



Publishing and Sharing Research Software

Research Data Management Service

University of Glasgow

Introduction

As a researcher you may develop your own software to meet specific needs related to the generation, processing and analysis of research data and results. This software can be integral to the reproducibility of your research and may need to be preserved and made available to others as part of your research outputs and data.

The University and many public funders, including the UK Research Councils, have policies requiring researchers to retain s¹. These requirements include any software created in the course of research that is necessary to replicate reported results.

Software can also be a valuable research output in its own right. By making your software available for others to use, you can help other researchers solve their own research problems, enable others to improve your software or adapt it for new purposes, and receive credit for the contribution you have made to your research community. Some software may even be viable for commercial exploitation.

However, publication of software requires some effort, if the software is to remain usable over the longer term.

Research software is often written in response to specific research needs, using whatever skills and resources may be to hand. Occasionally this software is developed according to a rigorous engineering process with formal specification, code reviews and testing. However, code in its

¹ The University of Glasgow Code of Good Practice in Research can be found here:

https://www.gla.ac.uk/media/Media_817774_smxx.pdf

The University of Glasgow Good Management of Research Data policy can be found here:

https://www.gla.ac.uk/media/Media_817778_smxx.pdf

Information about funder research policies can be found here: <https://www.dcc.ac.uk/guidance/policy>

raw state may not be written, organised and documented in a manner optimal for long-term maintenance and use by others.

This guide highlights the things you need to consider when you are preparing to publish or share your software. It will help you to publish your software in a way that maximizes the benefits for both yourself, the wider community of researchers and other users. It includes a selection of [Useful links](#) for further guidance and information, and a [Software publishing checklist](#), to help you decide whether and how to publish your software.

What is software publishing and when should it be done?

Software publishing involves making available by online means the source code of software that has been developed in such a way that it can be re-used by others, subject to any restrictions specified in the terms of use under which it is published. The software published may be an integral software product, or as is often the case, a segment of code that integrates with and builds on existing software or a script that automates the execution of a series of tasks in a given run-time environment. The following are typical research scenarios where software publishing would be expected to take place².

Scripts written to perform analysis in widely-used scientific software

Many researchers collect experimental data and then use established scientific software, such as Matlab or R, to perform statistical analysis and to visualise the results. It will enable validation of the reported results if you deposit in a data repository both the data themselves and the scripts that executed the analysis performed by the scientific software.

Scripts written to execute workflows

If you write a script to execute a series of operations that process data, for example by converting data from one format to another, it may not be necessary to make the script available, providing the data and metadata are accessible. But it may be of benefit to other researchers to execute the same workflow, so there could be value in making the script available with the raw data.

Software to implement a new algorithm

Research may involve performing complex mathematical operations on existing data, as in climate modelling and analysis of large economic and financial datasets.

If you write a piece of software to implement an algorithm for performing calculations on an existing dataset, and then publish the algorithm and the results of its implementation, while this may be sufficient in principle for the results to be validated, in practice it may be of limited use to anyone wishing to replicate your results.

² These scenarios have been based on those identified by the Software Sustainability Institute, in consultation with the EPSRC. See <http://www.software.ac.uk/resources/guides/epsrc-research-data-policy-and-software>.

As algorithms and their implementation in computer code can be highly complex, if the results are to be effectively replicated, it will generally be necessary to make the software that implements the algorithm available. This will also enable others to vary or improve the implementation of the algorithm. Anyone wishing to validate the results would also need access to the existing dataset in the version used.

Developing existing commercial software

If you develop or improve commercial software in the course of conducting research, your freedom to publish the code you have developed is likely to be constrained by the commercial interests of the software owner.

Published findings may be able to describe the new version of the software and the results obtained from its application, but you may not be able to make the software itself openly available. The owner of the software may permit you to make the software available ad hoc, subject to a non-disclosure agreement, so that other researchers can verify the results reported in published findings. You should clarify what you will be permitted to do by the software owner at the outset of your research.

Protecting or exploiting IP (Intellectual Property) in software

You may develop software which is necessary for the validation of your published research results, but which you believe has commercial potential that would be jeopardised by its disclosure. The University and many research funders permit a delay in publication of research results while arrangements are made to protect valuable IP. In this case you should seek the support of the University's Research and Innovation Services team³. They can help you ensure that if necessary, the intellectual property in the software is protected before your results are published. You will still need to take steps to make the code available to anyone who needs to validate the research once the results are published, and this can be done under a non-disclosure agreement if required.

