property & Licensing Group
Microsoft Corporation

It is important to point out early on that there is likely no single “one-size-fits-all” open source policy; each company will likely have somewhat different answers for the most common open source-related questions. Each company’s distinct business plan and licensing model (and possibly the company’s level of aversion to uncertainty and risks) will dictate the appropriate response for that company to each of these common open source issues.

The most important thing you can do is to start this exercise of creating an open source policy is to think through carefully some of the common OSS issues or scenarios that will arise and determine in advance- in a well reasoned fashion- how your company should address or respond to these issues or scenarios. By way of example, if a developer requests permission to download and incorporate source code governed by the GPL into company products, how would you respond? How would your response differ if the code were governed by the BSD license? What if the request was for permission merely to download and install open source applications for use within the company? What if the request was for an employee to release one of your products in source code form under an open source license?

Why Every Company Needs an Open Source Policy:

There is a common misconception that open source software is “free” in many senses of that word, including that it is free of legal restriction and free of any risk or liability. In fact, unfettered use of open source software by your employees could obligate your company to share its intellectual property in unplanned ways (or risk an injunction if it fails to do so) and might subject your company to an increased risk of third party infringement claims. Moreover, your employees might become involved in open source “community” projects in ways that might result in the unintended sharing of your company intellectual property or the improper introduction of third party intellectual property into your products. Implementing an open source policy is crucial to insure these “mights” and could” do not occur without an express determination that they will not harm your company business model.

The “unplanned sharing” scenarios mentioned above could result from the outbound licensing conditions or requirements that come into play when a developer incorporates GPL or other similar “Copyleft”¹ code into a company product. By

¹ “Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it.... [A Copyleft license] gives everyone the rights to use, modify, and redistribute the program's code or any program derived from it but only if the distribution terms are unchanged.... Copyleft is a general concept; there are many ways to fill in the details.” See <http://www.gnu.org/licenses/licenses.html#WhatIsCopyleft>

incorporating GPL code into one of its products, your company would likely be required to release source code and grant broad IP licenses to the larger work it creates. The GPL does, after all, condition its license grants on the licensee making the same grants with respect to the larger work it creates. These “reciprocal” grants (and access to the product’s source code) could be inconsistent with your company’s existing or planned outbound licensing regime (for instance a business model that generates significant revenue from the sale of end user licenses for the particular product).

The potential for increase in third party infringement claims relates to concerns about open source “pedigree.” Code of unknown or dubious pedigree can transmit with it risks of third party intellectual property claims. You will likely want to consider whether the open source code your employees want to use was written under the same type of circumstances and with the same level of legal supervision you require within your company. In particular, you would likely want to consider whether the various contributors to a particular open source project might have day jobs at companies where they work on related technologies and are subject to employment agreements that effectively dictate that the “contributions” being made to the open source project are actually owned by the employer. Because intellectual property claims are not dependent on intent or knowledge, your company would be an infringer if unauthorized third party intellectual property were transmitted in the open source code.

There are a number of other issues relating to the interaction between your employees and the open source “community” including risks that your employees are inadvertently giving away valuable company intellectual property or bringing back in infringing third party intellectual property when they interact with the community.

Factors to Consider When Drafting Your Company Policy:

The range of responses various companies might have to these issues set out above results from the fact that the intellectual property issues presented can look surprisingly different viewed through the lenses of each individual company’s business plan and licensing model. In other words, while one company might care greatly that incorporation of GPL code into one of its products would require company to make the source code to the entire product available (and grant broad IP licenses to that entire product), another company might be willing to live with these conditions imposed by the GPL. Whether this is the case depends primarily on the extent to which the company derives revenue from licensing its products for a fee to users. Although the GPL does not rule out the collection of a fee in connection with the distribution of a work, it does rule out charging a license fee. Moreover, the broad IP license grants in the GPL (as well as the requirement of providing source code) effectively destroy the ability of a company to prevent others from obtaining, reproducing and redistributing the GPL work to others without paying the licensing fee (in fact the GPL is intended to encourage such redistribution).

Even within a company there could be varied approaches depending on the particular circumstances and the particular product. A company might have one set of

rules for incorporation of open source code into so-called “core” software assets where intellectual property licenses generate significant revenues but different rules for non-core software assets (where the company might not extract much in the way of licensing revenues).

As a preliminary step to formulating your company’s open source software policy, you should first consider your company business model, specifically your outbound licensing plans for software products. You should consider which products the company licenses for a fee to end users, how significantly those license fees contribute to the company’s bottom line and whether reducing those licensing revenues (a likely result if the products must be released under GPL terms) will significantly impact the company business plan.

If your company generates revenue primarily through consulting or services work or the sale of hardware and your business plan does not include future plans to generate revenues from licensing software, then you might not be averse to including GPL code in company products (and thus broadly licensing your IP on a royalty free basis), because you might not see a reduction in revenues. In fact, there may be situations where giving away software and associated IP (and making it available in source code form) might drive services revenues or sales of collateral company hardware or software products, resulting in an overall increase in revenues.

On the other hand, if your company revenues are generated by both services/consulting work and software licensing fees, you will need to consider the potential implications on licensing revenues of particular products, particularly those “core” products that account for high percentages of software licensing revenues. If your company revenues are generated primarily by software license fees, you will likely want to be more conservative, carefully scrutinizing each proposed instance of open source use to make sure that the significant revenues from licensing fees are not jeopardized.

