

# Software Licence Agreements HSF Policy Guidelines

J. Harvey<sup>1</sup> M. Jouvin<sup>2</sup> A. McNab<sup>3</sup> E. Sexton-Kennedy<sup>4</sup> T. Wenaus<sup>5</sup>

<sup>1</sup>*CERN* <sup>2</sup>*Laboratoire de l'Accélérateur Linéaire (CNRS)* <sup>3</sup>*University of Manchester* <sup>4</sup>*Fermi National Accelerator Laboratory* <sup>5</sup>*Brookhaven National Laboratory*

## Abstract

These guidelines were prepared by the HSF Startup Team in order to serve as an aid in establishing a Software Licence Agreement for software projects hosted by the HSF. The report contains background information on open source licences approved by the Open Source Initiative and concludes with a set of recommendations for choosing a licence and instructions for drafting text to include with the source code.

# 1 Introduction

A simple survey of software packages in common usage in HEP reveals a de facto widespread adoption of different software licences. Moreover policy documents on software licensing in HEP are difficult to find. In 2011 CERN setup a Task Force to provide recommendations for the licensing of software developed at CERN. The final report [1] is worth reading as it contains much useful background information and reference material. The main recommendation states that, "Whenever possible, software owned in whole or in part by CERN should be made available as open-source software and that the open-source licences used for CERN-owned software should be widely used licences approved by the Open Source Initiative (OSI)". This approach is in broad agreement with the policies adopted at other laboratories and has therefore been taken as the starting point for the recommendations contained in this report.

The philosophy of openness is enshrined throughout our field as exemplified by making the results of our experimental and theoretical work generally available. The same approach is followed here in order to help achieve our goal of providing reliable and long-lived software products through collaborative open-source software development. Open Source Software (OSS) is computer software with its source code made available with a licence in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose. The goal of the guidelines described in this note is to allow HSF and its projects to distribute and build upon their respective work. In this regard, HSF follows the example of other leading open-source software endeavours, such as the Apache Software Foundation [2]. This does not preclude the full rights of contributors (copyright owners) to use their original contributions for any other purpose outside HSF.

## 2 Basic Terminology

We begin by defining some of the key terms as described in [3].

- *Copyright* is a legal right created by the law of a country that grants the creator of original work exclusive rights to its use and distribution, usually for a limited time. Copyright is a form of intellectual property, applicable to any expressed representation of a creative work. It is often shared among multiple authors, each of whom holds a set of rights to use or licence the work. These rights frequently include reproduction, control over derivative works, distribution, and "moral rights" such as attribution.
- *Public domain software* is software that has been placed in the public domain. In other words there is absolutely no ownership such as copyright, trademark, or patent. Unlike other classes of licences, there are no restrictions as to what can be done with the software. The software can be modified, distributed, or sold even without any attribution. This is the simplest way to make open source software and allows people to share the program and their improvements, if they are so minded, but it

also allows the program to be converted into proprietary software. They can make changes, many or few, and distribute the result as a proprietary product. People who receive the program in that modified form do not have the freedom that the original author gave them. A software licence is a legal instrument governing the use or redistribution of software. A typical software licence grants an end-user permission to use one or more copies of software in ways where such a use would otherwise potentially constitute copyright infringement of the software owner's exclusive rights under copyright law.

- An *open source software licence* is a notice that grants the recipient of a piece of software extensive rights to modify and redistribute that software. Copyright law usually prohibits these actions, but the rights-holder (usually the author) of a piece of software can remove these restrictions by accompanying the software with a software licence that grants the recipient these rights. Software using such a licence can meet the conditions to be classed as open source software as conferred by the copyright holder. Open source licenses broadly divide into free software licenses and permissive software licenses. Free software licences include "copyleft" provisions, which require all future versions to also be distributed with these freedoms. These are termed "restrictive" licences.
- *Permissive software licences* do not impose these additional conditions and are usually just a grant of rights and a disclaimer of warranty, thus also allowing distributors to add restrictions for further recipients, or to produce an extended proprietary version of the software. All major open source software licences require that acknowledgement is given to authors of the software in documentation and/or at runtime. In an academic context these provisions can be useful in establishing the impact of a software project, and even when software released under a permissive licence is reused in a closed-source commercial product.

## 3 Main License Types

### 3.1 Copyleft License

Copyleft is a general method for making a program free, and requiring all modified and extended versions of the program to be free as well. Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. Copyleft guarantees that every user has freedom and provides an incentive for other programmers to add to free software. A good example is the GNU C++ compiler.

