

Government Computer Software Acquisition and The GNU General Public License

B. Scott Michel, Lt. Cmdr., PhD, USN(RC)
Eben Moglen, Software Freedom Law Center
Mishi Choudhary, Software Freedom Law Center
Dorothy Becker, Navy OGC, SPD Patent Counsel

October 1, 2011

Disclaimer: The contents of this paper represent the views and opinions of the authors only. It is not intended to represent or convey Department of Navy or Department of Defense policy.

1 Executive Summary

Open Source Software (OSS) is now integral to many software development efforts, so much so, that two “Frequently Asked Questions” (FAQs) resources [1], [2] exist to provide software acquisition professionals with information dispelling misconceptions and encouraging OSS adoption. These FAQ resources do not provide guidance related to how these licenses interact with the Department of Defense Federal Acquisition Regulation Supplement (DFARS). This white paper discusses one of the commonly used OSS licenses, the GNU General Public License (GPL), and how the GPL can be successfully used within DFARS software acquisitions. In the wider DoD Open Architecture context, wider adoption of the GPL fits well with developing a more competitive software acquisition landscape.

The major points discussed in this white paper include:

- GPL software is commercial software in accordance with DFARS 252.227-7014(a)(1), “*Rights in Noncommercial Computer Software and Noncommercial Computer Software Documentation*”.
- The GPL is a “copyleft” license, i.e., a license that enables end users to freely inspect, modify and redistribute software, specifically, software executables. When copies of a software

executable are distributed or conveyed, the complete source code must also be distributed or include an offer to make it available such that the executable can be recreated in its entirety.

- The GPL does not require the source code to modified versions to be returned to the original software developer(s) or re-integrated with the original development effort.
 - The GPL allows serial, private modification to GPL-licensed source code, i.e., modification without redistribution to the public or modification solely distributed “inside” government. “Inside” government distributions are analogous to distributions between a corporation’s departments or divisions. The GPL’s source code distribution terms are only applicable when the modified executable is distributed to the public or “outside” the government.
 - Modifications are often re-contributed for a variety of reasons, resulting in a vibrant and innovative open source software ecosystem.
- The GPL provides software license rights that closely resemble DFARS unlimited rights. The GPL may also be applied to software acquired under the DFARS government purpose rights license.
 - Before the government purpose rights have converted into unlimited rights, executable and source code software distribution must be made “inside” the government due to the limitations of the DFARS government purpose data rights license. Distribution may be made “outside” the government so long as the government purpose software distribution is accompanied by a nondisclosure agreement (NDA). Government purpose rights generally convert to unlimited rights five years after the date of contract award.
 - Distribution of the software modifications can be made “outside” the government when the government has unlimited rights in the software.
 - An intellectual property (IP) attorney should be consulted when questions arise as to what type of distribution is appropriate.
- Distribution of GPL software source code may be subject to classification levels and other legal limitations such as ITAR, export control and distribution statements. GPL software development within classified programs is a private modification and the resulting executables and source code may only be redistributed to individuals and contractors with the appropriate access.
- For the purposes of bid and evaluation as provided in the DFARS, GPL software will be provided to proposers under the GPL’s licensing terms. This distribution is considered an “outside” government distribution and appropriate caution should be exercised. Distribution sensitivities should be considered prior to the release of the request for proposal and appropriate safeguards utilized (i.e., ITAR, export control, security classifications or distribution statements).

2 Introduction

Open Source Software (OSS) has been a driver of software innovation over the last twenty years. OSS is pervasive — today’s commercial software is likely to have at least one, if not more, OSS-developed element or component embedded within it. OSS has become the foundation for several successful business models, the most familiar of which are Google, RedHat Systems and the Android® family of smart phones. It also has reduced the barrier to entry into today’s web service-dominated, on-line economy through the so-called *LAMP* stack: Linux®, Apache, MySQL® and Perl, each of which is OSS software. It is not surprising that OSS has become integral to today’s software-intensive systems within the Department of Defense’s acquisition process, although some care is required to make effective use of OSS systems and technologies.

