

Software Citation Checklist for Authors

Produced by the FORCE11 Software Citation Implementation Working Group

Editor: Neil Chue Hong (ORCID: 0000-0002-8876-7606).

Contributors: Alice Allen (ORCID: 0000-0003-3477-2845), Alejandra Gonzalez-Beltran (ORCID: 0000-0003-3499-8262), Anita de Waard (ORCID: 0000-0002-9034-4119), Arfon M. Smith (ORCID: 0000-0002-3957-2474), Carly Robinson (ORCID: 0000-0002-8523-1478), Catherine Jones (ORCID: 0000-0002-5112-835X), Daina Bouquin (ORCID: 0000-0003-2626-3688), Daniel S. Katz (ORCID: 0000-0001-5934-7525), David Kennedy (ORCID: 0000-0002-9377-0797), Gerry Ryder (ORCID: 0000-0001-7444-4489), Jessica Hausman (ORCID: 0000-0002-1861-1526), Lorraine Hwang (ORCID: 0000-0002-1021-3101), Matthew B. Jones (ORCID: 0000-0003-0077-4738), Melissa Harrison (ORCID: 0000-0003-3523-4408), Mercè Crosas (ORCID: 0000-0003-1304-1939), Mingfang Wu (ORCID: 0000-0003-1206-3431), Peter Löwe (ORCID: 0000-0003-2257-0517), Robert Haines (ORCID: 0000-0002-9538-7919), Scott Edmunds (ORCID: 0000-0001-6444-1436), Shelley Stall (ORCID: 0000-0003-2926-8353), Sowmya Swaminathan (ORCID: 0000-0002-0285-4910), Stephan Druskat (ORCID: 0000-0003-4925-7248), Tom Crick (ORCID: 0000-0001-5196-9389), Tom Morrell (ORCID: 0000-0001-9266-5146), Tom Pollard (ORCID: 0000-0002-5676-7898).

DOI: 10.5281/zenodo.3479199

Background

This document provides a simple, generic checklist that authors of academic work (papers, books, conference abstracts, blog posts, etc.) can use to ensure they are following good practice when referencing and citing software they have used, both created by themselves for their research as well as obtained from other sources.

It may also be used and adapted by journal editors, publishers and conference chairs as the basis of more specific guidance for their contributors and reviewers.

Checklist for citing software

- Have I identified the software, including my own, which makes a significant and specialised contribution to my academic work, and therefore should be cited?
- Have I checked if the software has a recommended citation from the creators and used it if available? If this is to a paper, have I also cited the software directly?

- Have I created as complete a citation as possible if no recommended citation is given? Does it include who created the software, when it was created, the title of the software (and version if available) and where the software can be accessed (preferably via a persistent identifier to an archival repository)?
- Have I referenced the software appropriately in my academic work, in compliance with any citation formatting guidelines?

License

This guidance is licensed under the Creative Commons Attribution-ShareAlike 4.0 International license. You are free to share and adapt this material for any purpose, including commercially, as long as you give appropriate credit, provide a link to the license, indicate if changes were made, and you distribute your contributions under the same license as the original. More information: <https://creativecommons.org/licenses/by-sa/4.0/>

Additional Guidance on Software Citation for Authors

Why is software citation important?

Software is part of the research process, across a wide range of disciplines and domains, and it is therefore important that pieces of software that have contributed to the research are cited.

Software citation is vital in the acknowledgment of the crucial role of software in research, as it allows researchers publishing software rather than papers to participate in the academic credit system.

The scholarly communications system that underpins research relies on accurate citations to ensure research integrity, give appropriate credit and show dependencies with other works. As one step towards reproducing a computational procedure, researchers need to be able to access all inputs to the procedure including the specific version of software packages used to produce the original results. Research software should be as open as possible to enable traceability and reproducibility of research results.

Members of the FORCE11 community, representing authors, developers, researchers, publishers and funders, have published the Software Citation Guidelines [1] to encourage broad adoption of a consistent policy for software citation across disciplines and venues. This checklist aims to provide guidance to make it easier to implement these principles as an author.

What software should I cite?

You should cite software (see Appendix 1: Software Terminology) that has a significant impact on the research outcome presented in your work, or on the way the research has been conducted. If the research you are presenting is not repeatable without a piece of software, then you should cite the software. Note that the license or copyright of the software has no bearing on whether you should cite it.

This might include:

- Software (including scripts) you have written yourself to conduct the research presented.
- A software framework / platform upon which the software you wrote to conduct the research relies.
- Software packages, plugins, modules and libraries you used to conduct your research and that perform a critical role in your results.
- Software you have used to simulate or model phenomena/systems.
- Specialist software (which is not considered commonplace in your field) used to prepare, manage, analyse or visualise data.
- Software being evaluated or compared as part of the research presented
- Software that has produced analytic results or other output, especially if used through an interface.

In general, you do not need to cite:

- Software packages or libraries that are not fundamental to your work and that are a normal part of the computational and scientific environment used. These dependencies do not need to be cited outright but should be documented as part of the computational workflow for complete reproducibility.
- Software that was used during the course of the research but had no impact on research results, e.g. word processing software, backup software.

