

Open Source Software Licensing FAQ

Here we attempt to address some common questions that academics and researchers have about licensing software as open source. Most of the issues raised can be complex, so for more details we include links to relevant information on the [OSSWatch website](#).

What type of software can be licensed?

Anything from stand-alone programs and code libraries to programming languages and operating systems. The OSSWatch website lists some [examples of OSS](#). Content (e.g. images, text) either as part of an application or website or as documentation and data may also be licensed openly, although it is important to note the considerations for licensing software are somewhat different (so, for example, [Creative Commons licences are not recommended for software](#)).

What does a software licence cover? (See [An Introduction To Ownership And Licensing Issues](#).)

“When you write software, you are creating a kind of property.” The owner of this property will have certain rights (including copyrights and potentially patent rights) which limit the ability of other people to use, copy, alter and redistribute the software. Through a licence the owner provides permissions for these acts, providing certain conditions are met. The conditions may cover issues such as attribution, limitations of liabilities, and requirements regarding the sharing of code that has been altered. ([An Introduction To Ownership And Licensing Issues](#).)

The issue of software patents has been a contentious one, and the legal details vary around the world, however it is the general view that an open source licence is also a grant of a licence to the user regarding any patents necessary for using the software, either implicit or explicit. Thus it is important to be aware of whether you or anyone else owns patents relevant to the software. For more information see the OSSWatch briefings on [Software Patents](#) and [Free and open source software and your patents](#).

Do I have to choose a licence if I want to share my software? Can't I just put it online?

By default, the implied license for software is “all rights reserved”. There may be reasons for wanting to post software online without granting a licence ([Unlicensed code: is it ever OK?](#)), however code posted as such is not shared in a way that allows others to use it ([Unlicensed code: Movement or Madness?](#)).

Who can license software?

Only the rights owner can licence the software ([Can You Contribute Code To An Open Source Project?](#)). If the software is written by University of Bristol employees then the ownership of rights is governed by the University policy on rights ownership. However, others may also have rights and licensing will need their agreement. Be careful about work from students, contractors and project partners. Whenever people not employed by the University contribute to developing code it is best to have [contributor licence agreements](#) in place. Also, be aware of any relevant conditions from project partnership agreements and funder's requirements. Finally, any third party content that has been used (licensed-in code, icons, artwork, etc.) will have its own licence requirements which may affect what licence you can use.

What does it mean to license software openly?

The Open Source Initiative has [ten criteria](#) for open source software and the free software foundation defines [four freedoms](#). The consequences of both these definitions are similar: there is

more to open source licensing than just providing software for some people to use for free. One consequence, obvious from the name, is that the source code must be openly available; it must also be editable, and other people must be free to distribute the original code and any code derived from it. This means it is not advisable to try to restrict the development of open source software through governance or business practices. Other notable consequences are that all open licenses allow commercial exploitation, none are limited to a single sector (e.g. academic or research only) or a single country. ([What Is Open Source Software?](#))

What if I want a more restrictive licence, such as for academic use only?

In that case you will need to talk to Research and Enterprise Development (RED) about a specific proprietary licence.

Why would we license software openly?

Aside from the benefits for users ([Benefits of Open Source Code](#)), the reasons for licensing software openly include fostering innovation ([Open Source And Open Innovation](#) & [Open Innovation In Software](#)), providing a sustainable future for code developed during a project after the funding has finished ([Sustainable Open Source](#) & [Planning For Sustainability](#)), and potentially increasing the impact of the software by making it more widely used.

What Open licences are available?

Many. See for example the list of [OSI reviewed licenses](#).

Which licence is best for me?

The best licence for any given project will depend on what rights and requirements should be covered, and on the importance attached to factors such as jurisdiction, popularity, familiarity and community regard. The different factors are described in the OSS Watch briefing [What Kind Of Licence Should I Choose?](#) and the OSS Watch [licence differentiator](#) tool can be used to show which open licences that best match your preferences.

Custom licences are a pain and not usually worth the bother. Aside from an unfamiliar license putting users off, a non-standard license will typically require a special legal assessment by any commercial users

What should I do in order to release code as OSS?

[Making your code available as open source](#) involves more than just indicating on the project web page that the code is licensed under a particular open source licence. As mentioned under “who can license software” it is important to make sure that all owners agree to the licence being used, and that the requirements of all third party, licensed-in content are met. This ‘due diligence’ can be facilitated with the use of automated audit tools such as [Black Duck](#), [White Source](#) [FOSSA.io](#) and [Apache RAT](#). The code must be made available, preferably using a repository that facilitates its ongoing development.

What are the overheads of releasing software openly?

It is desirable to [avoid abandon-ware](#), and so consideration should be given to the ongoing maintenance and development of the code. One key factor in [sustainable open source](#) development is [building a community](#) (see also [Roles In Open Source Projects](#) for the types of contribution that members of a community may make). As well as the effort entailed in fostering the community, contributions from the community will need to be managed, including legal aspects (more due diligence and [contributor licence agreements](#)).

Can we commercialize software we have licensed as OSS?

Yes. One option is [dual-licensing](#), a business model where the same software can be obtained as open source software or under more liberal commercial terms. Most users would choose the free

open source version, but some others may be willing to pay in order, for example, to use your software within proprietary, closed-source, products. This comes at the cost of combining the overheads of releasing software openly with the requirement of creating and enforcing a commercial licensing regime.

Other ways of monetizing open source software include the provision of additional benefits such as support and consultancy (which can be useful in promoting knowledge transfer). Also the software can be provided as a hosted service on a commercial basis.

How do I establish the impact of code I have released openly?

Some obvious ways of monitoring the use of your software after it is released as OSS, such as requiring registration before downloading and installing 'call-home' routines in the software itself, are frowned on by the OSS community. Sustainable open source does provide evidence of impact, for example you can use download statistics from the site hosting the software. If your software is distributed using a package management tool (e.g. pip, npm or maven) then you can also obtain installation statistics including use in other products. You can also make use of evidence of user engagement in the form of bug reports, feature requests, and code contributions.

Version: 1.2

Date: December 2018