

NICOLAI LAW GROUP, P.C.

BUSINESS LAW & LITIGATION

Paul Peter Nicolai, Esq.
Also Admitted in New York,
Connecticut &
Washington, D.C.
Fellow, American Bar Foundation
Direct Dial Extension 222
E-MAIL Address:
PAUL.NICOLAI@NICLAWGRP.COM

Tarbell-Watters Building
146 Chestnut Street
Springfield, Massachusetts 01103-1539
Internet: WWW.NICLAWGRP.COM
Telephone: 413-272-2000
Facsimile: 413-272-2010

TO: Clients & Interested Parties

FROM: NLG Professional Staff

DATE: January 1, 2009

RE: Managing IP Compliance

Virtually all companies developing software now work in a mixed-IP environment. Software is created on layers of previous work, mixing internally developed software with licensed proprietary components and including open source elements. The re-use of available software components is inevitable. Not only does it reduce the time and cost of development, reusing components that have withstood time and the many eyeballs of critical peer review provides assurance of quality and reliability. Open source software is often necessary to meet important open standards. Just saying "no" to open source software is not a viable option for most companies in the business these days.

Prohibition Doesn't Work

Face it. Simply prohibiting open source software does not work. It may be convenient to have an absolute prohibition as official policy. A simple ban might have worked as long as there was nothing that could be done to actively monitor compliance. If the code had required a payment, the agreements would have gone through internal controls. Without a payment, its use easily evades all internal controls.

Why It Matters

Why should using open source software require attention? The short answer is technology is available to monitor compliance. One can no longer assume no one will ever know. The longer answer is developments have caused our society to be much more attuned to compliance concerns. Think about who might ask the questions and what the consequence might be if the answers were not satisfactory.

- Public companies now perform extensive due diligence reviews on the use of open source software before acquiring the intellectual property assets of a company.

NICOLAI LAW GROUP, P.C. • BUSINESS LAW & LITIGATION • 413-272-2000

This material is provided for information and education purposes to clients and others who may be interested in the subject matter. It is not legal advice or a legal opinion that can only be given on specific facts.

- Strategic partners want more than just assuring words that the companies they are allying with comply with their obligations.
- Customers demand broader intellectual property warranties and indemnities.
- Employees want to know their employer is meeting its obligations to the open source community.
- Investors want to know there are no skeletons in the closet affecting shareholder value.

The result is a new field called software compliance management. It will become a normal part of the operation of every company using software. In general, it means:

- Knowing what is in the code base and controlling the introduction of licensed materials into it;
- Knowing the obligations related to using licensed materials, whether the license obligations for combined components are compatible, and managing fulfillment of those obligations; and
- Managing the process from product architecture through deployment or distribution.

Although the immediate focus here is on open source software, the same asset management issues come up for code licensed from third parties. The trend is to deliver more source code, making even an inadvertent migration of proprietary source code a real possibility. There is also a trend toward reusing internally developed components. This requires effectively ensuring the use of the asset is consistent with business objectives. The replication of components across business units must be tracked for support and security purposes. All of these issues are addressed through software compliance management since they all start with knowing what is in the code base and managing compliance with use restrictions.

Due Diligence

The first step is to implement due diligence procedures for software asset acquisition. It helps you make sure you are not carelessly adding to the problems that will have to be addressed while you are preparing for your internal review. The learning that takes place in due diligence will make the internal review go more smoothly. Finally, the education process is much less painful when you are not dealing with your own products or people. It is difficult to be objective and use an issue as an opportunity to decide on proper policies and procedures going forward when that issue is tied to specific members of the management team and existing customer commitments.

There are many different ways to conduct due diligence. Due diligence relating to open source software is basically the same as due diligence on any issue. Checklists are used to get information. Oral interviews follow up on issues that deserve more attention. But there are methods uniquely suited to reviewing code base contents.

Certificate Of Originality

One method is preparing a certificate of originality. This is a list of detailed questions regarding the origins of the software. These forms are often handed to the developers who contributed to the code. The downside is the answers to the questions are all possibly useable in litigation. It is very difficult to prepare answers to detailed questions that are absolutely accurate for all purposes and in all contexts. This is particularly so when the questions are about a very complex subject matter and cover a time period months or years in the past. If the company can afford to invest in very sophisti-

cated intellectual property attorneys with deep technical expertise being deeply involved in the development of the responses, this can be an effective method. If the certificates are just handed to developers for completion, this method of gathering information can be damaging to everyone involved and others to come like an eventual acquirer who will be bound by the admissions as a successor in interest.

