

1

Government Policy toward Open Source Software: An Overview

Robert W. Hahn

This chapter provides a brief overview of policy issues associated with open source software and offers an economic framework for thinking about these issues. It includes a thumbnail sketch of the main points made by each of the contributors, highlighting areas of agreement and conflict on key policy questions.

This book examines the impact of government policy on open source software. “Open source” refers to access to the source code, written in a programming language, that constitutes a working software program. With open source software, users and others can read the code and change it to suit their needs.¹

Several open source software programs and their relatives, free software programs, are widely used today.² The best known is the computer operating system Linux.³ To date, Linux has been most successful as an operating system for servers—computers on a network used for tasks such as managing printers, storing files, and sending web pages to users—but could expand to the desktop (PC) market in the future. Other examples of widely embraced open source products include Sendmail (an email



server program), Apache (a web server program), and StarOffice (a business applications suite).⁴ And acceptance is growing: Linux gained enough use as a server operating system market in 2001 to be included in industry tracking studies.⁵

Physically, open source software and other software look the same. If you made copies of two operating systems—say, Linux and Windows—and put them on new CDs, you couldn't tell the difference unless they were labeled. Essentially, they both consist of binary code, strings of ones and zeros. Yet the way in which Linux and Windows were created is totally different. And the way in which they can be legally modified is totally different.

Linux comes out of the “open source” tradition, while Windows is proprietary. If you obtain a copy of Linux, the source code comes as part of the package. By contrast, access to the source code for Windows is tightly restricted and provided separately to Microsoft business partners on a “need to know” basis. Not surprisingly, then, the process by which open source and proprietary closed source software is created comes from very different—and sometimes conflicting—traditions.

On the one hand, programmers often flourish as part of communities that prize cooperation and openness. Status within the community is largely derived from showing how good one is at programming—which requires showing off the source code—and how committed one is to furthering the collective effort—which requires providing source code for others to work from.⁶ The Internet made it far easier to develop software this way and is generally viewed as the catalyst to the open source boom.

But open source can pay off on the demand side, too. Some users of software greatly value the option “to look under the hood” and to have the ability to make changes. Access to source code, for example, makes it possible for information technology professionals who maintain computer networks to tailor generic software to their specific needs and to debug software on the fly. Indeed, one reason the Linux operating system for server computers has been so popular is that it gives sophisticated users great flexibility and reduces their dependence on software vendors.

On the other hand, widely used programs like Microsoft's Office suite and Oracle's database series underpin the success of some of the most profitable corporations on the planet. And since the commercial value of these blockbuster products is almost exclusively embodied in source code, commercial software makers typically have powerful incentives to limit access.

Like other businesses relying on intellectual property, they use patents and copyrights to safeguard the value of their property. Unlike most others, however, software companies' first and strongest line of defense is secrecy.⁷ It may be relatively easy to copy a CD containing a compiled version of Windows that is ready to bring a PC to life. But that copy would not provide any clues to the design of the underlying source code and thus would be of little or no help in cloning Windows—even to a highly experienced programmer.

For many software users, the absence of source code is not a drawback. While open source software is often geared toward information technology specialists, to whom the availability of source code can be a real asset, proprietary software is often aimed at less sophisticated users. Most of the people who use Windows on their PCs—either at work or at home—would not know what to do with source code even if they had access to it. These users are more concerned about standard features and ease of use than they are about customization. As a result, proprietary software developers spend considerably more resources on documentation, customer support, and product training than do open source providers. Proprietary vendors also expend more effort researching what average users want through surveys and such, rather than relying on early product testers and fellow software developers' opinions.

Open source software can coexist with commercial software in market niches where flexibility is paramount and communal development works well. In others, where ease of use or customer support and training are more important and financial incentives spur innovation, commercial software may dominate.

But the availability of both open source and proprietary software in the same market can create tensions. For example, some



open source software is licensed under terms that make it difficult for the software to coexist with commercial software. Indeed, the Free Software Foundation apparently sees the destruction of incentives to produce software under the conventional commercial model as a prime objective. Its General Public License (GPL) requires that any modifications or additions to GPL-licensed software also be licensed under the same terms, making it difficult for developers to profit financially from distributing this software.⁸

The key features of the GPL, which is sometimes called “viral” because its integration with other code automatically transforms the integrated version into GPL code, may have important economic implications. For example, while Sun Microsystems was able to create a commercial hit—with early versions of the Solaris operating system—from the Berkeley Software Distribution (BSD)-licensed Unix operating system source code, no one will ever replicate that feat with Linux. Any software derived from GPL-licensed Linux must be distributed with an accessible copy of the source code, and the rights to use any modifications in the source code must be free to all, thereby making it very difficult to profit from using Linux code as part of proprietary software.

