Free and Open Source Software Use

Benefits and Compliance Obligations

Philip Koltun, The Linux Foundation

Abstract. Many systems developed for and deployed by the U.S. government now use Free and Open Source Software (FOSS). But FOSS use comes with potential license obligations. Essential compliance activities include identification of FOSS used in products along with communication of a FOSS bill of materials; review and approval of planned FOSS use; and satisfaction of license obligations. Compliance policies, processes, training, and tools enable contractors and government sponsors to use FOSS effectively. The Linux Foundation's Open Compliance Program provides many resources to assist with compliance.

Introduction

FOSS has been widely adopted as the software of choice in many core areas of computing. Linux dominates today in embedded systems and in servers, and other FOSS has gained widespread acceptance for operational use. Vibrant communities support Linux kernel development and many popular FOSS packages.

FOSS is all about freedom—freedom to use, study, modify, and distribute software under an open source license. Think of the word "free" as in free speech, not as in free beer.

The DoD has clarified[1] that FOSS use is acceptable and can provide significant benefits: high quality, reliable, and secure software resulting from continuous and broad peer-review; availability of source code for modification, which enables rapid response to changing situations, missions, and threats; avoidance of vendor lock-in; freedom to use and deploy in any context; low total cost of ownership; and so on.

Beyond the use of Linux, FOSS can be found in many domains, including (to name a few) software development tools and environments; computing infrastructure; graphics; mapping and geospatial imaging; modeling and simulation; communications and networking; security; database; and real-time computing. Indeed, a 2003 MITRE Corporation study[2] identified 115 FOSS applications in use in the DoD. Undoubtedly that number has grown quite substantially in the years since. The DoD has published an online Frequently Asked Questions (FAQ) page that dispenses useful information about DoD use of FOSS.[3]

With FOSS use comes responsibility. Typical license obligations consist of inclusion of attributions, copyright notices, and license text along with the product when it is distributed externally. Providing complete and corresponding source code or an offer of source code may also be required, depending on the FOSS licenses involved.[4]

Normally, license obligations are triggered when external distribution of a product occurs. The entity that distributes a product containing FOSS bears the responsibility for meeting relevant license obligations; they can not just point at an upstream supplier and say, "See them for whatever you are entitled to." On the other hand, what constitutes "external distribution" may be subject to legal interpretation. The aforementioned FAQ indicates that as long as a product acquired or developed by the U.S. government is not conveyed outside the U.S. government, external distribution has not occurred.[5] As a result, use of the software within the U.S. government context normally would not trigger license obligations.

Why, then, should the defense software community served by CrossTalk concern itself with FOSS compliance issues? At least two perspectives are worth examining. First is that of the DoD contractor delivering software to the government who uses FOSS to implement required functionality. Second is that of the government program manager overseeing the contractor and assuring that the government receives the freedoms, rights, and information to which it is entitled.

The contractor must assure that it knows what FOSS is included in its deliverable software and that it can satisfy any license obligations, so that the government will be able to enjoy its freedoms. Inasmuch as a contractor may use subcontractors, the task of knowing what is in the delivered code can be somewhat demanding.

The government, on its side, has an interest to preserve its options to distribute software to allies or to the public, actions that might trigger FOSS license obligations. So the government's interest is to assure that software delivered to it by contractors comes with all necessary freedoms. As a result, contractual agreements should require FOSS disclosure and FOSS obligation satisfaction from its suppliers. The government should also investigate its suppliers' FOSS compliance practices as part of its background diligence in contracting. Does a supplier have a policy on FOSS use, compliance training for its teams, automated code scanning to facilitate discovery and recognition of FOSS inclusion, a procedure to prepare a FOSS bill of materials, and so on? The Linux Foundation's "Self-Assessment Checklist" can be used effectively to assess supplier compliance practices and engage suppliers in discussion about compliance[6]. There would be good reason, as well, to incorporate FOSS compliance discussions in SEI assessments conducted to qualify the contractor.

FOSS Compliance

FOSS compliance refers to the aggregate of policies, processes, training, and tools that enables an organization to effectively use FOSS and contribute to open communities while respecting copyrights, complying with license obligations, and protecting the organization's intellectual property and that of its customers and suppliers.

What business processes enable organizations to comply with license obligations and project managers to assure obligations are satisfied? For a product being distributed externally, compliance involves three core activities: identification of FOSS; review and approval of planned use of FOSS; and satisfaction of license obligations for the included FOSS. Each of these will be discussed further below.

Identification of FOSS

First, identification of all FOSS in a product comes from the dual processes of disclosure and discovery. With disclosure, engineers and product managers of the contractor and its external suppliers typically identify FOSS based on prior knowledge of where the code came from. Discovery refers to audits (either manual or automated) that are used to identify FOSS code and its origin.