Code Review As Part Of Due Diligence

Introducing in-depth code reviews into a formal due diligence process requires thoughtful consideration. Especially if done as part of an acquisition process, there are a number of risks including exposing the acquiring entity's people to the source code before closing, the privileged or work product status of investigation results, and keeping access to the target's developers to complete remediation in the most efficient manner. Some of the questions that must be asked using this approach include:

Who does code review, where is code review conducted, and who has access to results?

When does code review happen?

When does remediation happen, who does the remediation, and who reviews the remediation?

Code Review Techniques

There is a broad range of code review techniques available to companies reviewing a code base. Some companies hire forensic computer consultants to review a code base. These experts are highly skilled programmers who scour code line-by-line for clues to origin and dependency. Many consultants and companies have internal tools that scan code for indications of copied code. The scan looks for words in the text like "copyright" or "license" or "free" or "open" or the names of individuals closely connected with the development of free software.

Newer tools analyze the code at a much deeper level and pick up copying even if the textual clues have been removed. They look for matches in the source code or indications that binary files have similar characteristics. These tools look below the component level and pick up copied statements; even statements that have copied material that has been reordered or modified. The scan results can present very difficult questions. When does merger of function and expression occur in source code such that we are no longer looking at copyrightable subject matter? What factors must be taken into consideration in drawing a conclusion that the copied material is de minimus? How much investigation is required before you can accept a developer's representation that identical code was not from copying but from the limited number of ways to do a given function?

Remediation

Once the code review is done, companies are faced with difficult decisions on remediation, especially in an acquisition setting. If the compliance failure is limited to replacing limited amounts of non-critical copied code, most companies feel comfortable proceeding with the transaction and letting the acquiring entity implement a remediation plan. The cost of the remediation efforts or the loss of functionality due to the removal of code may be a factor in valuation, but the delay in closing the transaction, which is often much more problematic, is minimal. In some situations, companies agree to a remediation plan to be completed before closing and, occasionally, use escrows or hold-backs to secure the obligations. If the compliance failure is more serious difficult questions arise about who will perform the remediation work. Acquiring entities will be concerned about perform-

ing the remediation work or directing the remediation effort before closing. The reason is that even if it were possible to avoid direct infringement, the chances of avoiding vicarious liability or being held responsible as a contributory infringer are not high. Obviously, it is far better for a company that might be an acquisition target or subject to due diligence review by a partner to conduct its own code review and complete all necessary remediation in advance.

How Will The Open Source Code Be Used?

The level of sensitivity an acquiring company will have toward open source code in the code base depends on the intended use of the software assets. If the acquisition is to temporarily fill a hole in the company's product line and the assets will remain in a subsidiary and not be rolled up into the parent, then there may be less sensitivity to problems disclosed in an extensive due diligence. The intent may be that the acquired code will be replaced by code developed as part of the company's existing product line.

If the intention is to merge the acquired code into the acquiring company's product line and combine it with other valuable software assets, the sensitivity will be extremely high and even minor problems may be enough to tip the balance toward make in a make or buy analysis.

It is not only what is found in the code scan that is important, but also how the discoveries jibe with the representations made by management and what they say about the effectiveness of business controls in place during development. Confidence in the management team will be reduced if there are too many surprises.

Internal Assessment

If a company conducts due diligence of all acquired code, it becomes obvious these same techniques ought to be used on the company's own code hygiene. This almost always starts with an assessment of the code base similar to the assessment performed on code to be acquired. It should not end there. Once the initial assessment is completed, real value comes in implementing automated, audit-able controls that enable the company at any time to answer questions regarding its code base and compliance status. An ideal process will have many benefits, including the following:

- Allowing the company to embrace open source software by setting the proper tone on compliance;
- Reducing the time and cost of development;
- Identifying issues early in the development cycle when good options are still available;
- Facilitating efficient use of legal counsel in the review process by gathering all available information automatically; and
- Efficiently tracking compliance through to product deployment or distribution.