Arguably more important, some governments have expressly tilted the playing field toward open source software, subsidizing its production and use. Instead of choosing software based on its merits—reliability, security, ease of use, and so on—these governments favor open source as a matter of policy. For instance, Singapore is offering tax breaks to companies that use the open source Linux operating system instead of commercial alternatives like Windows. This and other examples of government support of open source software around the world are shown on table 1-1.

In addition to the handful of measures that have already passed in Brazil, Germany, and Singapore, many more governments worldwide have open source proposals pending, and the European Parliament has called on member nations to promote the use of open source software whenever practical. Table 1-2 lists

Table 1-1. Existing Government Support of Open Source Software around the World

<i>Country</i>	<i>Proposal</i>	<i>Date adopted</i>
<i>Asia</i>		
Singapore	Government agency (the EDB) charged with planning and executing strategies to boost the Singapore economy offered temporary tax reductions and financial grants to fund Linux-related projects.	2001
<i>Europe</i>		
Germany	Compensation legislation requiring that right holders (such as copyrights holders) generally may not waive in advance their rights to adequate compensation for use of their works. An exception inserted at the request of open source lobbyists permits waiver of this right if the right holder grants rights to simple use of the work to the general public.	January 2002
	Bundestag mandated a new IT environment: Linux on servers, Windows on desktops.	March 2002
	Government-IBM agreement that offers German government offices discounts on IBM machines with preinstalled Linux provided by German Linux distributor SuSE.	June 2002
<i>Latin America</i>		
Brazil	Legislation mandating open source software be given preference in municipal governments of Recife, Campinas, Solonopole, Amparo, Sao Carlos, and Porto Alegre.	2001

Sources: The German compensation bill text is available at www.bundestag.de/aktuell/bp/2002/bp0201/0201028.html (German; August 21, 2002); for the Bundestag's adoption of the new IT environment, see www.heise.de/newsticker/data/anw-14.03.02-012/ (German; August 21, 2002). For legislation in Recife, see www.pernambuco.com/tecnologia/arquivo/softlivre1.html (Portuguese; August 21, 2002); for Campinas, see www.aful.org/politique/perou/english/referencias.html (August 21, 2002); for Solonopole, see www.solonopole.ce.gov.br/leis/614-2001.htm (Portuguese; August 21, 2002); for Amparo, see www.cipsgapf.hpg.ig.com.br/projetos/dimas_marchi.htm (Portuguese; August 21, 2002); for Sao Carlos, see www.softwarelivre.rs.gov.br/index.php?menu=maisnoticias&codigonoticia=1009655036 (Portuguese; August 21, 2002); and for Porto Alegre, see www.seprors.com.br/swlivre.htm (Portuguese; August 21, 2002). For a general overview of global proposals, see www.infoworld.com/articles/hn/xml/02/06/12/020612hnoosnapshot.xml (August 21, 2002).

Table 1-2. Pending Legislation on Open Source Software around the World

<i>Country</i>	<i>Proposal</i>	<i>Date proposed/ last action</i>
<i>Europe</i>		
France	Parliamentary bill forbidding government-related institutions to use anything but open source software.	December 1999
Italy	Bills mandating an open source software preference in all governmental offices.	February 2002 and March 2002
Spain	Bill requiring regional governments to prefer and promote open source products.	May 2002
	Bill submitted to the Catalan parliament mandating an open source preference in all regional administrative bodies.	May 2002
	Motion by the Izquierda Unida Party urging the Senate to ensure that all public administration websites, documents, and software are Linux-compatible.	July 2001
<i>Latin America</i>		
Argentina	Bill mandating use of open source software by all provincial administrations in the Buenos Aires province.	June 2002
	Bill mandating all governmental offices to use “free software.”	September 2000
Brazil	Legislative proposals mandating preference for open source software in all governmental offices.	1999, 2000, 2001
Peru	Legislative proposals mandating preference for open source software in all governmental offices.	March 2002 and April 2002