OSS software development can generally be categorized along the lines of two broad (sometimes overlapping) communities: the “free software” community and “open source” community. The two communities have similar goals and objectives, but differ philosophically: the free software community views source code inspection, modification and distribution as fundamental rights conferred upon end users, whereas the open source community views source code as a tool within a larger software development methodology. This difference is easily seen in the way the respective communities license their software. The free software community espouses the GNU General Public License (GPL), which requires the source code to accompany its respective executable when conveyed or distributed to an end user. By contrast, the open source community merely requires that source code be accessible and attributions to authors be retained, but does not tightly bind the source code’s distribution to the end user executable’s distribution.

The GPL has not been widely adopted as a software license in DFARS acquisitions. This lack of adoption is mostly the result of popular misconceptions related to the license’s “copyleft” terminology, “viral” software licensing, and the free software development workflow.

- *Copyleft*: The notion of “copyleft” is a play on words intended to contrast with traditional copyrights. The copyleft allows freedom to redistribute, freedom to inspect and freedom to modify, whereas a copyright limits distribution, inspection and modification. Moreover, the copyleft ensures that no one can infringe upon these three freedoms. This does not mean that the copyleft replaces the notion of copyrights; copyright law still applies to enforcing the GPL and the copyleft.
- *Viral software licensing*: One of the largest barriers to GPL adoption in DFARS acquisitions is the notion that the GPL is “viral” because it encompasses more than the licensed executable. GPL software is distributed in a state of completeness: it covers the source code, scripts, libraries and support code required to rebuild or reconstruct the licensed executable.

This “viral” notion is attributed to Craig Mundie, currently Microsoft Corporation’s Chief Research and Strategy Officer, who asserted that software libraries used in a GPL executable must also be licensed under the GPL or a GPL-compatible license [3]. This assertion attempts to cast the GPL’s source code completeness requirement in an unfavorable light, while

simultaneously diminishing software licensing consistency when components are integrated into an executable (i.e., climbing up the right hand side of the systems engineering “V”.)

This notion may also stem from and be reinforced by GPL software’s widespread availability, rapid evolution and development. To program managers accustomed to a formal, process-oriented software development model, rapid evolution and development could appear to be chaotic and “out of control”. However, the speed and range of development possible with GPL software demonstrably provides very tangible economic and societal benefits.

- *Software development workflow*: There is a pervasive misunderstanding with respect to free software and open source development workflow, namely, that modifications must be returned or re-integrated with the original developer(s) or development efforts. This point is discussed in more detail below in GPL Software Development Workflow. Suffice it to say that the GPL protects and encourages private modification, which does not require those modifications to be returned or re-integrated. This point is particularly important to remember when using GPL software within classified programs.

The remainder of this white paper provides an overview of the GPL’s salient features and how those features relate to DFARS software acquisitions. The GNU General Public License presents the GPL’s features, the different versions of the GPL and GPL software development workflow. GPL Software Systems and Government Contracting discusses how the GPL can be successfully applied to DFARS software acquisitions, including classified programs. These two sections lay the foundation upon which program managers and associated acquisitions offices can confidently include the GPL license and GPL software as part of their software acquisition strategies.

3 The GNU General Public License

The GNU General Public License (GPL) represents one familial branch of the OSS license tree that is explicitly designed to promote software distribution for the purposes of inspection, modification and protection of general free expression by free software developers¹. The GPL enables and explicitly requires original and modified source code distribution when an executable is conveyed from one party to another. If the application’s source code is not conveyed with its executable, an offer to make the source code available must accompany the executable. Conveying an executable does not have to be a commercial transaction; mere redistribution is sufficient to trigger the GPL’s licensing provisions.

Ordinarily, users take little interest in their application’s source code. For example, users are more interested in producing documents using their favorite word processor than understanding its

¹The other familial branch of the OSS license tree is represented by the MIT and UC Berkeley style of open source licenses. These licenses allow use and modification of source code and require attribution acknowledgements, but do not have the copyleft provision that tightly binds source code to executables when executables are distributed.

internal operation. In the case of government software acquisition and related research and development efforts, having the application's source code enables competitive and operative flexibility for the cognizant government program offices. The source code to an original, fully functional work is the foundation for system evolution through future modifications. Future modifications are not tied to any particular software developer because GPL breaks the link between the software developer and the source code's perceived value as intellectual property. The value proposition no longer resides in the ownership of the code. Instead, the value lies in the ability to freely modify the source code. As the software evolves, each modification must be provided along with the original source code such that an independent software developer, acting alone, can completely reconstitute the software as it was delivered by the last modifying developer.

