



December 2004

## **Open Source Software: Recent Developments and Public Policy Implications**

“Open source” software has become a topic of great interest in the press and among policymakers. Open source software – both as a type of software and as a software development and licensing model – is an emerging business reality that needs to be better understood. It is appropriate to address this subject in a balanced way and to look at the practical, pragmatic issues surrounding the emergence of this software development model in the marketplace, and the public policy implications.

### **World Information Technology and Services Alliance**

The World Information Technology and Services Alliance (WITSA) is a consortium of 65 information technology (IT) industry associations from economies around the world (list attached). As the global voice of the IT industry, WITSA is dedicated to:

- advocating policies that advance the industry's growth and development;
- facilitating international trade and investment in IT products and services;
- strengthening WITSA's national industry associations through the sharing of knowledge, experience, and critical information;
- providing members with a vast network of contacts in nearly every geographic region of the world; and
- hosting the World Congress on IT, the only industry sponsored global IT event.

Founded in 1978 and originally known as the World Computing Services Industry Association, WITSA has increasingly assumed an active advocacy role in international public policy issues affecting the creation of a robust global information infrastructure, including:

- increasing competition through open markets and regulatory reform;
- protecting intellectual property;
- reducing tariff and non-tariff trade barriers to IT goods and services; and safeguarding the viability and continued growth of the Internet and electronic commerce.

## “Open Source” Software

Open source software (OSS) refers to software for which the underlying "source" code (the program text written by a human in a programming language such as C, C++, or FORTRAN) is available for inspection and modification by anyone interested in doing so. This contrasts with proprietary software that is not shared with users or anyone else outside the team of employees, companies and researchers that developed the code, except for very limited purposes and under licensing controls. The term "open source," coined in the late 1990's, is often applied in two distinct ways: (1) to a software program licensed under particular terms and (2) to a software development model.

*Licensing features:* Open source software may be subject to a number of different software licensing models.<sup>1</sup> For example, open source software may be licensed with terms that allow the author's work to be appropriated by others, in specified ways, without requiring the new end product to be made available to the public. An example is software subject to the BSD license.<sup>2</sup> On the other hand, open source software may also be licensed under an arrangement setting forth specific rules for its use, distribution, reuse, or modification. A common arrangement is the GNU General Public License ("GPL"). Not all open source software is covered by the GPL. There are in fact several open source licenses with different terms. The Open Source Initiative is the organization that determines whether or not a given license meets the requirements to be deemed an "open source license." There are presently over 50 such licenses listed on the OSI web site.<sup>3</sup>

The term "open source" should not be confused with the term "public domain." Authors of open source software may be very specific in their licensing

---

<sup>1</sup> A useful table outlining the terms and features of ten different "popular or significant" open source licenses may be found at M. Fink, *The Business and Economics of Linux and Open Source* 42-25 (Prentice Hall PTR 2003)

<sup>2</sup> The BSD or Berkeley Software Development License only requires that redistributions of code covered under it identify the original copyright holder(s) and pass through a disclaimer in the form of an "as is" warranty. See BSD license, <http://www.opensource.org/licenses/bsd-license.php> (last visited August 13, 2004)

<sup>3</sup> See <http://www.opensource.org/licenses/>.

requirements for use, distribution, reuse or modification of their open source code. They do not simply make the code available to the public without restriction, as would be the case if it were in the public domain. The term “open source” also does not equate with “free” in the sense of price or cost of ownership.

*Development characteristics:* The open source software development model has many variations, and it is impossible to make absolute generalizations that hold across all examples. Typical characteristics to this software development model are changing as it becomes more common in the commercial world. The open source model usually involves an author of the code as well as a number of collaborators or contributors. The author or some other individual/group/foundation becomes the “maintainer” that plays a key coordinating role in the development project. The collaborators become known as the “community.” The community can range from being quite a small number of individuals to numbering in the hundreds and the author can be a group as well as an individual. The collaborators and contributors have access to all of the code and can review it and suggest modifications or improvements. In the classic case, all of the collaborators are volunteers with an interest in the code.

Today, a particular development community is likely to be made up of a diverse group, ranging from employees of an enterprise whose job it is to participate in a particular open source development project to others, often in the education and research communities, who are interested in the development of the code for a variety of reasons. The maintainer adopts a role of finalizing the program version and accepting modifications into a future version of the program, so not all changes are automatically incorporated; they are simply made available for review by the community. More recently, the establishment of the Open Source Development Laboratory by a consortium of technology companies has brought commercial practices more often associated with proprietary software to some major open source development projects such as the Linux operating system.

