



Australian Research Data Commons

Research Software Rights Management Guide

How researchers may license their software in order to share it with others



ARDC is enabled by NCRIS



Research Software Rights Management Guide

Australian Research Data Commons

Email: contact@ardc.edu.au

Report Completed in June 2021

More information on the ARDC
is available at www.ardc.edu.au

© ARDC, 2021



Except otherwise noted, the reuse of this document is authorised under a Creative Commons Attribution 4.0 International (CC-BY 4.0) licence (<https://creativecommons.org/licenses/by/4.0/>). This means that reuse is allowed provided appropriate credit is given and any changes are indicated.

DOI: 10.5281/zenodo.5003962

Contents

Introduction	1
Background	2
Permissive Licences	2
Copyleft Licences	3
Considerations when selecting a licence	4
What licences are incorporated into my software?	4
Are there implications from where I host my software?	4
Do I want derivatives, additions and distributions to use the same licence?	4
Can others use my software in their proprietary software?	4
Recommended licences	5
Ab initio development and permissive licences	5
Copyleft Licences and Incompatibility	5
Further Reading and Resources on Licensing	5
Popular Open-Source Licences	6
Comments on Workflows, Techniques and Algorithms	6
Workflows	6
Algorithms and Techniques	6
Acknowledgements	6



This guide explains how researchers may license their software to share with others.

Introduction

Many researchers who create and analyse data use specialised software.

Sometimes they create their own software because the functionalities they need for data analysis are not available 'off-the-shelf'. When the time comes to publish their data, these researchers may also see benefit in publishing their software as well. For example, they may wish to ensure that their analysis of their data may be reproduced by others.

This guide's companion publication, the [ARDC Research Data Rights Management Guide](#), recommends the use of Creative Commons licences for data. However, this guide doesn't recommend Creative Commons licences for software. While it's possible to license software under a Creative Commons licence, and some people do, there is an ecosystem of open-source software licences that are primarily directed toward sharing and improving software.

The advantage of using open software licences over Creative Commons is that many of the open software licences pre-date Creative Commons and, consequently, are widely understood and used by the software development community. Some licences also make express reference to, and provide separate requirements for, source code (being the code written by humans) and object code (being code created by compiling the source code), which Creative Commons licences do not.

This guide does not address the proprietary licensing of software. Rather, it addresses the types of open-source licences, and considerations you (as a researcher) should have in deciding which licence to adopt for sharing. The key point here is sharing: this guide is intended for researchers who want to share their software. Finally, this guide also briefly discusses the treatment of certain workflows, techniques and algorithms, which are also common research outputs.

Background

Copyright law protects software as a literary work—but unlike other literary works, software does not need to demonstrate originality.

Two software applications may perform the same function, but if their code is different, each may be afforded copyright protection. So, copyright protects software, and as discussed in the [ARDC Research Data Rights Management Guide](#), the law will therefore presume that all rights are reserved in software by the developer, unless a licence is applied to it.

There are two main types of open-source software licences that are popular in the software development community—these are **permissive licences** and **copyleft licences**. The term ‘copyleft’ is a play on words relating to ‘copyright’. Copyleft is not a new type of intellectual property right; rather, it is simply intended to denote that the licences are not as restrictive as all rights reserved proprietary licences. Within the copyleft licences, there are two sub-categories, **strong** and **weak**, which are discussed below.

Permissive Licences

Permissive licences, as the term suggests, enable users of software to adapt, modify and incorporate the software into other software with very few requirements. If you want to share your software and enable others to use it with very few restrictions, permissive licences are preferred. For example, permissive licences may only require that the original developer be attributed for their software, similar to the way authors adopting a Creative Commons Attribution licence are cited in relation to the use of their works. In the software development community the accepted practice to attribute a developer is to incorporate a ‘licence’ or ‘notice’.txt file with software, and or to provide a menu in a software application that provides access to licence details and attribution.



Examples of permissive licences include the Apache 2.0, MIT, and BSD-3 Clause licences, which are discussed later.

Copyleft Licences

Copyleft licences carry a requirement similar to the ShareAlike feature of Creative Commons licences. In essence, when copyleft licensed software is incorporated into a software project, or an adaptation is made of copyleft licensed code, the software project, or the adaptation must be licensed under the same or a compatible copyleft licence. Weak copyleft licences will require the same or a compatible licence to be used for derivatives of, or additions into, the licensed code. Strong copyleft licences require the same or compatible licence to be used also for distributable (compiled) software.

Figure 1 depicts the likely consequences of licensing open-source software with both weak and strong copyleft licences. The licence selection for the copyleft Open-Source Software (top) will have an impact on the licensing of derivatives, additions and distributions.