There are currently two versions of the GNU General Public License: GPL version 2 ("GPLv2") and GPL version 3 ("GPLv3"). Both GPLv2 and GPLv3 are strong copyleft licenses. "Strong copyleft" means that source code absolutely must accompany its respective executable. The primary differences between GPLv2 and GPLv3 relate to software patents, compatibility with other free software licenses and hardware-imposed restrictions that prevent software modification. For example, the GPLv3 requires any party distributing or modifying a work to provide a patent license to all claims that would otherwise be infringed. The GPLv3 also does not allow hardware manufacturers to implement mechanisms that prevent modified GPL software from executing, a circumstance not previously envisioned by the GPLv2.

Examples of software that use these two GPL versions include:

- GPLv2
 - The Linux operating system's kernel
 - The Oracle[®] MySQL relational database
 - Wordpress blogging software
- GPLv3
 - The SugarCRM[®] Open Source Business and Social Customer Relationship Management (CRM) software
 - The SAMBA file and print services
 - The GNU Compiler Collection (GCC)

3.1 The GNU Lesser General Public License

The GNU Lesser General Public License ("LGPL") is a weak copyleft license primarily intended for utility libraries. The LGPL does not automatically require an encompassing executable's source code to be published as the result of using LGPL software or a LGPL component. The LGPL only applies to the software component to which the LGPL is attached. Consequently, software systems

licensed under the LGPL can be aggregated or contained within proprietary software. For example, LGPL components in the Linux operating system include core system and runtime libraries such as the C Runtime Library (*libc*) and the C math library (*libm*). As a result, proprietary software can execute on GPL systems, such as Linux, without automatically becoming subject to the full copyleft.

Choosing between the GPL and the LGPL in a new software acquisition effort has long term implications on future program developments and modification. If the goal is to level the playing field for increased future competition, then the GPL is the preferred license. The LGPL increases future risk because non-LGPL portions of the software's source code are not required to be delivered with the executable. If the contractor delivers code under the LGPL license, that code is, by definition, incomplete, which will complicate future modification efforts (i.e., increase future software acquisition costs through vendor lock-in.) It is possible to relicense LGPL software as GPL software, in order to convert from a "weak" to a "strong" copyleft license.

Program managers need to carefully evaluate whether the LGPL license is applicable to their software acquisition strategy and whether a strong copyleft (e.g., the GPL), or a copyleft-compatible license (e.g., the Apache License 2.0) is a better choice for current and future program needs. The Free Software Foundation offers helpful information with regard to choosing between various copyleft and copyleft-compatible licenses, found in [4].

3.2 GPL Software Development Workflow

A frequently asked question regarding open source and free software modifications is, "Do modifications need to be sent back to the original developer(s) or project?" The basic premise of OSS and GPL software development is collaboration within a group of like-minded software developers. Within these self-organizing software development efforts, the reward system is based on receiving recognition or adulation for contributions and for improving the quality of OSS. This reward system ensures that modifications are contributed back to the original software development effort. As a result, the OSS software development ecosystem takes on a vibrant life of its own, fostering and inspiring innovation and creativity.

Just as importantly, employment and increased commercial opportunities also motivate re-contribution of modifications. In a sense, modifications that are integrated into new releases, such as significant enhancements or fixes to critical bugs, are good indicators of a software developer's skill level and, as such, can be used to assess the software developer's potential value as an employee. From a commercial perspective, the *LAMP* software stack has served as both an engine of employment and an engine of commerce. This commercial success is not only due to the low cost of the software (frequently, only the cost to download and install the software) but also due to the contributed enhancements, innovations and packaging. Consequently, modification re-contribution is a facet of a virtuous cycle.

When GPL software is modified, there is an "upstream" and a "downstream" software developer, relative to the direction in which source code contributions flow, as illustrated in the "Devel-

