Research Software credit and citation

Purpose: Guidance on research software citation and credit

Audience: UKERC researchers who create research software

Summary

Software is a key part of the research endeavour, and research software creators should be acknowledged, rewarded and credited. One mechanism for this is software citation.

As a research software creator you can help your software to be cited by **formally releasing versions with unique identifiers** and telling users how to cite your software by creating **README file** or a **CITATION file** stored with the software.

As a research software user you can **use citation guidance** in READMEs or CITATION files.

Introduction

Software citation is most effective when good software development practices are followed. It should be noted that research software need not be open source to be cited.

Software Citation Principles

These principles underpin the rationale & requirements for software citation, whilst appreciating that software is a complex area and it will not always be straightforward to apply them.

- **Importance:** legitimate and citable product of research
- **Credit and Attribution:** facilitate giving scholarly credit
- **Unique Identification:** machine actionable, globally unique & interoperable
- **Persistence:** identifiers and metadata should persist beyond the software's lifespan
- **Accessibility:** facilitate access to the software and associated material
- **Specificity:** details of the software version/release used

FAIR for Research Software

Principles to enable research software to be FAIR (Findable, Accessible, Interoperable and Reusable) were published in 2022. These show how the principles which have been applied to research data can be applied to research software.

As a software creator

Stable and persistent version

There needs to be a version which does not change, implying a formal release process –not the version being currently worked on in the code versioning system.

The released version could be deposited in another service such as Zenodo (https://zenodo.org automatically processed from GitHub), Figshare (https://figshare.com) , your local institutional repository or through webpages/repository links to provide a persistent, permanent record of the version. Software Heritage (https://www.softwareheritage.org/) harvests all public source code.

Persistently identified

The software should have a unique and persistent identifier. DOIs are an example of such as scheme and are being widely adopted. Creating a persistent identifier means that basic metadata needs to be created.

License assigned

All software should have a clear license when it is released. Which license is chosen depends on many factors including constraints or dependencies for libraries/other code used and your institutional advice.

Citation File Format and/or READMEs

Enable other people to cite your software by telling them how to do so. There are two methods: putting details in your README files and/or creating a CITATION.cff file.

Citation File Format is a human- and machine-readable file format which provides citation metadata for software. Information on how to create these is available from: https://citation-file-format.github.io/

Examples of accessible research software

- License and a .cff file: van Kuppevelt, D, Meijer, C, Huber, F, van Hees, V, Solino Fernandez, B, Bos, P, van der Ploeg, A. (2020,). **mcfly: deep learning for time series (Version v3.1.0)**. Zenodo. http://doi.org/10.5281/zenodo.3968518.
- License: Wilson, Andy (2020): **Energy Storage System @RISK financial model**. The Open University. Software. https://doi.org/10.21954/ou.rd.13110011.v1

As a software citer

Is there a CITATION file or README which gives the author's guidance? If so, follow that.

Otherwise follow the principles – ensure you credit the project and people involved; include a method of identifying and locating it such as a URL to a release, webpage with metadata, or as a last resort the software repository. Be as precise as possible as to which version was used.

Further Reading

- Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group. (2016) Software Citation Principles. *PeerJ Computer Science* 2:e86. DOI: 10.7717/peerj-cs.86
- Gent I, Jones CM, Matthews B, Guidelines for persistently identifying software using DataCite (2015) http://purl.org/net/epubs/work/31225221
- Chue Hong, N. P., Katz, D. S., Barker, M., Lamprecht, A.-L., Martinez, C., Psomopoulos, F. E., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., Honeyman, T., Struck, A., Lee, A., Loewe, A., van Werkhoven, B., Jones, C., Garijo, D., Plomp, E., Genova, F., … RDA FAIR4RS WG. (2022). FAIR Principles for Research Software (FAIR4RS Principles) (1.0). *Zenodo*. https://doi.org/10.15497/RDA00068
- Guidance on Software for the REF2021 exercise from the Software Sustainability Institute https://www.software.ac.uk/REF2021guidance

Author: Catherine Jones, Aug 2021. Revised Jan 2025.