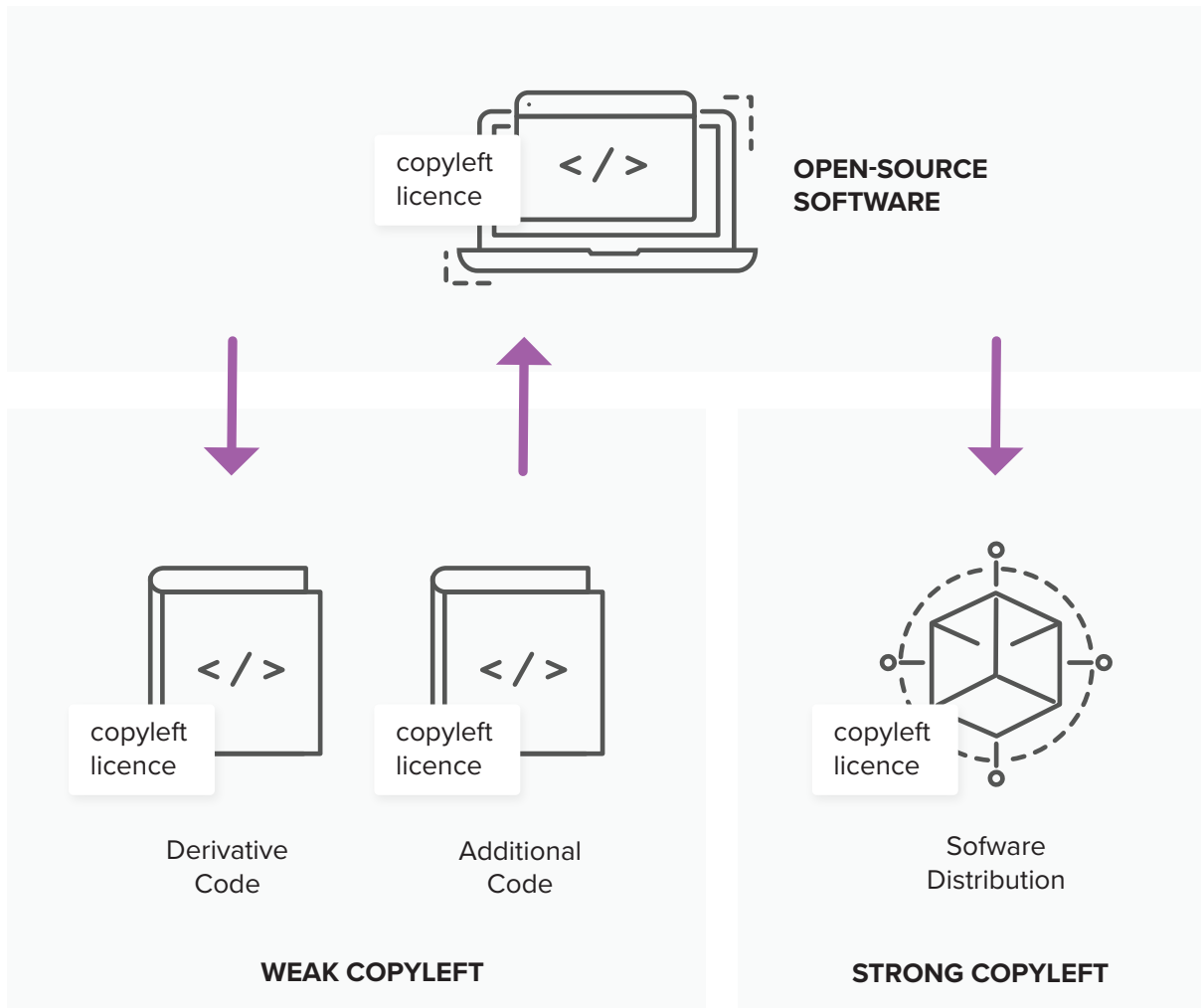


Figure 1. Derived from a diagram by Michael miles CC BY 3.0 Unported. The author's are grateful for Mr Miles conceptualisation of open source software and summary of the issues that pertain to it.

Considerations when selecting a licence

To determine which licence you should adopt, you should ask the following four questions:

1 What licences are incorporated into my software?

If you intend to publish software that incorporates elements from other open-source software projects you may need to consider licence compatibility. Licence compatibility is an issue that arises when an attempt is made to incorporate two or more pieces of software together, that are each licensed under separate and contradictory licences. For example, you may like to publish your software under a permissive licence (where you only require attribution), but if your software incorporates software licensed by someone else under a strong copyleft licence, it's likely that you will need to apply the same or a compatible strong copyleft licence to your software as well.

As a software developer it's important to make a record of any software you incorporate and the license under which it is supplied. This record will form the basis for your licensing decision.