## **Open Source Software in The Marketplace**

The software marketplace is large and complex. The value of shipments of commercial software in 2004 is predicted to approach \$150 billion.<sup>4</sup> This number does not include the vast value of software that is created within an organization for its internal use. *Open source* software is a growing part of the software marketplace. However, while there are thousands of open source software projects on the Internet, a relatively small number have significance in the software marketplace today. These include the operating system Linux, the Web server Apache, development tools like Eclipse, Perl and PHP, and MySQL, a relational database system. Today there are an estimated 2.6 million Web and file servers running Linux. The Apache HTTP Server Project, an effort to develop

---

<sup>4</sup> Gartner Group prediction. Business Week, January 12, 2004.

and maintain an open-source HTTP server software product for various modern desktop and server operating systems, is the leading product in its field.<sup>5</sup>

## **Drivers of open source software**

Following are a few of the drivers behind a more widespread adoption of open source software:

*The advent of the Internet.* One driver of open source development is the availability of the modern Internet to serve as a mechanism for the growth in open source development communities that are necessary for successful development and continued improvements in the programs.

*Software license cost.* There is a perception that open source software products cost less than products developed by companies following a closed source software development model. Both software development models are in flux today as each works to serve the needs of customers by focusing on different pricing models; licensing is only part of the total value equation.

*Flexibility.* Supporters frequently argue that because the source code is viewable to all, the underlying technology can be used in many innovative ways, offering a flexible platform to meet present and future software needs.

*Global innovation.* With many more developers able to view the source code, supporters argue that the pace of innovation is greater as the barriers to software modification are lower.

*Security.* Source code transparency is argued to promote more secure software because a wider group of people may inspect the software for flaws.

*Customer involvement.* Supporters suggest that open source development models may provide more opportunities for customer-driven innovation than the traditional proprietary approach.

Arguments persist over whether some of the advantages identified on this list are real or merely perceived.

---

<sup>5</sup> In a July 2004 Netcraft survey incorporating roughly 57 million Web sites, Apache ranked as the market share leader, with 69.5% market share and 16,774,339 active servers. According to Netcraft, the number of sites deploying Apache has risen to about 35 million. It has held the market leader position for about 10 years. See [http://news.netcraft.com/archives/2004/07/01/july\\_2004\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2004/07/01/july_2004_web_server_survey.html) (last visited August 13, 2004)

## Open Standards and Open Source Software

“Open standards” are publicly available technical specifications, without individual vendor restrictions on access, that are developed or affirmed in a collaborative or consensus based process. If the standard includes technology covered by intellectual property claims (incorporated with the permission of the rights holder), it may be subject to non-discriminatory licensing terms with reasonable royalties. In any case, the dissemination of the standard may (or may not) be accompanied by licensing terms that require the recipient to share the fruits of any improvements it makes in the standard with other licensees in the marketplace, in order to maintain the specification’s quality as a “standard” available to all market participants.

“Open standards” are not the same as “open source software.” There is no requirement that “open standards” be implemented by open source software.<sup>6</sup> Open standards (technical specifications) may be implemented by all types of software. Open standards do not inherently favor one business model of software development over another, but instead, as technical specifications, document requirements that must be met for implementation of the technology to be compatible with the standard.<sup>7</sup>

## IT security and Open Source Software

Open source advocates and proprietary developers make various claims about the security implication of their development models. Open source advocates say that the “open community” development model encourages many more individuals to study the code and search for, find, and correct vulnerabilities. This leads to a belief that open source is actually more secure than proprietary code, because so many programmers may review the code for vulnerabilities and then add security patches to close breaches they have found. It has also been argued, however, that this openness allows malicious hackers to review the source code and develop strategies to breach security restrictions or create their own security breaches that are difficult to detect.

It is worth emphasizing that the correct security question for software itself has less to do with development models and more to do with the quality of code, and

---

<sup>6</sup> “Open source standards” do exist. For example, a group called the “Free Standards Group” ([www.freestandards.org](http://www.freestandards.org) (last visited August 13, 2004)) is engaged in a number of standards projects, such as developing standards and test suites to allow/enhance portability of applications across Linux distributions and different numbered releases. Sometimes, however, it appears simply that the terms “open source software” and “open standards” are confused because both contain the word “open.”