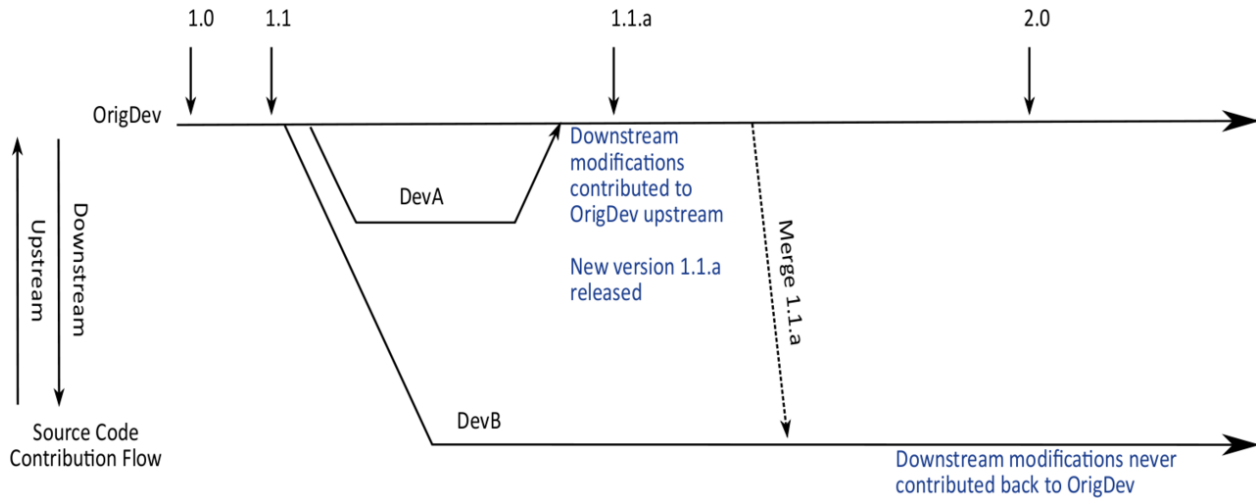


Figure 1: Development Workflow

opment Workflow” figure. The upstream developer, *OrigDev*, provides the source code that is subsequently modified by two downstream developers, *DevA* and *DevB*. Both *DevA* and *DevB* receive release 1.1 of *OrigDev*’s source code and embark on separate modification efforts. Only *DevA* eventually contributes their modifications back to *OrigDev*, which are subsequently re-integrated into *OrigDev*’s source code and released as a new version, 1.1.a. Customarily, modifications are passed from a downstream to an upstream software developer in the form of patches². Whether the original source and patches are conveyed or a new, integrated version of the source code is conveyed, the GPL requires both *OrigDev* and *DevA* to convey the source code to version 1.1.a to any party to whom version 1.1.a is distributed or conveyed. In the strictest sense of the GPL, *DevA* is responsible for conveying version 1.1’s source code and *DevA*’s modifications even though the same source code is available from *OrigDev* in version 1.1.a.

In the above workflow scenario, *DevB* chooses not re-integrate with or send any of their downstream modifications back to *OrigDev*. In open source and GPL parlance, *DevB* “forks” from *OrigDev*’s version 1.1 and creates an independent software development effort. *DevB* can also keep up with the changes and releases distributed by *OrigDev* and *DevA*, as shown in the figure when *DevB* merges *OrigDev*’s 1.1.a version’s changes. Even after becoming synchronized with version 1.1.a, *DevB* is still not obligated to re-contribute or re-integrate their changes into *OrigDev* or *DevA*’s development efforts. *The GPL contains no specific language stating, implying or requiring the downstream developer to send modifications to an upstream developer.* The GPL still requires *DevB* to convey the original version 1.1 source code and all subsequent modifications when *DevB* publicly conveys or redistributes an executable.

²Patches are files that contain the differences (i.e., lines added, updated, deleted) between the original and modified versions. “Patch” and “modification” are often used interchangeably.

4 GPL Software Systems and Government Contracting

DFARS defines two categories of software: commercial and non-commercial. OSS in general, and GPL software in particular, meets the DFARS definition of “commercial software”³ because it is licensed to the public. Thus, GPL software may be included as part of a government acquisition effort under DFARS⁴.

Most commercial software licenses are restrictive, limiting copying and modification, and prohibiting further distribution. But OSS is different from all other commercial software because the terms on which it is licensed to the public are terms nearly or completely corresponding to “unlimited rights” as that term is used in acquisition regulations. All OSS licenses provide that the party receiving the software may freely use, copy, modify and distribute unmodified and modified versions of the software without license fees, time limits, waiting periods, or other restrictions. “Copyleft” OSS licenses, of which GPL and LGPL are the leading examples, allow unlimited distribution but require that, when distribution occurs, the copies are distributed with GPL license rights. The GPL license also requires that source code, or an offer of access to source code, must accompany any distribution of an executable copy of the original work and its modifications.

