Guidance on licences for software

Version	Date	Amended by	Notes
V0.1		Aslam Ghumra	Initial Draft
V0.2	26/04/2022	Aslam Ghumra	Comments from Lisa Bird
V0.3	30/05/2023	Aslam Ghumra	Amendments from Alex Fenlon and Lisa Bird
V0.4	01/06/2023	Aslam Ghumra	Typos fixed by Alex Fenlon
V0.5	02/06/2023	Aslam Ghumra	Typos fixed by Stephanie Thompson
V0.6	17/07/2023	Aslam Ghumra	Amendments by Andrew Edmonson
V0.7	19/07/2023	Aslam Ghumra	Amendments by Judith Hegenbarth, Mike Dainton & Alex Fenlon
V0.8	22/01/2024	Aslam Ghumra	Added paragraph to enforce the message to software creators who collect personal data, about their responsibilities.
V0.9	12/02/2024	Aslam Ghumra	Final version

[1] Akhmerov, Anton, Bazuine, Merlijn, Beardsell, Julie, van den Bogerd, Rianne, Branchett, Susan, Dunning, Alastair, Keijzer-de Ruijter, Meta, Cruz, Maria, Martinez-Lavanchy, Paula, Schenk, Mark, Spaargaren, Margot, & Teperek, Marta. (2021). TU Delft Research Software Policy. Zenodo. https://doi.org/10.5281/zenodo.4629662

Contents

Guidance on licences for software 1
Purpose 3
Introduction 3
Intellectual Property 3
Licences 4
Permissive licences 5
Proprietary licences 6
Permissive or Proprietary? A Decision Tree 6
Signposting for further advice 7
Appendix A Definitions 9

Purpose

The purpose of this document is to help researchers, whether staff or students at the University of Birmingham, understand their options in allocating an appropriate software licence. The following information will be split up on web pages on the Library section of the Intranet.

Introduction

Software is essential for research in the contemporary academic context[2]. An overwhelming majority of researchers develop and re-use software as part of the research process to generate, process or analyse results[3]. Consequently, software is increasingly recognised as its own research output. As such, software should be well documented, preserved and whenever appropriate, the FAIR principles (Findable, Accessible, Interoperable, Reusable) should be applied[4].

Software can be made freely available in many ways, for example to a large audience through a repository such as GitHub or shared peer-to-peer under restricted licences. Alternatively, it is possible to run software as web services, restricting access to the output of the software, while keeping the source code confidential.

When making software available to others there should always be a balance between the level of openness, however, as possible commercial exploitation of software and any (legal) restrictions may prevent sharing it as Open Source. You may also need to consider any requirements of those funding the research[5]. This may be different if the funder is a charity or a government department, such as the MoD, where there may be sensitivity issues.

This document (guidance) sets out what the researcher needs to think about in order to share their software, as Free and Open Source Software (FOSS)[6]. We have also provided a set of definitions in Appendix A, to help understand what is meant by the terminology, acronyms, and phrases used in this guidance.

Intellectual Property

Intellectual property ("IP") is the result of creativity of the human mind and, as property, it can be protected using various intellectual property rights ("IPR") such as copyright, patents, trade marks, trade secrets etc. IPRs often protect the way something is expressed or a method of doing something, it will not protect the concept that is represented by it; for example copyright will protect the novels of a particular author but it will not protect against another person creating a different story along a similar theme.

The development, protection and commercial exploitation of IP generated by the research process is now a major focus for researchers, institutions, funders and industrial partners, with many grant applications requiring IP statements to be included within project proposals.