<sup>7</sup> An example may be helpful. HTML is an open standard. Mozilla, a product of Netscape, is an open source web browser software program that complies with the HTML standard. Internet Explorer, a web browser supplied by Microsoft with proprietary code, also complies with HTML standards.

whether it is more or less carefully written and developed. Quality code can be developed under both models. All code should be subjected to security evaluation and review. Moreover, software quality is only one of many important security considerations that need to be taken into account. Information and cyber security remain a combination of product and process, relying on informed users and administrators, as well as various security technologies and features of software, hardware and other physical systems, as well as the conditions of access to those systems.

At this point, one model of software development has not proved to be superior for security over another. All are accepted, all have their own limitations, and all are being used in the enterprise

## **Public Policy Implications of Open Source Software**

### *Procurement Preferences*

Since the late 1990's, governments have considered changing their public sector procurement laws to give preference to the open source development model by either creating barriers to acquisition of commercial software (or preferences for acquisition of open source software) or making the purchase of commercial software by government outright illegal. WITSA does not favor government-mandated preferences at any level. Governments, like all potential and existing customers, should choose software on a technology-neutral and vendor-neutral basis, examining the merits of the technology, its advantages and its total cost, not by banning software based on its licensing or development model. A blanket policy can never allow a competent software buyer to effectively weigh all factors. Whenever possible, WITSA and its member associations will help raise awareness among state institutions and, especially among international donor organizations about the availability of open source and proprietary software solutions.

## **Funding for Research and Development**

Many governments provide funding to government and academic institutions so they may undertake basic software research. The interest of these public institutions in research and development is often to use the created innovations for the benefit of society and is often the foundation of many commercial products. This funding often comes with guidelines that encourage, or ensure, the transfer of the technology into the private sector. Overly restrictive licensing terms can interrupt this smooth flow of innovation and deny its benefits to the public. When the public has funded R&D work on the technologies, the public should be certain that appropriate guidelines are followed to ensure technology transfer that permits further development and commercialization of the technology. Public funding should not favor any particular model of software

development. Software companies should be able to continue to benefit from commercialization of these publicly funded innovations.

## **Intellectual property concerns**

Software is mostly copyrightable expression (just as is a book, movie or music) and, as such, it is subject to the protections and limitations of intellectual property under the copyright law.<sup>8</sup> Similarly, software source code, like many other written works (e.g., customer lists, secret formulas for products, strategic plans for future competition and an almost infinite variety of similar materials) can be protected against unauthorized disclosure under trade secrets laws and with contractual non-disclosure agreements. Authors of software are entitled to the same intellectual property protection as authors of other forms of copyrightable expression.

Most software companies rely on a broad range of intellectual property protections, including trade secret, trademark, copyright and patent protection. Over time the software industry has come to rely on intellectual property rights to promote an atmosphere of innovation, to create an environment for sustainable businesses, and one that provides incentives to encourage firms to invest substantial resources to create new products. Both the proprietary development model and the open source development model rely on copyright law as the foundation for the allocation of rights under various license agreements, including the GPL.

## **Recommendations**

As open source software begins to emerge as a business reality in many markets, policymakers will begin to address the public policy implications for both the private and public sectors. Therefore, WITSA suggests that policymakers be guided by the following principles:

Intellectual property has value as determined by a competitive marketplace without undue regulatory interference, such as arbitrarily restrictive procurement guidelines.

Software should be procured for government, industry, or individual use based on its merits, its evaluation as the “best value” to the customer, not on its license or development model category. Such procurement decisions should be made on a technology neutral basis.

---

<sup>8</sup> It may be possible to describe a patentable invention solely in terms of software, giving rise to so-called “software patents”, but that is beyond the scope of the present discussion.

Open source and proprietary software are legitimate software categories providing increased choices for all customers.

Government sponsored research should not be about choosing to invest to support a particular business model, but rather should seek results that can be further developed and commercialized by the broadest possible community of software developers, regardless of their business models.

Encourage a “best value” analysis, including consideration of the total cost of ownership, when acquiring a software solution, including the cost of implementation testing and maintenance expense, as well as the benefits of adding new system capabilities to increase customer flexibility and value.

Support global open standards. We support their voluntary, consensus-based, industry-led development and adoption and note that open standards must be made available in ways that reflect their value, which can include utilizing “royalty free” terms and “reasonable and nondiscriminatory royalty rates” as appropriate.

Government should not set standards, except for itself in its role as a customer in the marketplace for its own systems. When setting standards for itself, government must use performance-based metrics for standards without reference to any particular model of software development.