4.1 Acquisition Efforts Under DFARS

As a commercial item, GPL software is compatible with DFARS acquisition efforts. Such software and associated source code can be provided to contractors and be freely modified for use by the government. Once a contractor has prepared a modified version of GPL software, that modified version must be delivered to the contracting program with GPL rights in accordance with the GPL license’s terms. To ensure the delivery of the modified software with appropriate rights markings, the government contract should include a clause requiring GPL markings along with a CDRL⁵ item

³DFARS 252.227-7014(a)(1), “Rights in Noncommercial Computer Software and Noncommercial Computer Software Documentation” (MAR 2011), provides that “commercial computer software” means software developed or regularly used for non-governmental purposes which—

- (i) Has been sold, leased, or licensed to the public;
- (ii) Has been offered for sale, lease, or license to the public;
- (iii) Has not been offered, sold, leased, or licensed to the public but will be available for commercial sale, lease, or license in time to satisfy the delivery requirements of this contract; or
- (iv) Satisfies a criterion expressed in paragraph (a)(1)(i), (ii), or (iii) of this clause and would require only minor modification to meet the requirements of this contract.

Similar language defining a “commercial item” is found in 7 USC, section 403(12).

⁴The DFARS tailors the provisions of the *Federal Acquisition Regulation* (FAR) to the Department of Defense’s acquisition efforts. Other government agencies have similarly tailored versions of the FAR, such as the Department of Energy Acquisition Regulation (DEAR). Comments made in this paper may not apply to non-DoD acquisition efforts as the result of differences between these agency-specific supplements. When questions arise, consult an IP attorney.

⁵Contract Data Requirements List

requiring delivery of the GPL executable and source code. Contractors may attempt to deliver the modified version with DFARS data license rights rather than with GPL data license rights. But even if the contract pursuant to DFARS would permit such delivery, the contractor would be infringing the copyrights of the original authors of the GPL work by delivering copies labeled “unlimited rights” or “government purpose rights,” because the GPL requires all modified versions to be distributed with GPL markings. Therefore, GPL is the only set of terms under which contractors can legally deliver modified versions of GPL code. Note that although the contractor may have violated the GPL’s terms by delivering incorrectly marked GPL software to the government, the license provides that this upstream violation does not affect the government’s compliance if the government does not itself violate the GPL’s terms.

The capability to modify inappropriate markings is provided by DFARS policy and executed via DFARS contract clause language. DFARS 227.7203-12, “*Government’s rights to establish conformity of markings*”, defines unjustified (i.e. inappropriate) markings as markings that inaccurately restrict the government’s rights to use the marked data. It also allows the government and contractor to agree to correct or strike the inaccurate markings. In the absence of such agreement, the government can verify the accuracy of the markings by requesting the contractor to provide information substantiating the markings (see DFARS 227.7203-13, “*Government’s right to review, verify, challenge, and validate asserted restrictions*”). If the provided information is insufficient, the government can also challenge or deny the markings as necessary, but this is a more complex process. In order to exercise this capability, DFARS 252.227-7014, “*Rights in Noncommercial Computer Software and Noncommercial Computer Software Documentation*”, and DFARS 252.227-7019, “*Validation of Asserted Restrictions—Computer Software*”, must be included in the acquisition contract. DFARS 252.227-7014(h) allows the government to ignore, correct, or strike a marking that is determined to be unjustified, under the provisions of DFARS 252.227-7019. An IP attorney should be consulted if there are questions about unjustified or inaccurate markings. Note that these DFARS provisions relate solely to markings for computer software. Different DFARS provisions apply to computer software documentation markings because computer software documentation is considered to be technical data. Accordingly, DFARS sections 227.7103-12, “*Government right to establish conformity of markings*”, and DFARS 227.7103-13, “*Government right to review, verify, challenge and validate asserted restrictions*”, are the applicable policy provisions and DFARS sections 252.227-7013, “*Rights in Technical Data—Noncommercial Items*”, and DFARS 252.227-7037, “*Validation of Restrictive Markings on Technical Data*”, are the clauses that must be included in the acquisition contract when software documentation or technical data is involved.