2 Are there implications from where I host my software?

Generally, open-source developers distribute their software from a website or platform that enables other developers to find and access their code. However, some platforms contain their own terms-of-service that place requirements on developers to license their software in a particular way. For example, GitHub, a popular platform in the software development community, requires developers to upload code under a licence that permits derivatives to be made. You should check the licensing terms of any third-party software incorporated into your software before you decide upon a hosting platform. Your organisation might also have licensing requirements or policies and hosting procedures that you should consider.

3 Do I want derivatives, additions and distributions to use the same licence?

If you want others to make derivatives, additions and distributions based on your software under the same (or a compatible) licence, you will want to select a copyleft licence. If not, choose a permissive licence.

4 Can others use my software in their proprietary software?

If you chose a copyleft licence in the question above, you have one further choice to make—will you permit others to use your software in their proprietary software. If not, you will select a strong copyleft licence.

Recommended licences

Ab initio development and permissive licences

If you intend to publish software you have developed in its entirety, we recommend the permissive licences. While there are numerous popular permissive licences, we suggest the Apache 2.0 Licence.

It provides permissions akin to the CC Attribution Licence, and is well recognised within the open software community. For example, it is used in the Android Open Source Project. Another important consideration is that of all of the permissive licences, the Apache 2.0 Licence’s strong limitation of liability clause—which must be interpreted subject to applicable law—means that it is less likely to be considered an unfair term under the Australian Consumer Law. The clause also most closely approximates similar clauses to the CC version 4.0 licences. Software licensed under the Apache 2.0 Licence can be incorporated into other open-source projects, including those licensed under copyleft licences.

To adopt the Apache 2.0 Licence into your software, include a NOTICE.txt file containing the following:

Copyright [yyyy][name of copyright owner]
Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License.
You may obtain a copy of the License at:
<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

Copyleft Licences and Incompatibility

Not all copyleft licences were created equal. Some software licensed under a copyleft licence cannot be legally mixed with software licensed under another copyleft licence. For example, software that combines code released under version 1.1 of the Mozilla Public License (MPL) with code under the GNU General Public License (GPL) cannot be distributed without violating one of the terms of the licenses. This highlights the benefit of using a permissive licence. Software less than 300 lines of code probably doesn’t warrant a copyleft licence. Software relicensing—where every contributor is contacted and invited to agree to a permissive licence or less restrictive copyleft licence—may be the only option if a licensing incompatibility proves insurmountable. This is a difficult and unpredictable process that can be avoided by proper consideration of licensing in software development decisions early in a software project.

Similarly, do not create your own licences. There are many open-source licences available. Creating your own licence will only add more complexity and limit interoperability.

Further Reading and Resources on Licensing

Read licences before using them. Tldrlegal.com provides summaries of many of the open-source licences as well as linking to their full text.

GNU provides a list of compatible and incompatible licences with their GPL’s.

Popular Open-Source Licences

Permissive	Copyleft (Weak)	Copyleft (Strong)
Apache 2.0	LGPL v3	GPL v3
BSD 3-Clause	Mozilla Public Licence 2	GPL v2
MIT		

Comments on Workflows, Techniques and Algorithms

Workflows

Workflows, either in the form of diagrams or written as part of code, will be protected by copyright in either form. A workflow diagram will be protected as an artistic work, and in code, it will be protected as part of the rest of the software. To share a workflow in the form of a diagram (being an artistic work), we recommend applying a Creative Commons Attribution 4.0 licence. To share your software that contains a workflow, you should apply the licensing considerations described above.

Algorithms and Techniques

Algorithms and techniques will not generally be protected by copyright. This is because copyright protects expressions of ideas rather than ideas themselves. However, it is not uncommon for algorithms to be expressed as code in software. When taken together, software that incorporates an algorithm expressed as code will attract copyright protection.

If other software containing the same algorithm or technique is developed in a different programming language, it won't infringe upon the copyright in the first software because the expression is different. Put another way, the presence of an algorithm expressed as code in your software will not deny

your software copyright protection, nor prevent it from being licensed. To share your software that contains an algorithm or technique, you should apply the same licensing considerations described above.

However, it should be noted that this area of law is continuing to develop. Cases in Australia and internationally have finely balanced the legal recognition and protection of algorithms and techniques incorporated into software. As technology advances, particularly in the area of artificial intelligence, it is likely that the courts will also advance new perspectives on capacity of the law to protect algorithms and techniques.

Acknowledgements

This document, ARDC Research Software Rights Management Guide, was written by Greg Laughlin and Baden Appleyard.






Australian Research Data Commons

CONTACT

ardc.edu.au
+61 3 9902 0585
contact@ardc.edu.au

FOLLOW

 @ardc_au
 australian-research-data-commons
 + subscribe to our newsletter



ARDC
is enabled
by NCRIS