---

[2] https ://www.ukri.org/files/infrastructure/the-uks-research-and-innovation-infrastructure-opportunities-to-grow-our-capacity-final-low-res/ Page 125

[3] S.J. Hettrick et al, UK Research Software Survey 2014: http://doi.org/10.5281/zenodo.14809

[4] https://fair-software.nl/about, & Lamprecht, Anna-Lena et al. 'Towards FAIR Principles for Research Software'. Jan. 2019 DOI: 10.3233/DS-190026

[5] https://www.ukri.org/what-we-do/developing-people-and-skills/mrc/fellowships/guidance-for-fellowship-applicants/2-2-12-considerations-before-applying/2-12-open-source-software

[6] https://opensource.com/resources/what-open-source

Members of staff, which includes both employees and honorary staff, at the University may generate IP in the course of their work at the University. Generally, under English law, IP that is generated by employees in the course of their employment belongs to the employer, in this case the University. When a member of staff generates some IP while carrying out their role and believe this IP may be of some economic value, (or could be developed to have some economic value), they should bring it to the attention of UoB Enterprise, either directly, or by reporting it to their Head of College (or their Senior Officer in the case of professional services staff). The Code of Practice for Research[7] (CoPR) outlines these requirements.

Conversely, students own the IPR in the work they create during their studies. Where students are involved in collaborative projects with staff members, for example working alongside staff and participating in research projects, the University may require that IP generated by the student is assigned to the University. Again, this is covered by the CoPR.

Computer software (also referred to as code or programs) will be protected via copyright which lasts for the author's lifetime, plus 70 years[8].

Those developing software also need to consider if they are using IP (including software, software libraries or software generated by AI) produced by other people within their software or as a library. Where this is the case, you may need to seek permission (a licence) from them for re-use. Open Source software is usually already licensed for re-use provided the users follow the terms of the licence, one of which may be the requirement to reference and acknowledge the original author in any new software created using that code. You should also bear in mind that not all reuse licences are compatible with each other[9].

You will need to consider the licence conditions of any software you have used to create your work. Some tools that you use to create software, e.g. Matlab and Labview, require you to have a commercial licence if you wish to make the software available for yourself or others to use for commercial purposes. If you have used source code for your software that was licenced under a GNU (General Public License) Public Licence (GPL) then you will need to release your software under the same licence.

The software developer should also consider the impact of collecting personal data, whether it will be held within the software or any associated files required by the software. Thought needs to be given as to where the data is stored and how it will be curated and disposed of, as per the Data Protection Act 2018 (commonly known as GDPR). More information can be found on the Information Commissioner's Office webpages (ICO)[10].

Licences

Software is usually classified under two broad headings with licences which grant the licensee specific rights; proprietary software and Free and Open Source Software (FOSS). The distinct conceptual difference between the two is the granting of rights to modify and re-use the software product obtained by a customer. FOSS software licenses both rights to the user and bundles the modifiable source code with the software ("Open Source"), while proprietary software typically does not license these rights and may keep the source code hidden ("closed source").

[7] https://www.birmingham.ac.uk/documents/university/legal/research.pdf

[8] https://www.gov.uk/government/publications/copyright-notice-duration-of-copyright-term/copyright-notice-duration-of-copyright-term

[9] https://joinup.ec.europa.eu/collection/eupl/licence-compatibility-permissivity-reciprocity-and-interoperability

[10] https://ico.org.uk/for-organisations/data-protection-and-the-eu/data-protection-and-the-eu-in-detail/the-uk-gdpr/

Permissive licences

Permissive licences have minimal requirements about how the software can be redistributed. They do not block appropriation, nor do they force the redistributor to open the modified source code. By its nature, code written under a permissive licence can generally be incorporated in code written by a more restrictive licence, but not the other way around.

There are many types of permissive software licences that are commonly used for FOSS. Such licences include minimal restrictions on how the software can be used, modified, and redistributed, usually including a warranty disclaimer. These can be referred to as Berkeley Software Distribution (BSD)-like or BSD-style licences[11] and examples include:

- **MIT (Massachusetts Institute of Technology) license-** gives users express permission to reuse code for any purpose, sometimes even if code is part of proprietary software. Example of software that use this licence are X Window System, Ruby on Rails, Nim, Node.js, Lua, and jQuery.

