29
Oct
2010

Licensing Considerations When Integrating FOSS and Proprietary Software

Recently, I was looking for resources to help explain the implications of integrating FOSS (Free and Open Source Software) with proprietary software. This question is important for any organization who might want to build an “embedded” or “dedicated” system or product which might include either their own proprietary software or third party applications. I discovered that most of the information available about FOSS licensing addresses this issue rather obliquely. This post will cover the basics so that such organizations can see how straightforward it would be to include FOSS in their projects. Standard disclaimers about my not being a lawyer apply.

Use of any software system requires understanding the software licensing involved. There are many dozens of FOSS licenses that could apply in your situation, so understanding the details is necessary to assess compatibility. In broad terms, these can be effectively summarized by the Debian Free Software Guidelines and The Open Source Definition. Any software that complies with those specifications will freely (in both the “freedom” and money senses of the word) permit running proprietary and FOSS applications on the same system. This is easier said than done. Since Debian has analyzed most common FOSS and quasi-FOSS licenses, their archive and their license page can be used to assess how the license might work in your situation (for example, anything in “main” would be OK for co-distribution and running in mixed environments). So we always start with Debian’s meticulous and well-documented analysis to assess any license.

From a high-level perspective, the requirements that FOSS licensing will impose on an organization wanting to include it with proprietary software will primarily be in the form of providing appropriate attribution (acknowledgement) and providing the source code for any FOSS software (including any modifications) shipped as part of an integrated system. Typically the attribution requirement can be satisfied by simply referencing the source code of the FOSS components. The source code requirement can be met by providing the source code for all of the FOSS distributed as part of the integrated system, for example, by placing it on a documented ftp site.

There are some subtle issues that may arise during the software development process if proprietary and FOSS code are “linked” together. In such cases it is necessary to ensure that code over which one wants to assert proprietary control is only combined with FOSS code that explicitly supports such commingling. So it is necessary for a company wanting to keep its code proprietary to keep it “separate” from the FOSS code used in the integrated system. The use of the words “linked” and “separate” is intentionally vague because the terms of each FOSS license will need to be examined by legal counsel to understand the precise requirements. In these situations, there are license compliance management systems that can be adopted to help ensure that this separation is maintained.

Those are the basic issues. The references below describe these and related issues involved with Linux & FOSS licensing in much more depth:

- FOSS Licensing Wikibook

- Guide to FOSS licensing
- Free and Open Source Software Compliance: The Basics You Must Know (registration required)
- Establishing Free and Open Source Software Compliance Programs: Challenges and Solutions (registration required)
- Shareware Redistribution of FOSS Software
- A Practical Guide to GPL Compliance
- CC HowTo #1: How to Attribute a Creative Commons licensed work
- International Free and Open Source Software Law Review

Posted by CJ Fearnley *in* Development, Systems Management, *0 comments*