



ACCESS™ White Paper

Open Source Best Practice



Open Source Best Practice

The use of the word “free” throughout this document should be interpreted as a reference to freedom rather than gratuity.

Overview

Linux® and more generally open source software are often associated in end-users’ minds with free software. For others, such as software and hardware companies that have to deal with the open source landscape, Linux is generally associated with source code redistribution and code license contamination, as well as with intellectual property rights. It is these three points that generate the biggest fears in discussions about embracing the open source software world. The purpose of this document is to debunk myths about open source software and licenses.

Open Source Software Licenses

The Open Source Initiative (the “OSI”)¹ is a non-profit organization that manages and promotes open source software. It determines whether or not a software license is in line with the OSI definition of Open Source², which is based on 10 criteria. In particular, to be approved by OSI, a license to software must require the free (unrestricted) redistribution of that software, both the binary version and its complete source code. The license terms must also allow modifications and creation of derivative works that must be distributed under the

same license terms as the original software.

According to these criteria, the OSI has approved 58 different license schemes so far. The most well known ones are the GNU General Public License (GPL)³ and the GNU Lesser General Public Licenses (LGPL)⁴, both distributed by the Free Software Foundation⁵.

The terms of the GPL grants users the freedom to use, study, copy, change and redistribute software programs under the condition that the modified source code for those software programs accompany any distribution of the binary files. Entitled Copyleft⁶, this concept advocates free software and subsequent improvements to be given back to the community, such as the GNU system and the Linux Kernel. To enforce these principles, GPL specifies that any software that embeds a GPL licensed program, or a portion of it, automatically inherits the GPL and thus must make its source code available for redistribution. This also applies to software that would either link dynamically or statically with a GPL licensed program. For this particular reason, one talks about license contamination.

The LGPL is just like the GPL but it is less restrictive in the sense that it allows the creation of proprietary software. Indeed, a software program that combines or links software, itself containing a LGPL library or a portion of it, does not inherit from the LGPL.

¹ Open Source Initiative, OSI, <http://www.opensource.org>

² Open Source Definition, OSD, <http://www.opensource.org/docs/definition.php>

³ GNU General Public License, GPL, <http://www.gnu.org/licenses/gpl.html>

⁴ Open Source Definition, OSD, <http://www.opensource.org/docs/definition.php>

⁵ Free Software Foundation, FSF, <http://www.fsf.org>

⁶ Copyleft : <http://www.gnu.org/copyleft/copyleft.html>

⁷ Mozilla Public License, MPL, <http://www.mozilla.org/MPL/>

One additional license, the Mozilla® Public License (MPL)⁷, is interesting to consider. As with the license described above, this license specifies that MPL'd source code that is copied or changed must remain under the MPL. However, code under the MPL can be combined in a software program with proprietary files without contaminating them. In other words, this license allows you to take advantage of the benefits of MPL'd source code while creating proprietary software.

The exposure to other components' licenses is an intrinsic factor around open source. Therefore it is necessary for developers to take into account the different license constraints at all phases of development. These constraints are summarized in Table 1.

Code based on:	Can be proprietary	Exposes IP to open source ¹	Becomes open source ²	Contaminates larger works
Proprietary Code	Yes	No	No	No
"MPLed" Code	Yes	Yes	Modified files only	No
"LGPLed" Code	No	Yes	Yes	No
"GPLed" Code	No	Yes	Yes	Yes

Table 1: License Constraints

¹- IP added in such licensed code becomes open source

²- Code that includes such licensed code becomes open source

Now that clarifications have been given regarding constraints associated with GPL and LGPL, it is important to understand where these licenses can be found in a Linux® based operating system.

Licenses Throughout the Platform

Each Linux platform is typically split into two parts that, besides any technical differences, may be subject to different license terms.