Sources: For the parliamentary bill in France, see www.senat.fr/leg/pp199-117.html (French; August 21, 2002); for the bills in Italy, see www.senato.it/leg/14/bgt/schede/ddliter/16976.htm (Italian; August 21, 2002); for the parliamentary bill in Spain, see www.senado.es/legis7/publicaciones/pdf/congreso/bocg/b0244-1.pdf (Spanish; August 21, 2002); for the bill submitted to the Catalan Parliament, see www.hispalinux.es/modules.php?op=modload&name=sections&file=index&req=viewarticle&artid=49 (Spanish; August 21, 2002); and for the motion by the Izquierda Unida Party, see www.senado.es/legis7/publicaciones/html/textos/i0259.html#9 (Spanish; August 21, 2002). For the bill in the Buenos Aires province, the proposal in Cordoba, and the bill mandating all governmental offices to use “free software” in Argentina, see www.grulic.org.ar (Spanish; August 21, 2002) and www.aful.org/politique/perou/english/referencias.html (August 21, 2002). For the legislative proposals in Brazil, see www.camara.gov.br/internet/sileg/prop_detalhe.asp?id=17879, www.camara.gov.br/internet/sileg/prop_detalhe.asp?id=19028, and www.camara.gov.br/internet/sileg/prop_detalhe.asp?id=26688 (Portuguese; August 21, 2002). For the legislative proposals in Peru, see www.gnu.org.pe/proley4.html and www.gnu.org.pe/proley3.html (Spanish; August 21, 2002).

regulations that are still working their way through the legislative process.⁹

To economists, the market itself seems a natural place to resolve the conflict between diverging models for developing and distributing software. But this is only true if the market for software works reasonably well to benefit consumers. If, however, there is a significant “market failure,” some kind of government intervention may be justified. In the case of software, such failures could arise either on the demand side or the supply side—if either buyers or sellers do not capture the full value of the product.¹⁰

Even if there were a significant market failure, however, government intervention may not be justified on economic grounds. A reasonable economic standard for government intervention is when the benefits of such intervention are likely to outweigh the costs by a substantial margin.¹¹

Is the software market characterized by a significant market failure? Or is the movement to promote open source an ironic throwback to an era in which government was widely seen as the appropriate manager of technological change? And what, if any, government intervention is needed?

The following chapters offer the reader diverse views on government policy toward open source software by four leading experts in the field. In chapter 2 James Bessen argues that open source software meets specialized needs not met by either packaged or customized proprietary software. He contends that open source extends the software market by addressing market failures associated with incomplete contracts and asymmetric information. To encourage open source software development, he argues, the government should remove the impediments it has imposed in the form of software patents, which tilt the market in favor of proprietary developers.

In chapter 3 David Evans argues that there is no market failure in the provision of software and therefore no need for government remediation. On the contrary, the government should act like a business in making its own software investment decisions, evaluating software on its merits as a product rather than



attempting to promote a particular kind of software. Moreover, if the government decides to support open source through research and development funding, it should remain consistent with its approach in other fields and help to promote the commercialization of the resulting research by licensing software on terms that are less restrictive than those imposed by the GPL.

In chapter 4 Lawrence Lessig argues that the government has broader interests in software than private companies, including an interest in achieving and maintaining an open platform. As Lessig explains, “Between two systems for producing a public good, one that releases the information produced by that good freely and one that does not, all things being equal, public policy should favor free access.”¹² Thus a “neutral” government could still come out in favor of open source software.

In the final chapter, Bradford Smith posits that the marketplace, rather than the policy arena, creates the best combination of incentives and flexibility to ensure that software continues to satisfy consumer needs. He argues that government policies favoring open source over proprietary software would disrupt the software “ecosystem.” Instead, government could play a role in promoting software research under licensing terms that allow the results to be commercialized.

While there are clear differences of opinion among these experts, there are also some areas of agreement. Table 1-3 provides a glimpse of where they stand on key policy questions.

The first row of the table examines whether the author explicitly identifies a significant market failure in the development or production of software. As noted above, economists generally think it is important to identify such a failure before considering government intervention in a market. Interestingly, only one of the authors identifies a significant market failure in the provision of open source software. Yet several of the authors do think significant government policy interventions should be implemented or considered.

The second row in the table examines the author’s view on subsidies for open source software. None explicitly supports direct subsidies for open source software.

