MENU

Open Standards Requirement for Software

The Requirement

An "open standard" must not prohibit conforming implementations in open source software.

The Criteria

To comply with the Open Standards Requirement, an "open standard" must satisfy the following criteria. If an "open standard" does not meet these criteria, it will be discriminating against open source developers.

1. **No Intentional Secrets:** The standard MUST NOT withhold any detail necessary for interoperable implementation. As flaws are inevitable, the standard MUST define a process for fixing flaws identified during implementation and interoperability testing and to incorporate said changes into a revised version or superseding version of the standard to be released under terms that do not violate the OSR.

2. **Availability:** The standard MUST be freely and publicly available (e.g., from a stable web site) under royalty-free terms at reasonable and non-discriminatory cost.

3. **Patents:** All patents essential to implementation of the standard MUST:
   - be licensed under royalty-free terms for unrestricted use, or
   - be covered by a promise of non-assertion when practiced by open source software

No Agreements: There MUST NOT be any requirement for execution of a license agreement, NDA, grant, click-through, or any other form of paperwork to deploy conforming implementations of the standard.

5. **No OSR-Incompatible Dependencies:** Implementation of the standard MUST NOT require any other technology that fails to meet the criteria of this Requirement.

Posted July 24, 2006

Terms of Service. For questions regarding the OSI website and contents please contact us.

Proudly powered by WordPress. Hosted by Pressable.

MENU

The Open Source Definition

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.
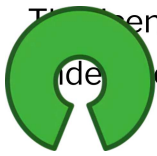
4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

The Open Source Definition was originally derived from the [Debian Free Software Guidelines](#) (DFSG).

Version 1.9, last modified, 2007-03-22

Here's the historical "[Annotated OSD](#)" from the early 2000's.

Terms of Service. For questions regarding the OSI website and contents please contact us.

Proudly powered by WordPress. Hosted by Pressable.

Open Standards Requirements for Software – Rationale

Draft 5; 2006-09-19

Please send comments and suggestions via the [OSI Contact Form](#).

Preamble

As the Internet shows so clearly, there is great social, technical, and financial benefit that comes from massive interoperability. This interoperability was not achieved in a vacuum, nor was it necessarily intended or happened as expected by the major players of the industry in its early days. But the processes and practices of the IETF, the W3C, the effect of events like the early Interop conferences, and the inclusive participation of virtually any party willing to follow the published protocols ultimately created something broadly useful and valuable.

Metcalfe's Law predicts that the value of interoperability increases geometrically with the number of compatible participants, and Reed's Law predicts that the utility of a network (implied by interoperable equivalence) increases exponentially due to the number of possible subgroups that interoperability enables. Both theories have successfully informed the investment of literally billions of dollars of capital investment as the Internet has become mainstream. Whichever law ultimately governs, interoperability is a positive function governing value, and thus any force that diminishes interoperability must be carefully scrutinized as it relates to ultimate and/or total value.

If interoperability is a grand goal as it relates to software, then standards are the critical tools for achieving this goal. Moreover, standards that permit:

- any license (free, open, or closed)

- any implementation

- any implementor

will expand the universe of interoperability compared with standards that militate against specific types of license, specific implementations, or specific implementors. Thus, for all the benefit that a standard *can* offer, the important question to ask is what the standard *does* or *does not* offer based on its implicit or explicit license or technology freedoms or restrictions.

At this point in time, it is has become largely intuitive across the industry and among users that broad and widely accepted standards are a Good Thing, better, certainly, than proprietary standards. In spite of these widely understood benefits, many market participants paradoxically believe that they can obtain a commercial or strategic advantage through unique, i.e., non-standard functionality. Some seek to burnish the perception of their products or technologies by claiming that they implement "open standards" while at the same time adding extensions that are not part of the standard. Others go farther, claiming that their unique implementations are themselves "open standards", a reversal of standards logic. The result is that the (usually undefined) term "open standard" has become more of an aspirational term than a defining term, a problem that we seek to rectify.

While the Open Source Initiative is not in a position to completely *define* what is meant by the term "open standard" (a term that may well become enshrined into law in some countries, and hence the perview of legal authorities, not the OSI), we believe that the OSI does have both the opportunity and the obligation to try to *constrain* the problem. In particular, as custodians of the Open Source Definition, we believe it is essential to help decision makers and developers understand how "open standards" relate to "open source software" and to ensure that a poor understanding of the former does not discriminate against the latter.

The Purpose of Open Standards

The purpose of an open standard is to increase the market for a technology by enabling potential consumers or suppliers of that technology to invest in it without having to either pay monopoly rent or fear litigation on trade secret, copyright, patent, or trademark causes of action. No standard can properly be described as "open" except to the extent it achieves these goals.

The industry has learned by experience that the only software-related standards to fully achieve these goals are those which not only permit but encourage open-source implementations. Open-source implementations are a quality and honesty check for any open standard that might be implemented in software; whether an application programming interface, a hardware interface, a file format, a communication protocol, a specification of user interactions, or any other form of data interchange and program control.

To help industry participants (suppliers, consumers, and regulators) identify and specify standards that permit open source implementations, the OSI has defined a minimal Open Standards Requirement (OSR). The OSI has also created a set of Criteria that can be used to judge whether a standard fully complies with that Requirement.

Problems not solved (because not in scope)

The problem of creating a robust and high-quality software standard, one that meets the needs of vendors and customers, users and implementors, public and private sectors, now and over time, is complex and context-dependent. In some cases, experienced hands are essential to the successful birthing of a standard; in others, a single idea from a single mind ensures the integrity of the problem definition and hence the problem's solution. The Open Standards Requirements for Software does not prescribe how open standards for software should be created, debated, ratified, and maintained except that they not preclude a viable implementation in open source.

There are any number of reasons why a particular open standard for software may succeed or fail in the market, and meeting the Open Standards Requirements is no guarantee that the standard will magically succeed on that fact alone. However, standards that preclude open source implementation are unhelpful to what is now a significant community of software developers and users, and it would be disingenuous to call any such standards "open" when they are, in fact, closed.

Issues

- How can we be broad enough to be relevant, yet precise enough to be defensible?

- Implementation *vs.* Specification

- *cf.* Larry's paper is at: http://www.rosenlaw.com/DefiningOpenStandards.pdf

1. Everyone should be free to implement open standards in both proprietary and open source software.

2. Open standards should be available to everyone on royalty-free terms.

3. Open standards should be developed using a collaborative, balanced and consensus-based approval process.

4. Open standards should be developed under formal and binding commitments for the disclosure and licensing of copyrights and patent claims.

5. Open standards should be made available under reasonable reciprocal licenses that require licensees to share under the same terms their own patent claims reading on the standard.

6. The specifications for open standards should be available to everyone on open source copyright license terms.

Terms of Service. For questions regarding the OSI website and contents please contact us.

Proudly powered by WordPress. Hosted by Pressable.