The first part, the kernel space, is where Linux itself resides. It contains device drivers, one or more file systems, a scheduler, and optionally networking components and other operating-system dependant code. All of the components that are included in the kernel space, or that are dynamically or statically linked with it, are required to be licensed under the GPL by virtue of the viral nature of the GPL. Nevertheless, this GPL contamination stops at the edge of the kernel space. Applications that are executed outside of this space and communicate with the kernel solely through system calls (a hardware architecture-dependent mechanism that cedes program control to the kernel), can be licensed under any terms desired by the application owner. These applications need not be licensed under the GPL, as long as the applications themselves were not developed separately using GPL-licensed code.

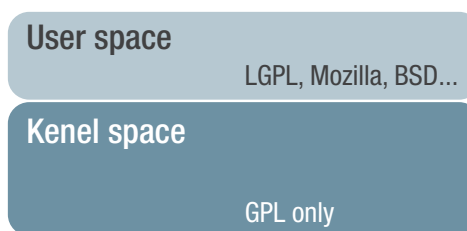


Figure 1: Licenses areas in Linux

The second part of the platform, which is exempted from the kernel GPL contamination so long as it contains no GPL components, is the user space. This is the place where typical middleware components

and applications with different license types can co-exist. Middleware is defined as code that enables or provides services to user applications. Types of middleware can include databases, interpretive language run-time code (like a Java™ Virtual Machine), user space network stacks, multimedia software, and so forth.

About Intellectual Property

Even if the definition of open source advocates source code redistribution for the sake of the community, there are several cases where this cannot be done or may not be desirable due to competitive or otherwise sensitive information. Companies simply don't want to expose what is the heart of their competitive advantage: in a word, their IP.

The only way to protect intellectual property when dealing with open source software is to follow a rigorous process during the development of proprietary code in order to isolate sensitive components from others that are less sensitive, and from others that may infect the whole product. Here is what such a process should include:

- Identify which components constitute the essence of the IP that you most want to keep as proprietary (the “crown jewels”).
- Forbid the improper use of “GPLed” or viral software, and the use of any other open source software that is subject to any other license terms that could require unrestricted delivery or disclosure of the source code for the IP you want to protect.
- Isolate as much as possible the open source components used in the development phase (e.g. keep proprietary components in a common location in the source tree, and open source components in another common location in the source tree).

Several companies, such as Blackduck Software⁸, offer solutions to help determine if source code includes parts of existing open source projects. Such solutions represent an additional way to ensure that IP will not suffer from a contamination.

Developing Software on Linux

User Space Freedom

Most of the software developed on a Linux based platform targets an execution in the user space. Therefore, to avoid GPL contamination, the following rules must be observed:

- Do not include GPL code
- Do not include code that derives from preexisting “GPLed” programs
- Do not link to GPL libraries or other GPL program code

However, as stated earlier, simply invoking system routines (making system calls) will not itself result in contamination.

Kernel Modules (GPL)

Development that must be done in the kernel space (e.g., device drivers) must take into account the following: Linux offers the option of loading

extensions into the kernel after boot-time. This post-boot time loading of extensions will cause such extensions to be contaminated by the GPL. Any time code is embedded in a dynamically loaded module, it must link with the kernel's libraries and will not access them via ordinary system calls.

Nevertheless, some hardware companies, such as video card manufacturers, already have distributed binary-only versions of their drivers by either:

- Infringing the GPL license by loading their driver at boot time
- Creating a GPL component in charge of dynamically loading its own proprietary code

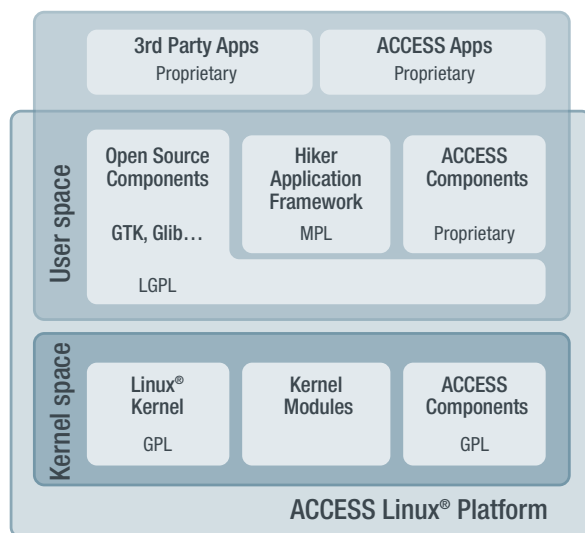


Figure 2: ACCESS Linux Platform™ & Licenses

These methods are not in line with the definition of open source software and have raised several stormy discussions, recently tempered by Linus Torvalds⁹.