The spirit behind a Copyleft licence is the creation of an open community of users or developers where the licencees are encouraged not only to improve, correct, complement and integrate the software they receive but also to make available these enhancements to the entire community. The difference between copyleft and non-copyleft licences is that users

cannot take the free software and turn it into proprietary software, thus preventing any member of this open community to depart from the principles of reciprocal contribution.

The Copyleft principles were laid down by Richard Stallman of the Free Software Foundation in 1985, which was at the inception of the OSS movement through the creation of the GNU project. Copyleft is a general concept, and therefore cannot be used directly; you can only use a specific implementation of the concept. In the GNU Project, for example, the specific distribution terms used for most software are contained in the GNU General Public Licence (GPL) [4]. GPL version 3 was published in 2007 but many copyleft projects (eg Linux) have chosen to continue with GPL version 2. GPL 2 and 3 software amount to 24% and 10% of open source software respectively [6].

### 3.2 Weak Copyleft License

These typically follow the same rules as the GPL except that the user may use, unmodified, the free software component in a larger program which is released under a licence different from the free licence. The chief consequence is that the user is not obliged to provide the full source code of its larger work under a copyleft licence.

The most widely used example of this type of licence is the GNU Lesser General Public Licence (LGPL). Licences such as LGPL target libraries of software, which are designed to be incorporated unchanged into larger programs. For example, the ROOT software project [5] has adopted an LGPL licence.

LGPL is also frequently used for non-library software when there is a particular concern from the licensor that the obligation to release the source of a work incorporating unchanged the GPL-licensed software would seriously hamper its wide adoption. The most common case is when a free library's features are readily available for proprietary software through other alternative libraries. In that case, the library cannot give free software any particular advantage, so it is better to use the LGPL for that library. The LGPL licence is used for the GNU C library, for example, since using the GPL would have driven proprietary software developers to use one of the many other C libraries.

As with full copyleft licences which prevent modified versions from being distributed under a proprietary licence, weak copyleft licenses are intended to ensure the non-appropriation by third parties of the Open Source software. As of July 2013, the LGPL was used by 7% of all open source licenced projects [6].

### 3.3 Permissive License

These licences allow redistribution of the original or modified software and source code, including under a different licence. Depending on the terms of the permissive licence, the different licences may be proprietary licences or copyleft licences or other permissive licences.

The Apache Software License (ASL), initially from 1999 and currently at version 2.0, is one of the most widely used examples of a permissive licence. Like other open source software licences, the licence allows the user of the software the freedom to use the software

for any purpose, to distribute it, to modify it, and to distribute modified versions of the software, under the terms of the licence, without concern for royalties. The Apache Licence does not require a derivative work of the software, or modifications to the original, to be distributed using the same licence (unlike copyleft licences). The Apache Software Foundation and the Free Software Foundation agreed that the Apache Licence 2.0 is a free software licence, compatible with version 3 of the GPL licence, meaning that code under GPL version 3 and Apache Licence 2.0 can be combined, as long as the resulting software is licensed under GPL version 3.

Other well-known examples of widely used free software licences approved by the OSI include the MIT and BSD licences. The MIT licences from 1988 onwards permit reuse within proprietary software provided all copies of the licensed software include a copy of the MIT Licence terms and the copyright notice. Such proprietary software retains its proprietary nature even though it incorporates software under the MIT Licence. The licence is also GPL-compatible, meaning that the GPL permits combination and redistribution with software that uses the MIT Licence.

BSD licences from 1988 onwards are another family of permissive free software licences, imposing minimal restrictions on the redistribution of covered software. Two variants of the licence, the New BSD Licence/Modified BSD Licence (3-clause), and the Simplified BSD Licence/FreeBSD Licence (2-clause) have been verified as GPL-compatible free software licences by the Free Software Foundation, and have been vetted as open source licences by the Open Source Initiative.

As of July 2013, the ASL, BSD and MIT permissive licences accounted for 42% of all open source licensed projects [6].

## 4 Specific Constraints

### 4.1 Licence Compatibility

The software stacks used in HEP experiments are generally composed of large numbers of software packages, with complex inter-dependencies on one another. The term licence compatibility refers to the problem of combining software programmes (such as the software packages in typical HEP stacks) when they are subject to different licences that may contain contradictory requirements. This issue and the related terminology are well described in the Annexes to the final report of the CERN Task Force on OSS Licences [1].