Reliance only on disclosure can be problematic. Few products these days are written from scratch. Most evolve from legacy products and externally acquired source code (either FOSS or commercially licensed software), with new code being written to implement differentiating features and functionality. Sometimes millions of lines of code may be included in a product, some of it pre-dating the engineers currently working for the company. It is unlikely that any one individual or team will know all of the code and where it came from. So it is hardly surprising that disclosure alone would be incomplete or inaccurate. Commercial scanning tools aid in the discovery process and are marketed by companies such as Black Duck Software, OpenLogic, Palamida, and Protecode, among others.[7] A number of open source scanning tools are also available.[8]

New technologies are being developed to codify and communicate in standard format a FOSS bill of materials. For instance, the Software Package Data Exchange specification (SPDX™), version 1.0, was released in the fall of 2011.[9]

Review and Approval

Reviewing and approving planned FOSS use is the second essential activity in compliance, typically requiring a panel of skilled and knowledgeable individuals known as an Open Source Review Board (OSRB). An OSRB must review FOSS use in context, so a product architectural diagram will be needed to show how the product's software components (including FOSS) interface and interact. The OSRB examines licensing implications of the architecture, compatibility of components from a license perspective, and resultant license obligations. Therefore, an OSRB must incorporate the expertise of skilled software architects and licensing experts.

Satisfaction of Obligations

The third essential activity in a compliance program concerns satisfaction of FOSS license obligations. Many organizational actions must come together to assure obligations can be met. As stated earlier, obligation fulfillment typically involves inclusion of attributions, copyright notices, and license text along with the product when it is distributed externally. Providing complete and corresponding source code or an offer of source code may also be required, depending on the FOSS licenses involved.

Ultimately, an effective compliance program must integrate compliance activities into day-to-day business processes so that identification, review and approval, and obligation satisfaction steps are routinely accomplished in time for scheduled product delivery.

Individuals or teams responsible for product documentation must perform necessary tasks to assure that documentation obligations are met.

As part of the process to satisfy source code obligations, the contractor typically should place into a software repository the complete source code corresponding exactly to each FOSS package used in a given product release. The complete source code may include any associated interface definition files, plus the scripts used to control compilation and installation of the executable. Verification activities should assure that source code used to produce product binaries has been cleansed of any inappropriate comments and that all FOSS packages in the product have been approved by the OSRB.

Ultimately, an effective compliance program must integrate compliance activities into day-to-day business processes so that identification, review and approval, and obligation satisfaction steps are routinely accomplished in time for scheduled product delivery. Key elements of a compliance program include company policy, employee training, assignment of compliance responsibility, staffing of the compliance function, and automation to enhance efficiency and accuracy. Compliance program implementation dovetails very nicely with CMMI®.[10]

Key process capabilities that must be brought to bear in compliance include supplier management, software configuration management, training, software architectural design and review, and verification, at a minimum.

Compliance Resources

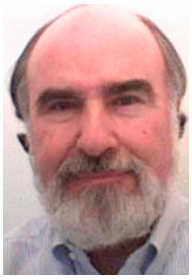
The Linux Foundation's Open Compliance Program is the software industry's only neutral, comprehensive software compliance initiative. By marshaling the resources of its members and leaders in the compliance community, the Linux Foundation brings together the individuals, companies, and legal entities needed to expand the use of FOSS while decreasing legal costs and reducing fear, uncertainty, and doubt.

Organizations seeking greater insight into compliance practices can take Linux Foundation compliance training courses; download freely available Linux Foundation compliance white papers and the Self-Assessment Checklist; participate in the SPDX working group; participate in the FOSSBazaar community and discuss compliance best practices; and access other helpful resources. More information can be found at <http://www.linuxfoundation.org/programs/legal/compliance>.

Disclaimer:

CMMI® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

ABOUT THE AUTHOR



Dr. Philip Koltun formerly directed the Linux Foundation's Open Compliance Program. He has implemented comprehensive open source compliance programs for Motorola and NAVTEQ, including policies and procedures, training, OSRB function, supplier compliance, and compliance tool introduction. Previously, he managed a software productivity and quality center at Harris Corporation; facilitated SEI, ISO, and Baldrige-style assessments; managed software suppliers; and consulted on business process improvement. He earned a Ph.D. in Computer Science from University of North Carolina-Chapel Hill.

The Linux Foundation
1796 18th Street, Suite C
San Francisco, CA 94107
Phone: (815) 353-2748
Fax: (415) 723-9709
E-mail: philip.koltun@gmail.com

NOTES

1. "Clarifying Guidance Regarding Open Source Software," October 16, 2009, from the DoD CIO, <http://cio-nii.defense.gov/sites/oss/20090SS.pdf>
2. "Use of Free and Open Source Software (FOSS) in the U.S. Department of Defense," The MITRE Corporation, 2003, <http://cio-nii.defense.gov/sites/oss/2003survey/dodfoss_pdf.pdf>
3. See http://cio-nii.defense.gov/sites/oss/Open_Source_Software_(OSS)_FAQ.htm
4. The author is not a lawyer and this article should not be construed as providing legal advice. Please consult qualified counsel for interpretation of license terms and other questions requiring legal guidance.
5. See "DoD Open Source Software (OSS) FAQ" at <http://cio-nii.defense.gov/sites/oss/Open_Source_Software_(OSS)_FAQ.htm#Q:_Under_what_conditions_can_GPL-licensed_software_be_mixed_with_proprietary.2Fclassified_software.3F>
6. "Self-Assessment Checklist," The Linux Foundation, November, 2010, <http://www.linuxfoundation.org/programs/legal/compliance/self-assessment-checklist>
7. See <http://www.blackducksoftware.com>, <http://www.openlogic.com>, <http://www.palamida.com>, and <http://www.protecode.com>.
8. See, for example, the FOSSology tool at <http://fossology.org> and OpenLogic's Discovery tool, <http://www.openlogic.com/downloads/ossdiscovery.php>.
9. See <http://spdx.org>
10. <http://www.sei.cmu.edu/cmmi>