Source Code Redistribution

In the case where your software inherits from GPL or LGPL (falls under the scope of the GPL or LGPL terms), here are the three options for distributing the source code:

- Direct supply with the binary distribution
- Offer to supply upon request
- In limited circumstances, point the user back to the party where the binaries came from

ACCESS Linux Platform™

Ensuring the protection of intellectual property is a core concern of ACCESS. This concern is evident in the design of ACCESS Linux Platform, which provides an environment of:

- Trust, as particular attention has been paid to avoid unanticipated exposure to third party intellectual property dependencies.
- Choice, as the platform allows users to create either open source or proprietary software on top of the platform, as they desire.

Application Framework

ACCESS, with its experience in the creation and maintenance of a software ecosystem, has designed ACCESS Linux Platform to address developer and licensee needs. To support this ecosystem ACCESS used the Mozilla Public License to create an application framework, called Hiker Application Framework™, that allows proprietary code to run on top of the platform. In addition, the middleware frameworks, running in the user space, provide the choice of creating either open source or proprietary

software on top of the platform. Error! Reference source not found.shows how the different ACCESS Linux Platform areas are exposed to the OSI licenses. It emphasizes the statement made earlier on the non-contamination of the user space by “GPLed“ components running in the kernel space. Taking into account the license constraints listed previously, and knowing that it is composed of proprietary, “MPLed”, “LGPLed” and “GPLed”

components, ACCESS Linux Platform provides a concrete solution for protecting IP and developing business at all levels, from end user applications down to the mobile handset hardware. By pairing its embedded systems know-how with the Linux world, and by enabling business solutions to be developed on top of Linux, ACCESS Linux Platform provides more control and autonomy than any other existing solution.



About ACCESS CO., LTD.

ACCESS CO., LTD., is a global company providing leading technology, software products and platforms for Web browsing, mobile phones, wireless handhelds and other networked devices. ACCESS' product portfolio, including its NetFront(tm) Browser, Garnet(tm) OS (formerly Palm OS(r)) and ACCESS Linux Platform(tm), provides customers with solutions that enable faster time to market, flexibility and customizability. The Company, headquartered in Tokyo, Japan, operates 29 subsidiaries and affiliates within Asia, Europe and the United States. ACCESS is listed on the Tokyo Stock Exchange Mothers Index under the number 4813. For more information about ACCESS, please visit <http://www.access-company.com/>.

Copyright © 2007, ACCESS Systems Americas, Inc. ACCESS, ACCESS Linux Platform and certain other trade names, trademarks and logos are trademarks which may be registered in the United States, France, Germany, Japan, the United Kingdom, and other countries and are owned by ACCESS Systems Americas, Inc. or its affiliates. These marks may not be used in connection with any product or service that does not belong to ACCESS Systems Americas, Inc. or its Affiliates (except as expressly permitted by a license with ACCESS Systems Americas, Inc.), in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits ACCESS Systems Americas, Inc., or its subsidiaries or affiliates. All other brands and trademarks used herein are or may be trademarks of, and are used to identify other products or services of, their respective owners. All rights reserved.

The registered trademark LINUX® is owned by Linus Torvalds, owner of the mark in the U.S. and other countries, and licensed exclusively to the Linux Mark Institute, from whom ACCESS obtained its non-exclusive license to the mark.

Java is a trademark of Sun Microsystems, Inc.

Mozilla is a registered trademark of the Mozilla Foundation.