As compatibility is not necessarily reciprocal, compatibility is generally described as a directional property between two licences. One refers to upstream or downstream compatibility to specify the direction of the licence compatibility; when the compatibility direction is not specified, this is usually understood as equivalent to the upstream compatibility. For example, as mentioned above, the Apache v2.0 licence is (upstream-)compatible with GPL v3, and GPL v3 is downstream-compatible with Apache v2.0: it is possible for programmes licensed under GPL v3 to incorporate programmes licensed under Apache v2.0, with the resulting work still licensed under GPL v3, but the opposite is not true.

Licence compatibility is an important factor that the owners of a specific package should keep in mind when choosing the licence for their software. Downstream-compatibility, with the licences of any other programs upon which a given package depends, may effectively restrict the choice of the licence that can be adopted for that package, as some licences might be prohibited by compatibility issues. Amongst the licences allowed by downstream-compatibility, the most appropriate licence for a given package should be chosen taking into account its potential upstream-compatibility with the licences chosen by the prospective customers of that package: the choice of a particular licence for a package may in fact be an encouragement or a deterrent for other people to use that package within their software. These considerations are reflected in the HSF recommendations described in section 5.

## 4.2 Changing the License

The ability to change the license term of a project, including the right to dual-license it, is an exclusive right of copyright holders. Except when explicitly stated otherwise, copyright holders are all the people who contributed to the project. In large projects, after some time, it may make impossible to change the license used by a project.

Although this rule applies to any license, it is more a concern for copyleft licences as permissive licenses give anybody the right to fork the project with a new license. For this reason, some projects, when there is no risk (or a low risk) of appropriation of the work by a third party, prefer to use a permissive license in order to keep a greater flexibility to evolve (including restrict) the licence in the future. A well known example is Apache where a development community exists and where most people (including commercial vendors) contribute their modification back to the community even though this is not a legal requirement of the permissive Apache licence.

To avoid problems in changing licence, some projects or software foundations (like the Apache Software Foundation) have an explicit transfer of copyright to one single legal entity by each project contributor. This is the main alternative for project with copyleft licenses. As with any change related to licensing, it has to be decided early in the life of the project as it requires the agreement of all copyright holders.

Where a non-permissive licence is required to distribute software binaries or packages, one option is maintain the source code repository under a permissive licence but re-licence the software at distribution time under the required licence. This maintains flexibility about what licence to use in the future, but allows linking or repackaging with more restrictively-licensed open source software in the present.

## 4.3 Collaboration Agreements

For software developed in collaboration between partners from different institutes consideration may be given to establishing a Collaboration Agreement. This should define the licence to be used for the jointly developed software and typically also describes other rules for governing the way decisions are taken e.g. rules for accepting new members and rules for managing the development life cycle of the product. Typically, it also identifies

a 'prime distributor' that takes the role of managing and deploying new releases of the software. Transferring copyright to the prime distributor may also help ensure the software can be maintained over the full life-time of the project in situations where the original developer (i.e. owner) can no longer be contacted.

## 4.4 Commercial Exploitation

Any software distributed under a given licence may also be distributed under one or more different licence(s). This is often referred to as dual or multiple licensing. A frequent case of dual licensing is the public release of a programme under a Copyleft licence (such as GPL) and, contemporaneously, a bilateral agreement between the programme owner and a third party company for the commercial exploitation of the software.

In the case of permissive free software licences, as all permissions for appropriation have been given to any third party, and so commercial exploitation by dual licensing becomes unnecessary. Therefore, permissive licenses, such as the ASL, MIT and BSD licenses, are preferred by many companies because such licenses make it possible to use open-source software code without having to turn proprietary enhancements back over to the open source software community. These licenses encourage commercial adoption of open-source software because they make it possible for companies to profit from investing in enhancements made to existing open-source software solutions.

## 5 Recommendations

HSF encourages all its members and partners to make available the software they develop as Open Source, unless forbidden due to external constraints such as collaborative agreement. Only open-source software can become HSF projects. The open-source licence(s) adopted should be widely used licences approved by the Open Source Initiative (OSI). It should not be necessary to create a new licence and using a unusual licence may hinder the redistribution of the software by third parties.

The exact licence chosen may depend on several factors but they should enable the following key points:

- Make the software distributable by other projects through their natural software distribution channels. This should anticipate their need to distribute modified versions of the software to fix bugs downstream or address compatibility requirements.
- Make the software and its source code reusable by other HSF or open-source projects using the most widely used open-source licences, whether copyleft or permissive.
- Build a community around the software project and maximize the contributions by the users back to the project.

The GNU and Apache projects have demonstrated that these goals can be achieved either with copyleft or permissive licence approaches. Both approaches have vocal supporters and no consensus has emerged in the last 30 years of open source software development.