Some Specific Scenarios to Consider:

In drafting your company’s open source policy, you will want to consider a range of potential scenarios that developers may bring to you. Obviously one of the most important scenarios for you to consider is the one in which your developers ask to include open source code in a company product. But you will also want to consider whether or when (a) company code should be released under an open source license, (b) employees should be allowed to participate in external “community” projects and (c) your company will set up external community projects for its products, including taking back in community contributions.

Depending on your company business and licensing models, you might ultimately make the decision to allow developers to include open source code into a particular company product. But given the fact specific nature of each inquiry, the complexity of open source licenses and the unsettled questions concerning what product architectures might require compliance with certain open source license provisions, you will likely

want to have a policy that requires approval of each specific instance where open source code is incorporated in a company product.

For example, depending on the open source license it might make a difference if the employee (a) cuts and pastes the code into company product (b) statically links to the code or (c) dynamically links to the code. And, your developers might interpret an open source license quite differently from how you would as legal counsel. Overall then, it might make sense to have a policy that requires approval prior to any incorporation of open source code into a company product, with each specific request being reviewed by an appropriate legal or business decision maker.

The person or persons responsible for reviewing requests for permission to incorporate open source code should be guided by a much more detailed policy document. This document should reflect the company's own business plan and should point out acceptable and unacceptable requests. For instance, one company might decide that its reliance on end user licensing of its products cuts against using any "Copyleft" code, but that with appropriate review it can get comfortable with "pedigree" risks from non-copyleft code. Another company, while generally restricting incorporation of GPL licensed code might allow incorporation of code licensed under the Common Public License (CPL) or Mozilla Public License (MPL) if certain product architecture requirements are met (i.e., files or modules are segregated). In the end, this detailed policy document should reflect what you think each open source license means (when it comes into play, what it requires you to do, etc.), what products your company needs to continue licensing for a fee as well as how you feel about potential pedigree issues with open source code.

In addition to addressing open source code "incorporation" as discussed above, you might want to include in your open source policy your company policy for internal use of open source products. There are a number of useful development tools released under open source licenses that your developers may want or need to use in developing your products. The vast majority of these tools are used only in the development process and they in no way are incorporated into the shipping product.² Your policy can either explain that tools may be used without limitation provided they are not in whole or in part incorporated into a company product, or set out an approved list of tools that you have confirmed do not cause code to be incorporated into the product under development.

You will also want to make sure that your policy addresses whether and when employees may release company code under an open source license. Many developers are actively involved in open source projects outside of work and may not understand the

² A tool called Bison is one such exception. See <http://www.gnu.org/licenses/gpl-faq.html> ("Some programs copy parts of themselves into the output for technical reasons--for example, Bison copies a standard parser program into its output file. In such cases, the copied text in the output is covered by the same license that covers it in the source code. Meanwhile, the part of the output which is derived from the program's input inherits the copyright status of the input. As it happens, Bison can also be used to develop non-free programs. This is because we decided to explicitly permit the use of the Bison standard parser program in Bison output files without restriction. We made the decision because there were other tools comparable to Bison which already permitted use for non-free programs.")

implications of releasing company code under an open source license. If the code includes valuable company intellectual property, the company may lose a competitive advantage in the marketplace, since release under most open source licenses would result in the grant of broad licenses to this intellectual property. Again, the specific instances where a company might want to prohibit or allow the release of certain of its code under an open source license will vary from company to company, but each instance should be reviewed by an appropriate legal or business decision maker (namely one who has a solid understanding of the company intellectual property policy and its licensing plans).

Whether you know it or not, your developers are likely involved in various “community” projects and activities. Community is a broad term used to refer to newsgroups, Sourceforge-like sites and other areas where developers “gather” and collectively work on building code or solving more general problems. You should consider whether these activities might expose your company to risks. For instance, if you have a product that you license to end users, you probably want to restrict how your developers who work on that product get involved in community projects that relate to the same or similar functionality. If you impose no such restrictions, your developers might wind up contributing valuable trade secrets, copyrights or patents to the community project. Although you may want your employees to be involved in certain aspects of the community, you likely do not want them to contribute all of the know how and other intellectual property that makes your product superior to the similar open source product. In addition, your developers might bring something back from the community and incorporate it into one of your products and, as discussed above, not realize the implications of doing this. Depending on your specific situation, you will likely want your open source policy to explain the possible risks of participating in community projects, whether or not on company time, and at a minimum require approval for involvement in projects involving functionality that is similar to what the employee develops at your company.

A final area that you should consider addressing in your open source policy concerns when or whether your company will set up or “sponsor” community projects and what your company will do with the fruits of such a project. While a community project might seem like an obvious choice for one or more of your company products (to harness free developer resources), there are a number of risks and administrative burdens you should consider before you undertake such a project. For instance, just as there are potential pedigree concerns with code your developers download from the internet, there are such concerns with code added in a community process. You would want to get some sense of who your contributors are, confirm they have intellectual property rights in the contribution sufficient to allow you to incorporate it into your company product (the Free Software Foundation in fact does just this) and possibly have them assign their rights in such contributions to ease future administrative burdens.

Your Corporate Open Source Policy as Part of a Larger OSS Compliance Plan:

Drafting a corporate open source policy is only the starting point for what should be a larger open source compliance plan. The diagram below briefly outlines the

additional components you should consider, including a corporate training plan, an exceptions approval process and various tools to check for and track open source usage, as well as compliance with any open source license terms.