If the government receives a modified version of a GPL work that is unmarked (i.e., delivered with unlimited rights), the government may be justified in adding the appropriate GPL markings in order to maintain compliance with the letter and intent of the GPL. If the government receives a modified version of a GPL work, where the contractor has insisted on labeling the modified version as “Government Purpose Rights,” the government is justified in correcting the incorrect markings as provided for in the DFARS to conform to the required GPL markings. As is the case with any other commercial or noncommercial software, the government program office has an obligation to protect IP rights by ensuring that all received software is appropriately marked and

that any software copies provided by the program office are also appropriately marked.

Once such modified software has been delivered to the government, its GPL rights allow the modified code to be the subject of additional contracts for modification by contractors. Each contractor who successively modifies the GPL software receives the source code under the GPL and delivers the next modified software version under the same GPL terms. As is usual with DFARS acquisitions, the contractor ordinarily retains ownership of copyright in the modifications made to the GPL software. These ownership rights subsequently allow the contractor to further modify and distribute copies of the executable and source code.

4.2 GPL and DFARS Data Rights Software

GPL software can also be modified by combining it with existing government-funded software if the government possesses sufficient rights in the existing software. The DFARS provides four types of rights that relate to computer software: unlimited rights, government purpose rights, restricted rights, and specially negotiated rights. The government's right to modify software is provided without restriction when unlimited rights exist. Unlimited rights generally exist when software copies have been provided to the government without any data rights legends or other restrictive markings.

When software bears a government purpose rights legend, the DFARS provides that the government only has the right to provide the software to a contractor for modification when it is accompanied by a non-disclosure agreement (NDA). The NDA is only required while government purpose rights are in effect, which is generally five years in duration from contract award. When the government purpose rights time period expires, the rights convert into unlimited rights and a NDA is no longer required. When the software is marked with a government purpose rights legend, the contract should also contain language specifying that the modified software is for the government's exclusive use in order to remain compliant with GPL license terms.

If the software to be modified is marked with a restricted rights legend, written permission from the software owner is required before any modifications can be made. A special license may need to be negotiated in order to deal with unusual, unique or complex situations. If a specially negotiated license exists, it will be attached to the contract file. Consult an intellectual property (IP) attorney or appropriate legal authority when questions about the interaction between the GPL and DFARS arise.

5 Classified Modification of GPL Software

Copyright law contains rules defining property interests, protections which apply to both GPL and DFARS software. Infringement of copyright is a misuse of property, compensable by payment of

damages⁶. Within classified programs, security law establishes additional obligations that limit access to and distribution of the software. The fact that the GPL does not relieve anyone of the obligations imposed by security law, while maintaining the requirement to provide the source code, scripts, support code and libraries when GPL software is distributed or conveyed results in an apparent contradiction between security and GPL requirements. Fortunately, Section 7 of the GPLv2 and Section 12 of the GPLv3 resolve this seeming contradiction. These sections state that if a covered work (i.e. modified GPL software) cannot be distributed or conveyed so as to simultaneously satisfy the GPL obligations and the outside (i.e. security) requirements, then the modified GPL software may not be distributed or conveyed. This means that GPL software may be developed and serially modified in classified programs without triggering the requirement to distribute the software outside the classified programs, thereby satisfying both security and GPL obligations.

Classified programs rely on the “need to know” and the “need to share” as the protection boundaries erected around the acquisition activity. Within these protection boundaries, there is an entity authorized by law to receive computer software from an entity authorized by law to deliver computer software. GPL-ed executables and the accompanying source code should only be offered to an authorized receiving entity by an authorized delivering entity. Thus, an authorized contractor may receive classified source code from a government program office in order to make the desired modifications. Modifications to classified software by authorized entities is considered an exercise of the freedom to privately modify under the terms of the GPL. For clarity, private modification is modification without redistribution to the public or modification solely distributed “inside” government (see following section). Accordingly, in such situations, the GPL allows the contractor to modify the GPL software and to deliver the modified version of the classified code to the government (or any other entity having appropriate security access) without having any rights to further modify the code or any requirement to distribute the code outside the terms of the contract. With the protection provided by the GPL’s freedom to privately modify software, GPL software can safely and legally be used within classified program efforts. This principle is also relevant in some situations concerning requests for source code by non-U.S. parties not legally authorized to receive source code, such when ITAR, export control laws and distribution statements apply.