Note that open licensing of source code may be compatible with commercial exploitation, for example by provision of software development or support services, and it may facilitate distribution of the software and development of a user community.

Software with ongoing research value

The University and public funders of research expect researchers to provide the means by which others can replicate and validate results reported in published research findings. Where software code is integral to validation of research results, researchers must make the code accessible to other researchers. If there is a valid reason for not making software openly available, for example, where there has been significant intellectual investment in developing the software, and where open publication would seriously compromise your future research, it may be possible to make the software available on an ad hoc basis only, subject to a non-

³ <https://www.gla.ac.uk/myglasgow/ris/ipcommercialisation/inventors/intellectualpropertymanagement/>

disclosure agreement, so that other researchers can validate the results reported in published findings.

Ownership and rights

Software is intellectual property, and as such attracts legal rights of ownership. It is not necessarily the case that you have the right to do whatever you want with software you develop:

- you may not own the software, or have the automatic right to distribute it, even though it is your work.
- even where you do have rights in the software, these may not be exclusive. Codevelopers and owners of software that you develop may have rights that need to be considered.

Before you can consider publishing your software, you must be sure that you either own the software or have been given permission to distribute it by any parties with rights in the software.

What rights exist in any software you develop will also determine the terms on which you can make the software available, and what uses others can be permitted to make of it. This will affect how your software can be licensed.

The University's Research and Innovation Services team can help you make the most of your software while respecting the rights of other interested parties. They can advise on your publishing options and the most suitable licence for your software. You should [contact this team](#) for specific advice, but the following general considerations are worth bearing in mind.

University staff

Intellectual property created in the course of employment is generally owned by the employer (unless any contract stipulates otherwise). This applies to all intellectual property, including software, created by University staff in the course of their employment.

University researchers are encouraged to disseminate their research outputs, and are allowed discretion in choosing how and by what means to publish their work, but they are expected to do this responsibly, in such a way that will not contravene any law, breach any contract, prejudice any ongoing or intended exploitation of the intellectual property, breach any confidence, or in any other way interfere with the rights of any other parties with interests in the work. When publishing software, you are expected to ensure that it is covered by an appropriate licence agreement: in many cases an Open Source licence will be suitable (see the section on Licensing).

University requirements for the management of intellectual property are set out in the Code of Practice on Intellectual Property. [REF] You should always seek advice on the appropriate means of publishing your software, to ensure that it is published responsibly.

Students

Students by default own the intellectual property in their own creations, but this ownership may be shared, if they are part of a research team or collaboration, or it may be assigned to another party or be subject to the terms of any contract governing the student's work, such as an industrial sponsorship or employment agreement.

Tips

Bearing in mind these considerations, you should consider the following when publishing software:

- If the software was written collaboratively by staff and students, any students involved will need to give written permission for the software to be published, if they are free to do so, or seek permission for publication from any funding parties with contractually-established rights in their research outputs. Under some circumstances students may assign their IP to the University by formal agreement.
- If the software includes contributions from people who are neither staff nor students of the University, they or their employer will need to give written permission for the software to be published.
- If any part of the software code has been copied from a copyright source, such as a book containing programming recipes, then the permission of the copyright owner to reproduce the code will need to be sought, or the code will need to be removed from the published software.
- Code downloaded from the Web with no stated licence terms is still someone's intellectual property. If you want to use it, you must seek permission from the provider. If you are unable to secure this permission, you cannot assume you have the right to publish this code.

Where permission to publish software needs to be sought, the Research Data Manager or the IP Management team can advise on the correct way to request this permission.

Licensing

Software, like any intellectual property, should always be distributed under a licence. There are two very good reasons for this:

- A licence is an explicit statement of intellectual property rights and makes it easier to protect these rights.
- A licence provides a clear set of permissions and obligations under which the work may be used, so that any users of the work understand what they can and cannot do with it.

Licensing your work need not be difficult or complicated: in most cases the work can be straightforwardly licensed by the application of a standard licence statement. But it does require some forethought to get it right. Two principal considerations should be borne in mind when deciding how to licence your software:

- If you do not own the software or any other software with which it integrates, then your options are likely to be already predetermined to a greater or lesser extent. You may be constrained in your licensing options by the requirements of any collaborating parties in the software or the terms of use of any existing software that you have used.
- Once you understand any operative constraints on your licensing options, you must then decide which among those options is most suitable for your software, given how you intend the software to be used.