Table 1-3. Viewpoints on Government Policy toward Open Source Software

Issue	James Bessen	David Evans	Lawrence Lessig	Bradford Smith
1. Identifies a significant market failure in the development or production of open source software.	<p>No, <i>but</i>: notes that open source solves a market failure in software provision by overcoming imperfect contracts and information asymmetry.</p>	<p>No: the software industry has performed extremely well in terms of production and innovation without any government intervention.</p>	<p>Yes, <i>but</i>: open source developers cannot fully capture the value of their work, but this does not necessarily destroy the incentive to innovate.</p>	<p>No: there is currently no market failure in software.</p>
2. Argues for direct government subsidies for open source.	<p>No: where the government has intervened (in patents), it has created a market failure.</p>	<p>No: the government should not be in the position to pick industry winners.</p>	<p>No: the government should be neutral but careful to address its own interests.</p>	<p>No: only the marketplace can satisfy actual market needs.</p>
3. Believes open source and proprietary software both have important roles to play.	<p>Yes: open source software extends the market.</p>	<p>Yes: the open source method is an important organizational innovation.</p>	<p>Yes: both forms of software production should compete in the market.</p>	<p>Yes: both open source and commercial software are integral parts of the software ecosystem.</p>
4. Argues that government should make software procurement based on benefit-cost framework similar to a profit-maximizing firm.	<p>Yes: products should be considered on their merits for the project at hand.</p>	<p>Yes: governments ought to pick the best products for their own internal needs.</p>	<p>Yes, <i>but</i>: the government has a greater interest in externalizing benefits such as an open platform.</p>	<p>Not addressed.</p>
5. Suggests government should encourage firms to commercialize research and development by not permitting GPL or “viral” license to be used in government-funded research.	<p>Not addressed.</p>	<p>Yes: governments should ensure that the results of publicly funded research are not subject to licensing restrictions.</p>	<p>No: no general rule can be asserted. Sometimes it will make sense for the government to support GPL projects.</p>	<p>Yes: governments can help promote innovation through research and development and should facilitate commercialization of the resulting research.</p>
6. Suggests government should change patent policy to allow open source to be more competitive.	<p>Yes: Congress could restore subject matter limitations on patents and strict standards on patent quality.</p>	<p>No, <i>but</i>: might strike a better balance between protection and innovation if software patent standards were tougher and patent lives shorter.</p>	<p>Yes: a system with software patents is biased against open source and free software.</p>	<p>Not addressed.</p>

The third row of the table shows that all authors believe that open source and proprietary software have important roles to play in the market. Indeed, both sources of software are widely used, though it is fair to conclude that proprietary software plays a dominant role at this point. In chapters 3 and 5, respectively, Evans and Smith discuss the remarkable growth that has occurred in proprietary software. In chapter 2, however, Bessen notes that a growing number of users are turning to open source, especially for custom software projects.

The question of the government's role in procurement of software is addressed in the fourth row. All three authors who address the issue argue that the government should base its purchasing decisions for software on criteria similar to those used by private firms—that is, on narrow economic grounds. Lessig notes the importance of considering the value derived from standardization and openness when the government makes its purchasing decisions. I also think a private firm would want to consider such values.¹³

Some scholars and some firms have argued that the government should not fund research that is licensed under the GPL or similar “viral” licenses. They note that such funding could dampen innovation by deterring private firms from using the results in new services or products because of the peculiar nature of viral software.¹⁴ This issue is of more than theoretical interest. The U.S. government is already licensing software this way. For example, software underwritten by NASA and by the Sandia National Laboratories has already been distributed under the GPL. The fifth row in table 1-3 summarizes the views of the authors on this issue: Evans and Smith argue against such funding, but Lessig argues that it may sometimes make sense for the government to support GPL projects.

The final row in the table shows the authors' views on changing patent law to promote innovation in software. There seems to be a consensus among the three authors who address the issue that some change would be useful. Bessen and Lessig argue that patent law is currently biased against open source software, and they favor stricter standards for patents. Evans also believes

there may be an argument for stricter standards because it would generally help software innovation. This is an area that would benefit greatly from more empirical research.

This book is meant to contribute to the debate on open source policy. At this point, I think it is fair to say that a strong quantitative case has not been made for government intervention in this market. This is perhaps why none of the experts here favors direct government subsidies for open source. At the same time, there may be good economic reasons for changing aspects of government policy toward software in areas ranging from procurement to patents. This book is aimed at shedding light on such policy issues as well as on broader aspects in the debate over open source software.