⁶Copyright law also provides for other penalties, including injunctions, costs and attorney fees, impounding of infringing items and criminal punishment. For additional details, refer to the “Copyright and Infringement Penalties” (sections 501–513) of Title 17 of the U.S. Code. Furthermore, copyrights on open source software are legally protectable; see *Jacobsen v. Katzer*, 535 F.3d 1373 (Fed. Cir. 2008), in which the Circuit Court of Appeals held that open source copyrights are enforceable.

6 Distribution and Conveying of GPL Code (“Inside” vs. “Outside” Government)

As was noted earlier, the GPL license requirement to provide source code, or to make it available, is not triggered until an executable is delivered to the the contracting program office. The trigger for this requirement varies according to the GPL license version that was applied to the software, notably, the GPL version 2 (GPLv2) or the GPL version 3 (GPLv3).

GPLv2 regulates the “distribution” of computer programs, using the concept of “distribution” defined by U.S. copyright law. U.S. federal courts have interpreted copyright law to state that “distribution” is synonymous with “publication”. “Publication” is defined by the Copyright Act as “the distribution of copies or phonorecords of a work to the public by sale or other transfer of ownership, or by rental, lease, or lending.”⁷ The courts have held that activity to copy and distribute copyrighted works within a single organization is not “to the public”. In a corporation or enterprise organized into divisions, movement of copyrighted material from one division to another is also not transfer “to the public.”

For this reason, the Free Software Foundation, which authors the GPL and which speaks authoritatively on its intention, has taken the position in the past, and continues to believe, that copying and transmission of GPL software within the U.S. government, including inter-agency transmission, is not “publication” (i.e., distribution), requiring compliance with the provisions of GPLv2. In other words, this distribution “inside” the government does not trigger the requirement to provide source code when an executable is delivered.

GPLv3 differs from its predecessor, GPLv2, in that it removes the U.S.-specific language of “distribution.” GPLv3 instead uses the term “conveying” to describe the act triggering GPL conditions. Only actions that transmit a copy of a program *and* require copyright license under local law are “conveying.” Because intra-governmental distribution is not “publication” under U.S. copyright law, it is not “conveying” under GPLv3’s terms. Again, this “inside” distribution does not trigger the requirement to provide source code.

Therefore, software licensed under GPLv2 or GPLv3 can be transmitted among and between agencies of the U.S. government without limitation by GPL licensees. GPLv3 software transferred to a contractor is generally considered to be an “outside” government distribution. Exceptions to this “outside” distribution include software contractually limited to the exclusive use of the U.S. government or running on a third party data center exclusively for government purposes. These exceptions, which are considered to be “inside” government distributions, may be effected or established by adding to the contract a clause specifying such exclusive government use or use by others exclusively for government purposes. In the case of GPLv2, all distributions to contractors are considered “outside” distributions, without exception. Care should be taken to determine which GPL license version applies; consult an attorney if there is any question regarding “inside” versus “outside” government distributions.

⁷Refer to 17 U.S.C. sect. 101.

The distinction between “inside” and “outside” government is particularly important when GPL software is combined with software to which DFARS data license rights apply, because the government’s rights vary depending on the different data rights licenses. The government can provide copies “outside” the government without a problem when it has unlimited rights in the software. But, when the government combines government purpose rights software with GPL software, delivery or conveyance of the resulting modified software is limited. While the government can deliver the modified software at any time “inside” the government, a NDA must be in place if delivery is to be made “outside” the government — at least until the government purpose rights period has expired (generally, five years after contract award.) At that time government purpose rights convert into unlimited rights and delivery “inside” and “outside” the government is freely allowed. A specially negotiated license may also provide the right to distribute or convey “outside” the government. For restricted rights, however, written permission from the software owner is always required. Do not hesitate to consult an IP attorney if questions arise as to what behavior is appropriate or allowable.

7 Evaluation for the Purposes of Bid and Proposal

Under the DFARS, proposers receive software from the government and the proposers’ future actions are constrained by the rights attached to that software. The permissiveness of the GPL enables the government to distribute GPL software to the proposers during the bid and proposal process. This distribution is an “outside” government distribution, so, before government-funded software is provided to a proposer, the government must ensure that it has sufficient rights to do so.

An “outside” distribution of GPL software permits a proposer to independently make modifications and a derivative work. These independent modifications entitle the proposer to redistribute the newly created derivative work under the conditions imposed by the GPL. However, the derivative work could be redistributed to entities other than the government. If redistribution sensitivities exist with respect to GPL software, e.g., an association between a software component and an integrated weapons system, then the issued RFP should employ the necessary safeguards and restrictions as would ordinarily be done with other software works and contracting activities. These safeguards include ITAR, export controls, security classification levels or distribution statements, as appropriate.