For most in-house attorneys, the prospect of conducting an assessment of a code base causes a certain queasy feeling. They know that copying excellent software off the Internet to meet development schedules is extremely enticing and that, in the absence of technical controls, the company has been relying entirely on the sensitivity and good will of developers and their managers to impose discipline on the development process. In-house counsels also know that the process of conducting the assessment will be neither easy nor straightforward. All investigations result in the creation of ques-

tionnaires, interim processing, and preliminary conclusions that contain inaccurate information. It takes time to get to the point at which enough information is available and there is sufficient corroboration of all important data to feel confident that the results should be considered final. This healthy hesitation to bring an investigation to a close too quickly is particularly wise when the investigation involves technical data. Cross-discipline communications between experts in a technical field and lawyers are often conducted through the use of analogies. The analogy may work very well to assist the lawyer in understanding one particular aspect of the problem, but the lawyer often is drawn to extend the analogy to explain related matters for which the analogy does not hold. For in-house counsel at public companies, in particular, creating a lot of preliminary information regarding possible compliance failures requires consideration of complicated and exacting disclosure obligations.

Preserving Attorney-Client Privilege

Many consider whether there are steps that should be taken to preserve a possible claim of attorney-client privilege over the investigatory phase. Although the availability of privilege will vary from state to state and will, of course, not protect the company from the facts, steps taken to preserve the possibility of a claim of privilege are wise counsel for reasons unrelated to avoiding disclosure of work product. There are real benefits to managing the investigation within the confines of the privilege analysis. For example, it should be clear in advance:

- Who will conduct and who will be involved in the investigation;
- Who has requested the investigation and for what purpose;
- How the investigation will be conducted;
- What the schedule for the investigation is; and
- When it will be brought to a conclusion.

If a realistic schedule is established that provides for the conduct of a thorough investigation over a six-month period, and a date is set for review of the final results with the requesting executive on a date seven months later, it will be much easier to answer questions about why the executive was not made aware of preliminary information that became available three months into the process.

Make The Time

An internal assessment of source code requires the commitment of valuable resources for as long as it takes to finish it. This is not a project that can be begun and put on hold. It is important to make an honest assessment of the resources necessary to follow through to conclusion. It is tempting to pull the best development staff into a remediation effort, but this is probably not the best use of valuable resources. By identifying the right people, you can limit the number of people and recognize the importance of their contributions. Adequate recognition and encouragement is very important.

Prioritize

In planning the assessment, it is important to prioritize projects and avoid assessing the entire code base at once. The first project is a learning opportunity that enables following projects to be done more efficiently. Gather contextual information in advance for the code selected for review. For example, you want to understand what parts are shipped to customers and what source code is made

available to OEM partners before you begin reviewing code. If concerns are raised by the results of the review, you will not want to be looking for this information in a mode that will spread rumors. The goal is to avoid having more issues open than can be addressed efficiently.

The hardest part is triage. Don't let the perfect be the enemy of the good and get bogged down in minutiae. Save energy for the issues that demand attention. Avoid crying wolf. The goal is a determination in good faith that the company is in compliance. The goal is not the eradication of all code that might arguably indicate some level of familiarity with source code owned by another. Sometimes the proper course is determined by fine distinctions about what is copyrightable subject matter, what is permitted as fair use or whether a use can fairly be deemed both quantitatively and qualitatively de minimus.

Anticipate Employee Concerns

It is important to anticipate employee concerns and to limit involvement to those employees who are necessary to the process. Many employees are very sensitive to obligations to the open source community and rumors on compliance concerns raise issues not only about the company's legal obligations, but also its moral obligations and commitment to fairness. Remember that laws protecting whistleblowers may apply to allegations of copyright infringement.

Remediation Planning

Remediation plans may include many factors. Sometimes, simply revising the interaction between components is enough. Contacting the authors of open source projects to get the rights necessary to the use is an alternative. What is common is that these plans require cooperation between the development and legal communities.

Build A Consensus

Developing a remediation plan should avoid mandating a course of action constructed in a vacuum. The technical options will only be fully explored when there is a consensus an issue must be addressed. Since remediation always involves redirecting valuable resources, you should expect a push-back to any proposal and be prepared to convince that remediation is required. Developers will not want to be involved in a remediation effort unless they sense they will be fully supported and the value of their contributions will be recognized.