  Arguably, the biggest advantage of using the MIT License is that it is very permissive. The quality of the licence allows it to be both business-friendly and open-source friendly, while still making it possible to be monetise. Further details of the MIT Licence can be found here: https://choosealicense.com/licenses/mit/

- **GNU General Public License** is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. A popular example of software that uses GPL is WordPress.

  Further details of the licence can be found here - https://www.gnu.org/licenses/gpl-3.0.en.html

- **BSD 3-clause license** allows you almost unlimited freedom with the software. An example of software that uses this licence is Numpy, Nginx and C Shell.

  Further information on this licence can be found here- https://opensource.org/licenses/BSD-3-Clause

- The **Apache License** allows users 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. An example of software that uses this licence are PyCharm, Singualrity and Jetty.

  Further information on this licence can be found here- https://www.apache.org/licenses/LICENSE-2.0

- **Apple Public Source License (APSL)** is the open-source and free software licence under which Apple's Darwin operating system was released in 2000. A free and open-source software licence was voluntarily adopted to further involve the community from which

[11] https://en.wikipedia.org/wiki/Permissive_software_license#cite_note-nmr-bsdlike-1

much of Darwin originated. An example of software that uses this licence are Keychain and XQuartz.

More information on this licence can be found here- https://opensource.apple.com/license/apsl/

- **CC0** (CC Zero) is a public dedication tool, which allows creators to attempt to give up their copyright and put their works into the worldwide Public Domain. CC0 allows re-users to distribute, remix, adapt, and build upon the material in any medium or format, with no conditions, including no requirement to reference the original creator. However, it is an essential part of academic practice to reference any code written by other people, failure to do so could raise plagiarism issues.

  Further details of the licence can be found here : https://creativecommons.org/share-your-work/public-domain/cc0/

Permissive licences are often non-exclusive. This means that an owner can licenses their software under many different routes if they choose too, although this might make managing those rights difficult. While open licences are often irrevocable (i.e. once the software is openly licensed it cannot be closed again) an owner could choose to remove the code from the repository and release a new edition under different licensing terms, including all rights reserved. This is especially important for works that are released under CC licences or other so called public domain declarations. the nature of CC0 is such that any one is free to do anything with it without restriction and that would include taking a copy (full or partial) and wrapping it up behind an "all rights reserved" regime. While this may be ethically questionable and open to challenge, it is certainly possible.

Proprietary licences

Many software programs are proprietary with all rights to access and use them being controlled by the owner or creator. Here the owner of the software grants the right to use the software under an End-User Licence Agreement (EULA), but the ownership rights in those copies remains with them (hence use of the term "proprietary"). It is typical of EULA's to include terms which define who can use it and the permitted uses of the software, the number of installations allowed, and many other terms. An example of this would be Microsoft Windows.

Permissive or Proprietary? A Decision Tree

The decision to license your software may depend on the rights of others, whether they are software owners, researcher funders or collaborators. However the choice of licensing is the creators, so we have created the following decision tree to guide software developers, researchers, and staff on when they can apply an appropriate licence to their software.

The following is a step-by-step guide, followed by a workflow diagram showing these steps visually,

Step	Question	Response
1	Can the software be defined as Dual-Use?	Yes -> Talk to Legal Services
		No -> Go to Step 2
2	Is the work funded or financed?	Yes -> Go to Step 3
		No -> Go to Step 4