When the government receives solicited and unsolicited proposals that contain GPL software, these proposals, associated software and software licenses need to be subjected to the same source selection process and software acquisition strategy evaluation. This does not differ from the normal practices established and conducted under the DFARS. While GPL software may generally be more desirable from an intellectual standpoint, GPL software must still be analyzed along with competitive proprietary software in terms of overall value to the government.

8 Conclusion

The GNU General Public License is compatible with the DFARS and most closely resembles unlimited rights licensing. This resemblance arises from the unique feature of the GPL, the “copyleft”: when an executable is delivered, the complete source code, including the modifications, must be delivered or made available. GPL-licensed software has not been widely adopted within DFARS acquisition efforts due to uneasiness caused by numerous misperceptions surrounding the copyleft. While it is true that the GPL protects the freedom of a software developer to distribute or convey source code, distribution may be restricted to authorized entities when GPL software is developed or modified within classified programs. In fact, an important purpose of the GPL is to promote and protect the right of serial private modification, the development model of free and open source software. More importantly, serial modification does not imply or require that subsequent modifications be contributed back to the original software developer or development effort.

GPL software falls under the DFARS definition of commercial software. As such, it can be used in DFARS acquisition efforts. Source code distribution is only triggered when the executable is delivered or conveyed by the contractor to the receiving government program office. With respect to DFARS acquisition efforts, when these GPL terms are triggered depends on whether the distribution is made “inside” or “outside” government. Acquisition efforts can also use GPL software in combination with other government-funded software, although the government’s rights to modify and distribute the resulting aggregate software depend upon the DFARS data license rights attached to the government-funded software. Consult an IP attorney or appropriate legal authority when questions about the interaction between the GPL and DFARS arise.

Program offices can confidently take delivery of, use, modify and distribute GPL software, either alone or in combination with government-funded software. But before selecting GPL software for use in an acquisition effort, make sure that existing practices, from bid and proposal through operations and maintenance, are followed. GPL software offers many benefits, social and economic, and should be evaluated in the same manner as other commercial software or software that has been developed using government funding. The ultimate question to be answered is what is best for the program’s objectives and results in the best value to the government.

9 Acknowledgements and Notes

The DoN authors gratefully acknowledge the support and encouragement from Mr. Nickolas Guertin (DASN/RDT&E), Ms. Jane Barrow (NAVSEA SEA 00L) and Dr. Craig M. Arndt (Defense Acquisition University).

This white paper was written with *Pandoc* [5], a multi-format text processing tool written in Haskell. Lt. Cmdr. Michel implemented several *Pandoc* feature enhancements, such as better support for on-line (web page) bibliographic references in the *citeproc-hs* bibliography formatter,

URL hyphenation and better support for internal document links in the *ConTeXt* and *LaTeX* output formats⁸. In keeping with the spirit of free and open source software, patches were re-contributed to the *Pandoc* and *citeproc-hs* software development efforts.

10 References

- [1] ASD/NII, *DoD Open Source Software (OSS) FAQ*. [Online]. [http://cio-nii.defense.gov/sites/oss/Open_Source_Software_\(OSS\)_FAQ.htm](http://cio-nii.defense.gov/sites/oss/Open_Source_Software_(OSS)_FAQ.htm).
- [2] CENDI, *Frequently Asked Questions about Copyright and Computer Software Issues Affecting the U.S. Government with Special Emphasis on Open Source Software*, 2009. [Online]. http://cendi.gov/publications/09-1FAQ_OpenSourceSoftware_FINAL_110109.pdf.
- [3] C. Mundie, *Prepared Text of Remarks, "The Commercial Software Model" to The New York University Stern School of Business*, 2001. [Online]. <http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.msp>.
- [4] The Free Software Foundation, *How To Choose A License For Your Own Work*, 2011. [Online]. <http://www.gnu.org/licenses/license-recommendations.html>.
- [5] J. McFarlane, *Pandoc A Universal Document Converter*, 2011. [Online]. <http://johnmacfarlane.net/pandoc/>.

⁸If you are reading this in Adobe® Reader®, you are reading the result generated from the *LaTeX* output format.