Some general guidance on licensing follows, but you should seek advice on the most appropriate licence for your software. Contact the Research Data Manager or the University's IP Management team if you require assistance.

Further guidance on licensing is also available from the Software Sustainability Institute.

Licence options

There are a range of licensing options for software, as for other types of intellectual property, which may be more or less protective of the interests of rights-holders, and more or less restrictive in what they allow others to do with the software. Proprietary licences are always an option, and may be suitable for some purposes, typically to enable commercial exploitation. But much research software is licensed on an Open-Source model, often using one of several standard licences. Open Source licensing encourages collaborative development, re-use and distribution of software, and is not necessarily incompatible with commercial exploitation.

The Software Sustainability Institute has produced a comprehensive guide to [choosing an open-source licence](#), and we recommend researchers consult this when licensing their own code.

Publishing practicalities

It is important to attend to the practicalities of publishing software. As with any publishing activity, there are minimum editorial and production processes to be gone through if the work is to be published to an appropriate standard and distributed in a manner appropriate to its intended audience and use.

These processes can be considered in terms of presentation and documentation, so that the software is easy to read and understand; *packaging*, so that it is easy to use and execute; *preserving and sustaining* the software, so that it remains viable in the long term, and *publishing a software paper*, so that you can promote the software to the wider research community and receive academic credit for your work. Key considerations are outlined below.

Presentation and documentation

- Format your code clearly and consistently, so that it is straightforward to read and navigate.
- Include comments in your code, to explain what it is doing and why.
- Include at least one README file alongside your code files, with basic operational information, e.g. how to compile the software.

- Include a user guide for the software, explaining how it should be run and used. For simple software, this information can be included in a README file.
- Include references to any publications that explain the research problem the software was created to solve and the mathematical basis on which it was built.

For guidance on writing readable source code, consult the Software Sustainability Institute⁴.

Packaging

- Package your code in a directory that reflects its logical structure, and archive it as required in a Zip or tar.gz format that preserves the directory structure when unpacked.
- Ensure the code has a version number and maintain a version history.
- If the code needs to be compiled, can the program be built with a free compiler (e.g. GCC⁵, G++⁶, GFortran⁷)?
- Does the code use a portable build system (e.g. Autotools⁸, CMake⁹)?
- Does the package contain test data to demonstrate application of the program?

Preserving and sustaining software

You will need to consider whether your purpose in providing the software is solely to enable validation of your research results, or to allow the software to be actively re-used, maintained and developed. If your intention is to release software for active development, you must be prepared to support the software and your eventual user community. Code repositories can help you do this, but you should be prepared to offer some support in terms of handling user queries, resolving bugs, releasing updates, etc. The Software Sustainability Institute offers guidance on supporting Open Source software¹⁰.

If you simply intend to ensure the replicability of your research results, it may be sufficient to preserve the software code and documentation files alongside your data using a suitable repository, such as the University's Research Data Repository¹¹ or a general preservation and sharing service, such as Zenodo¹².

If you want your software to be re-used, maintained and developed, you may prefer to use a code repository platform. There are a number of popular services such as GitHub¹³ or Bitbucket¹⁴ available for this purpose. Code repositories enable the maintenance and

⁴ <https://software.ac.uk/resources/guides/writing-readable-source-code>

⁵ <https://gcc.gnu.org/>

⁶ <http://www.cprogramming.com/g++.html>

⁷ <https://gcc.gnu.org/wiki/GFortran>

⁸ https://www.gnu.org/software/automake/manual/html_node/Autotools-Introduction.html

⁹ <https://cmake.org/>

¹⁰ <https://software.ac.uk/resources/guides/supporting-open-source-software>

¹¹ <https://researchdata.gla.ac.uk/>

¹² <https://zenodo.org/>

¹³ <https://github.com/>

¹⁴ <https://bitbucket.org/product>

development of software by multiple developers, and in addition to code hosting typically provide version control, bug tracking, release management, mailing lists and documentation functionality.

The two approaches can be combined, so that your software is made easily accessible for ongoing use and development via a code repository, while milestone versions of the code, such as those used to generate results reported in an article, are archived for the long-term in a preservation service. For example, GitHub has an integration with Zenodo that enables users to archive a version of a code repository to Zenodo, where the files will be preserved and assigned a Digital Object Identifier (DOI), so that the given version can be easily cited and retrieved¹⁵. If your code is maintained in another code repository service, you can archive a version by exporting the code files and documentation and depositing them in your chosen data repository, such as the Enlighten: Research Data.