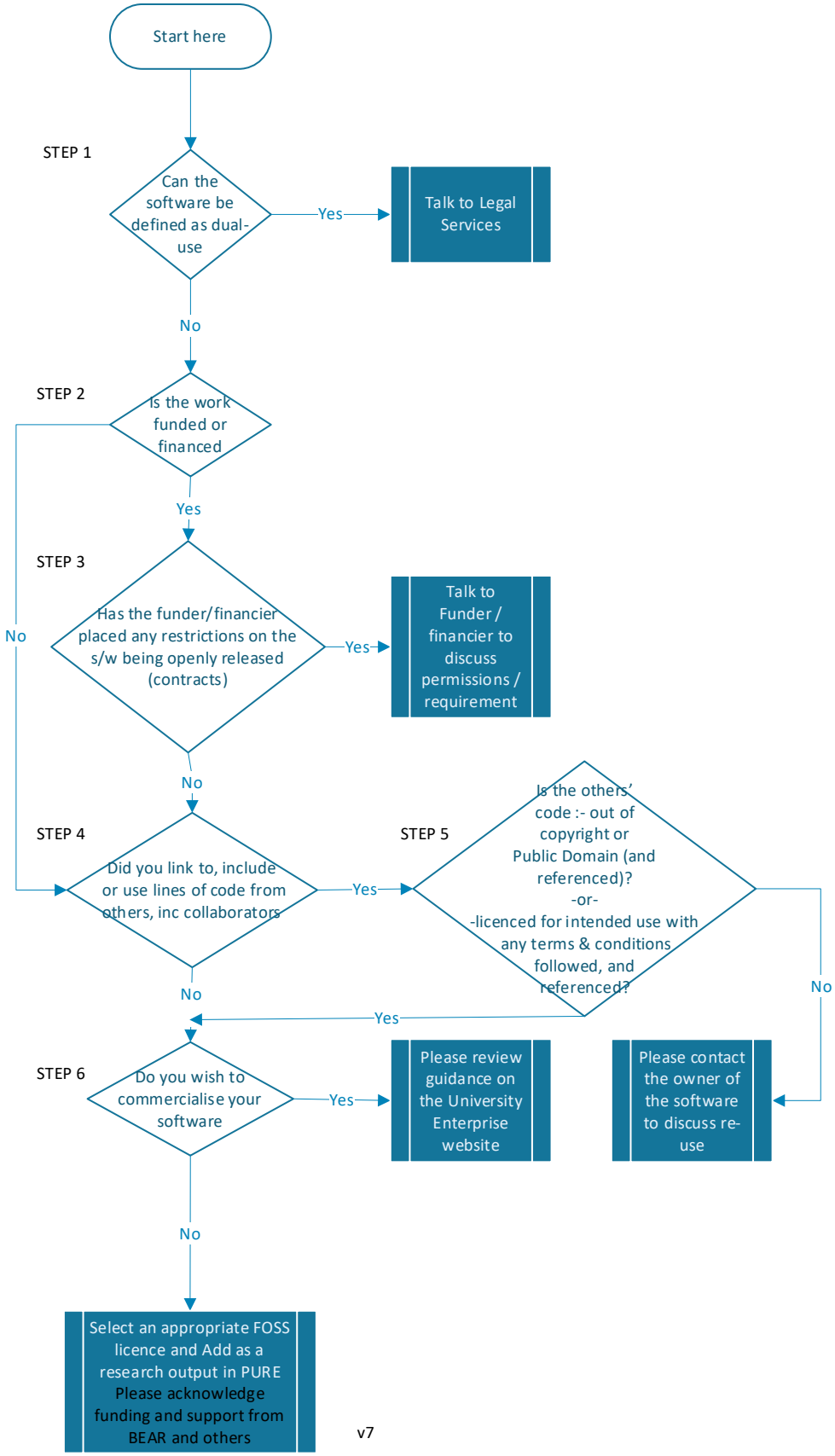
3	Has the funder or collaborators placed any restrictions on the software being openly released (contracts)?	Yes -> Review the funding grant or contract, or Talk to Funder /financier to discuss permissions / requirement
		No -> Go to Step 4
4	Did you link to, include or use lines of code from others, inc collaborators?	Yes -> Go to Step 5
		No -> Go to Step 6
5	Is the others' code: • out of copyright or Public Domain (and referenced)? Or • licensed for your intended use with any terms & conditions followed, and referenced?	Yes -> Go to Step 6
		No -> Please contact the owner of the software to discuss re-use
6	Do you wish to commercialise your software?	Yes -> Please review guidance on the UoB Enterprise website and add as a research output in PURE
		No -> Select an appropriate FOSS licence and add as a research output in PURE. Please acknowledge funding and support from BEAR and others

Signposting for further advice

The CoPR reiterates the commitment to academic freedom found in the University's Ordinances, clarifies the University requirements, and offers information on the University's facilities for advice on regulatory and ethical issues, which also cover research output.

