Why the Public Domain Isn't a License

by Lawrence Rosen

The notion of tossing software into the air to be blown about haphazardly by the wind is not entirely frivolous. The image is apt to describe the public domain.

Software engineers use the term "public domain" as if it means a place where anyone can do anything they want with software. Public domain software has no owner. Even the government doesn't own it. It is simply "free," as in the terms "free beer" and "free speech."

There *is* such a thing as the public domain. In it one can find Bach's sonatas, Shakespeare's plays, the drawings of DaVinci, and the design of the Eiffel tower. These things literally are available for anyone and everyone to use without permission.

Intellectual property enters the public domain only when it grows old. Everything else, including certainly any computer software of recent vintage, is owned by somebody somewhere. It is not "free" for the taking.

The legal monopolies for software under copyright laws last a very long time. Under current law, copyright extends for the life of the author plus 70 years; in the case of pseudonymous or anonymous works, or works made for hire, copyright extends for 95 years from the year of its first publication or 120 years from the year of its creation, whichever expires first. The software industry is new, and so it is rare today to find any important software for which the copyright has expired. (Congress recently extended the length of copyright term in a provision that has been described derisively as a special boon to the Disney corporation to protect its copyrights in Mickey Mouse comics. That extension has been challenged in the U.S. Supreme Court as inconsistent with the Constitutional objective to grant copyright monopolies in order to encourage the progress of science and the arts.)

Just as there is nothing in the law that permits a person to dump personal property in the public highway, there is nothing that permits the dumping of intellectual property into the public domain — except as happens in due course when any applicable copyrights expire. Until those copyrights expire, there is no mechanism in the law by which an owner of software can simply elect to place it in the public domain.

There is one exception to this in section 105 the Copyright Act. Works written by the U.S. government cannot obtain copyright protection and so are automatically in the public domain. Obvious examples of this are court decisions and Congressional statutes. Be careful, though. This exception applies only to works created by employees of the U.S. government, not typically by contractors to the government. University researchers and government laboratories doing work for the government ordinarily own copyrights in their works and can license them to third parties.

For these reasons, the "public domain" solution for free and open source software is largely irrelevant.

Even though there is no useful "public domain" repository of computer software, it is still possible for the creator of software to give it away. One doesn't have to be a lawyer to craft appropriate language:

> "This is my software. I hereby give it away to anyone who wants it for any purpose whatsoever."

Unfortunately, lawyers will explain, such gifts are illusory. Under basic contract law, a gift cannot be enforced. The donor can retract his gift at any time and for any reason (with some mostly unimportant exceptions). This is scant security for someone intending to make long-term use of a piece of software.

This "Give-It-Away" license provides no protection for anyone if the donated software causes harm. Obviously one cannot intentionally give away something he knows to be dangerous; that is criminal behavior! But neither, in this litigious society, can one escape a lawsuit just because his gift was only accidentally harmful. As any lawyer will warn his client, the risk of such a license is far greater than the warm feelings that enrich the soul of the giver. One important value of a license is the opportunity to disclaim warranties and distribute the software "AS IS." If you give software away, you may retain a risky warranty obligation.

Notice also that the donor under this "Give-It-Away" license has not placed any restrictions on the gift. For example, a recipient can make secret changes to the donated software and re-release the changed version to the world for a fee under a proprietary license. To many people in the free and open source movement, this violates another fundamental objective: the recipients of free or open source software should abide by the same "published source" rules as the original donor. If recipients distribute the donated software, with or without changes, they should also publish their source code. The "Give-It-Away" license doesn't force reciprocal source code publication. (Neither, for that matter, do the BSD, MIT, Apache and similar software licenses.) If you want to impose conditions on copying or distribution of software – even the minimal set of conditions allowed under the open source definition – you must use a license rather than rely on a gift to the "public domain."

Caveat emptor. Use the "Give-It-Away" license at your own risk. And don't accept gifts of software presuming they are in the public domain.

Lawrence Rosen is an attorney in private practice, with offices in Los Altos and Ukiah, California (www.rosenlaw.com). He is also executive director and general counsel for Open Source Initiative, which manages and promotes the Open Source Definition (www.opensource.org).