The Software Sustainability Institute provides guidance on choosing a suitable repository for your software and lists a number of resources that can be considered¹⁶.

Publishing a software paper

Once you have published your software, you may want to use a peer-reviewed article to publish information about it, so that relevant user communities can learn about it and you can receive academic credit for your work.

Where the software is a significant output of your research and has potential for re-use, it can be worth publishing a paper that is primarily focused on the software itself, rather than the research questions it was used to solve. There are a number of both general and subject-specific journals which accept submissions that are primarily about software: the Software Sustainability Institute maintains a list of these journals¹⁷.

¹⁵ <https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content>

¹⁶ <https://software.ac.uk/choosing-repository-your-software-project>

¹⁷ <https://www.software.ac.uk/which-journals-should-i-publish-my-software>

Useful links:

UofG (University of Glasgow) IP and commercialisation

<https://www.gla.ac.uk/myglasgow/ris/ipcommercialisation/inventors/intellectualpropertymanagement/>

Open Source Initiative <https://opensource.org/>

The stewards of the Open Source Definition (OSD) and the community-recognised body for reviewing and approving licenses as OSD-conformant.

Research Data Management Policy https://www.gla.ac.uk/media/Media_817778_smx.pdf

The University of Glasgow's Research Data Management Policy.

Software Sustainability Institute <http://software.ac.uk/>

A national facility for cultivating and improving research software. The website provides guides and resources to assist with management and publication of research software and includes information about training and support offered by the SSI (Software Sustainability Institute).

Choosing an open-source licence <https://www.software.ac.uk/resources/guides/choosing-open-source-licence>

A guide to licensing open-source research software.

choosealicense.com <http://choosealicense.com/>

Dataset Licensing <https://datasetlicencing.wordpress.com/>

User-friendly guides to choosing and using licenses.

TLDR Legal <https://tldrlegal.com/>

A guide to software licences in plain English, provided by legal experts.

Enlighten: Research Data <https://researchdata.gla.ac.uk/>

The University of Glasgow's institutional data repository.

Software publishing checklist

1. Is the software necessary to validate my published research findings?

Yes: you should preserve and publish the software no later than publication of the research findings. You can delay publication or restrict access if there is a need to protect the IP pending commercial exploitation.

No: you do not need to preserve or publish the software.

2. Does the software have commercial value or potential?

Yes: contact the Intellectual Property Manager to discuss commercial exploitation the software.

No: the software may be worth publishing if it has value for non-commercial research uses.

3. Could the software be of use to myself or other researchers in the future?

Yes: you should consider preserving and publishing the software.

No: consider whether other reasons apply for preserving or publishing the software.

4. Do I own the software or have the right to distribute it?

Yes: you can publish the software, subject to any conditions specified in the owner's grant of permission to publish. Where the software has been created in the course of employment by the University and is not subject to any other project-specific contractual conditions, the University will own the software, but researchers have discretion to choose the most suitable publication option. If the software support research findings, in most cases publication under an open licence will be suitable.

No: you must obtain permission from the owner(s) of the software before you can publish it.

5. Does anyone else hold ownership or rights in the software?

Yes: you must obtain permission from any co-owner(s) or other rights-holders in the software before it can be published, and/or clarify the terms on which the software can be made available.

No: you can proceed to publish.

6. Have I decided how to licence my software?

Yes: if you have any concerns, check with the Research Data Manager or Intellectual Property Manager that this licence can be used for your software and is appropriate.

No: consult the Licensing section of this guide and contact the Research Data Manager or Intellectual Property Manager if necessary to discuss your licensing options.

7. Is my software code ready to publish/distribute?

Yes: you can proceed to publish.

No: make sure your code is clearly and consistently formatted, organised and documented, and is packaged with everything necessary to enable users to read and execute the software.

8. Do I want my code to be used and developed by other researchers or software developers?

Yes: use a suitable software development code repository to publish your software, such as the University GitLab platform or GitHub, and consider publishing an article about your software in a dedicated journal.

No: if the software is required solely to validate research results, use a suitable repository for preserving your software, such as Zenodo or the University of Reading Research Data Archive.

Acknowledgement

This guide has been modified from the University of Reading 'Publishing Research Software: A guide for researchers', 2022, which was shared under a CC-BY licence.