There are several teams who can help guide you:

- Advanced Research Computing / BEAR: bearinfo@contacts.bham.ac.uk
- Library: Copyright & Licensing Team copyright@contacts.bham.ac.uk
- Research Strategy and Services Division : Can help you decide what you can do in respect to copyright and how your research was funded: https://intranet.birmingham.ac.uk/rssd/contacts/index.aspx (via university login)
- University of Birmingham Enterprise: If you decide you want to protect your software and license it out commercially, see: https://www.birmingham.ac.uk/partners/enterprise/index.aspx


Start here
STEP 1
Can the software be defined as dual-use
Yes
Talk to Legal Services
No
STEP 2
Is the work funded or financed
Yes
STEP 3
Has the funder/financier placed any restrictions on the s/w being openly released (contracts)
Yes
Talk to Funder / financier to discuss permissions / requirement
No
No
STEP 4
Did you link to, include or use lines of code from others, inc collaborators
Yes
STEP 5
Is the others' code :- out of copyright or Public Domain (and referenced)?
-or-
-licenced for intended use with any terms & conditions followed, and referenced?
No
Yes
No
STEP 6
Do you wish to commercialise your software
Yes
Please review guidance on the University Enterprise website
Please contact the owner of the software to discuss re-use
No
Select an appropriate FOSS licence and Add as a research output in PURE
Please acknowledge funding and support from BEAR and others

v7

Appendix A Definitions

In this document, “software” means a set of coded instructions designed to cause a computer, a machine, or automatic data processing equipment to perform a task. For the purposes of these guidelines, the term software also includes (where relevant) the associated documentation (including but not limited to a manual), hierarchy, platforms, API’s and specific hardware (if any) required to run the aforementioned instructions.

Attribution: The requirement that the original authors must be credited for their work.

Derivative a piece of code based upon one or more pre-existing software, such as a translation, an abridgement, a condensation, an expansion or any other form in which a code may be recast, transformed, or adapted. In order to constitute a derivative, the change needs to be significant. In general, a dynamic link to another program, a database or library does not provide a derivative code, but a static link can (under circumstances) provide a derivative code (though the Open-source foundation advocates a stricter view.

Dual-use (Goods): goods, software and technology which are commonly used for civil purposes, but that can have military use or that can contribute to the production and deployment of weapons of mass destruction (WMD). For a non-exhaustive list see https://www.gov.uk/guidance/uk-strategic-export-controls

Intellectual Property (IP): all forms of intellectual property rights including patents, utility models, trade marks, trade or business names, database rights, service marks (be it either registered or unregistered), copyright, right to mask work of integrated circuit, design right (be it either registered or unregistered), know-how and other proprietary knowledge and information, rights protecting goodwill and reputation and the applications of the protected forms of intellectual property rights or having equivalent effect and all rights of licence and consent in respect to any of these aforementioned rights.

Inventor: A natural person who has made a substantial, significant intellectual contribution to the development of Intellectual Property is regarded as an inventor. To qualify as an inventor, an individual must have contributed sufficiently to the software to take public responsibility for its content and the individual’s intellectual contribution must be critical to its main conclusions. When there is disagreement, an inventor can or should be able to identify, demonstrate and provide documented evidence of this inventive contribution to specific claims in a patent application.

Licence including “Licence” or “Software Licence”: a document that provides legally binding guidelines and requirements for the use and distribution of the software. Licences typically provide end users with the right to one or more copies of the software without violating copyright law. FOSS licences typically grant this right to all of humankind rather than to a specified end-user.