Third-Party Issues

The most difficult remediation issues are those that extend to software that has been distributed to customers or other third parties. Do you have an obligation to notify them? Is there a method for ensuring the third party receives and uses replacement code after remediation? Has the third party further distributed non-compliant code? Is the third party obligated to install delivered updates? Do the warranties or indemnities in your agreement require any action? Thorough preparation in advance of any discussion is extremely important. Focus should remain on compliance going forward.

Wrap-Up

When the assessment is done, bringing the process to closure requires enormous discipline. Report back to the requestor on the established date. Avoid previews or interim reviews that leave the executive with information about compliance concerns that is not final and does not have developed remediation recommendations. Generally, there is no need to review minor remediation work com-

pleted. The report should be limited to issues requiring continued remediation work. Discussion of specific, fully developed recommendations and alternatives will avoid any hiatus during which management knows of risk but has no plan in place to address it.

Real-Time Compliance Programs

After completing a compliance review of the existing code base, the company should maintain control over future introduction of licensed materials into the code base. The first step here is to evaluate existing processes. There is generally some formal or informal process in place already that should be understood before any new process is imposed. Identify opportunities to capitalize on good things already happening.

Involve your own financial compliance experts. CFOs and auditors are experts in internal controls. They will be familiar with materials prepared by organizations with the mission to define, review, and assess the sufficiency of internal control programs.

There are several approaches to pre-approval of the use of open source software commonly employed even in the absence of any formal, audit-ready compliance program.

Identification Of Open-Source Projects

Some companies request developers to identify in advance those open source projects they would like to use so that one or more identified individuals can analyze the project for appropriateness. The review is often quite extensive, looking beyond the license declared for the open source project. All information available about licenses that may apply to subcomponents of the project and the compatibility of those licenses is explored. In some cases, the author of an open source project is contacted to confirm licensing terms and to provide some assurance of originality.

After project approval, it is placed on an internal Web site that indicates it is has been approved for use. This review may be done by a committee. The creation of an internal group of individuals savvy about open source in general and knowledgeable about open source use in the enterprise is an excellent way to communicate an organization's openness to use of open source. It fosters awareness, serves educational purposes, creates formal records, builds internal precedent that improves decision-making, and establishes a process to bring issues regarding open source usage to closure.

Listing Approved Licenses

Another method is to publish a list of approved non-reciprocal licenses. Open source software made available under selected licenses is pre-approved and may be used in development.

This method ignores the fact that many extremely valuable projects are under reciprocal licenses and these projects can, in some cases, be used without incurring obligations to release source code. It also can encourage license laundering. Code originally made available under a reciprocal license may have been copied for use in projects made available under non-reciprocal terms. It is always important to consider the true source of open source software.

Analyzing Code Origins

A final approach is to analyze code origins in final product quality review before the product release. Not only can this create a logjam, it means issues come up when there are few good options and

enormous pressure to release and remediate later. It is far preferable to raise the issues early in the development cycle when many options remain viable.

To recap, a compliance program usually begins with establishment of a due diligence policy to assess code acquired or licensed from third parties. This is followed by an assessment of the code base and remediation of compliance issues raised in the review. When the internal review is complete, an evaluation of existing processes is the first step in the development of a process that is tailored to meet the goals and culture of the organization. Finally, a compliance process implementing automated and auditable business controls is rolled out across the enterprise.

Visualizing Compliance

In today's environment, a product offering is likely to be a complex, multi-layered program including components from different sources; internally-developed legacy code, code from commercial providers, and code from the open source community. Code from each of these sources was developed by many individuals. Some are employed by the developer; many are employed by other companies or by universities. All of the code must be tracked from its origins to the end product to know its origin.

For outsourcing, contract provisions establish code origin and make sure assignments of intellectual property rights are effective under local law. These include representations, warranties and indemnities. Even with contracts, concerns linger because of the lack of personal contact with the developers and concerns about different cultural perspectives on the importance of intellectual property rights. Implementing a technical solution as part of the contracting process can both set the proper tone regarding controls and verify compliance.

Although concerns are often raised about open source software, the same concerns apply to proprietary code. The amount of open source software that may have leaked into a code base may be greater because of an absence of controls over code available for download, but the same issues apply to proprietary source code often readily available overseas. Software compliance management is just another business process that solves the problems presented by failing to manage the introduction of licensed materials into a code base. An effective program sets the proper tone on compliance, raises issues in a timely manner, and improves communications between developers and attorneys by providing immediate access to all of the information that is relevant to resolution.