For projects producing libraries and taking the copyleft route, LGPL should be preferred for program libraries when the goal is to allow wide and rapid adoptions by applications with different licenses.

Permissive licences are good candidates when adoption by commercial partners must be possible and that there is a risk that at a later stage it will be difficult to contact all the copyright holders to discuss dual licensing. This is sometimes a requirement in projects funded by governmental bodies. In the copyleft case, it may be necessary to require that the copyright of contributions are assigned to the project to achieve this.

Whatever the licence chosen, software must contain in the notice a statement acknowledging the copyright owner(s) and the licence chosen. See next section for examples.

In addition, the following points must be taken into consideration:

1. When contributing to an existing project, release your modified versions under the same licence as the original work.
2. A licence should be assigned to tutorials, reference manuals and other large works of documentation. The GNU Free Documentation Licence (GFDL) [7] is a strong copyleft licence for educational works, initially written for software manuals, and includes terms that specifically address common issues arising when those works are distributed or modified. Licences from the Creative Commons family are also gaining ground in this area and provide a viable alternative. [8]

## 6 Examples

This section contains examples for specifying licence terms, based on real licenses from different HEP laboratories. You can use them as a source of inspiration but you need to customize them to your specific needs and local context.

The licence should contain a statement in the header of each source file acknowledging the copyright of the owner(s) and the applicable licence.

(i) Copyright

### 6.1 Copyright

In the following we give some examples of Copyright statements that are used by CERN depending on whether the software is owned solely by CERN or by CERN and external partners:

- for software owned solely by a single institute, in this case CERN:

© Copyright [year] CERN

- for software developed by a collaboration but where ownership has been transferred to a single institute, in this case CERN:



© Copyright [year] CERN [for the benefit of the [Name of appropriate group] Collaboration]

- for software owned by partners in small collaborations:

© Copyright [year] [names of all copyright holders]

- for software owned by partners in large collaborations:

© Copyright [year] Copyright Holders of [name of the collaboration or joint project]. See [https://link] for details of the Copyright Holders

## 6.2 Applicable licence

One of the following licence statements must be included, immediately following the copyright statement, **and followed by the text of the relevant license as shown in the references**:

- For software distributed under the default GPLv3 licence [9]:

This software is distributed under the terms of the GNU General Public Licence version 3 (GPL Version 3).

- For software distributed under the LGPLv3 licence [10]:

This software is distributed under the terms of the GNU Lesser General Public Licence version 3 (LGPL Version 3).

- For software distributed under the Apache licence v2 [11]:

This software is distributed under the terms of the Apache version 2.0 licence.

- For software distributed under the BSD-2-Clause licence [12]:

This software is distributed under the terms of the BSD-2-Clause licence.

- For software distributed under the BSD-3-Clause licence [13]:

This software is distributed under the terms of the BSD-3-Clause licence.

- For software distributed under the MIT licence [14]:

This software is distributed under the terms of the MIT licence.

The verbatim text of the licence should be copied either in a dedicated file which is part of the distribution (in this case the filename is COPYING) or directly below the licence statement.

The text of each licence to be copied verbatim for each of these licences can be found here [9,10,11,12,13,14].

## References

- [1] Final Report from the task force on Open Source Software Licence at CERN: <http://indico.cern.ch/category/4252/>
- [2] The Apache Software Foundation: <http://www.apache.org>
- [3] [http://en.wikipedia.org/wiki/Software\\_licence](http://en.wikipedia.org/wiki/Software_licence)
- [4] GNU GENERAL PUBLIC LICENCE Version 3, 29 June 2007 <http://www.gnu.org/copyleft/gpl.html>
- [5] ROOT software terms and conditions: <https://root.cern.ch/root/License.html>
- [6] Top Open Source Licences, source BLACKDUCK, July 2015: <https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>
- [7] GNU Free Documentation Licence (GFDL): <http://www.gnu.org/licenses/fdl.html>
- [8] Creative Commons licences: <http://creativecommons.org>
- [9] Text of GPL v3 licence, June 2007: <http://opensource.org/licenses/GPL-3.0>
- [10] Text of LGPL v3 licence, June 2007: <http://opensource.org/licenses/LGPL-3.0>
- [11] Text of Apache 2.0 licence Jan 2004: <http://opensource.org/licenses/Apache-2.0>
- [12] Text of BSD-2-Clause licence: <http://opensource.org/licenses/BSD-2-Clause>
- [13] Text of BSD-3-Clause licence: <http://opensource.org/licenses/BSD-3-Clause>
- [14] Text of MIT licence: <http://opensource.org/licenses/MIT>