How should I cite software?

Software should be cited in the same way as any other research object: in the list of references.

To identify what citation should be used (i.e. whether it is a specific piece of text, a specific paper, or direct citation of an archive or repository), follow these steps:

1. Determine if the software developers provided a mandatory or recommended citation(s). If so, use it. These citations are often found in a README file, a CITATION file, a CITATION.cff or codemeta.json metadata file, on the software's website, or in its documentation. In some languages and software platforms (e.g., R) a command can be used to generate the

recommended citation. If there is a mandatory or recommended citation, use it.

2. If there is no mandatory or recommended citation provided, use the general principle that a reference should include the following: who, when, what, where. This is similar to the guidance for data. Specific communities may have their own guidelines, for example [4]
 - Who: name the project as the author, unless the individual authorship of the software is clear (e.g., single developer).
 - When: should be the release date of the version you are using or the date you accessed/downloaded the software if using an unreleased version or one where the release date is unclear.
 - What: should be the name of the software, along with specific version / release information. This should be as specific as possible, for instance the name of a package, program or library rather than the platform or programming language it runs on.
 - Where: should be a DOI, URL or other identifier that points to the location of (ideally) a landing page for the software release, or else directly to the software itself. This might be a DOI pointing to an archive in a digital repository, a URL pointing to the code repository, or a URL pointing to the website for the software. Persistent identifiers to archival repositories are preferred over URLs which may change.
3. To further describe the location where the software is preserved, a Software Availability Statement / Code Availability Statement may be required by publishers. This would include the title of the software, the repository location, and additional information needed to access the software such as sensitivity and security issues required by a government, or other entity. This does not replace the citation, but further clarifies access to the software.

If you are the author of the software, you should follow this guidance to generate a suitable citation for your software, and you should put your citation in your software's documentation.

Examples

Example from APA documentation [2]

In text citation:

“We used the Obscure Reference Generator (Version 2.1; Esolang, 2014) and Version 1.0 of Customized Synergy (2014) to complete our work.”

Example reference:

Esolang, A. N. (2014). *Obscure Reference Generator* [Computer software]. Washington, DC: E & K Press.

Example from astronomy - APA style [3]

In text citation:

The scripts and data used to perform the analysis and generate this manuscript are available on GitHub2 and archived in Zenodo (Cruz et al. 2019).

Example footnote:

2. Supplementary materials: <http://github.com/kelle/eudoxus-sup>

Example reference:

Cruz, K. et al. (2014), “Eudoxus”, v1.0, Zenodo, doi.10.5281/zenodo.12345

Example from geosciences - APA style [4]

Example references:

Xu, C., & Christoffersen, B. (2017). *The Functionally-Assembled Terrestrial Ecosystem Simulator*. Version 1. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States). <https://doi.org/10.11578/dc.20171025.1962>. Accessed 2018-07-01. McGibbney,

L.J., Reddy, O., Jarif, I., Spahn, N., & Goodman, A. (2018). *nasa/podaacpy:Podaacpy v2.2.1*. Zenodo. <https://doi.org/10.5281/zenodo.1751973>.

Example from mathematics - APA style [5]

Voevodsky, Vladimir and Ahrens, Benedikt and Grayson, Daniel and others. *UniMath* — a computer-checked library of univalent mathematics. <https://github.com/UniMath/UniMath> [accessed 2018-04-27]

Eigen Project. (2017). *Eigen* [software] version 3.3.4 Available from <https://bitbucket.org/eigen/eigen/> [accessed 2018-04-27]

Vince Knight, & Ria Baldevia. (2018, January 31). *drvinceknight/Nashpy: v0.0.13* (Version v0.0.13). Zenodo. <http://doi.org/10.5281/zenodo.163694>

Additional guidance

Additional guidance is available from <https://research-software.org/citation/researchers/> [6]

References

1. Smith, Arfon M., Katz, Daniel S., Niemeyer, Kyle E., & FORCE11 Software Citation Working Group. (2016). Software citation principles. *PeerJ Computer Science*, 2, e86. <https://doi.org/10.7717/peerj-cs.86>
2. McAdoo, Timothy. (2015). How to Cite Software in APA Style. <https://blog.apastyle.org/apastyle/2015/01/how-to-cite-software-in-apa-style.html>
3. Daina Bouquin, Gus Muench, Kelle Cruz, Daniel Chivvis, Edwin Henneken (2019): Citing Astronomy Software: Inline Text Examples. <https://www.astrobetter.com/blog/2019/07/01/citing-astronomy-software-inline-text-examples/>
4. Hausman, Jessica; Stall, Shelley; Gallagher, James; Wu, Mingfang (2019): Software and Services Citation Guidelines and Examples. Version 1. ESIP. Paper. <https://doi.org/10.6084/m9.figshare.7640426>
5. Katz, DS & Chue Hong, N 2018, Software Citation in Theory and Practice. in *Mathematical Software – ICMS 2018: 6th International Conference*, South Bend, IN, USA, July 24-27, 2018, Proceedings. *Lecture Notes in Computer Science*, vol. 10931, Springer, Cham, pp. 289-296, 6th International Congress on Mathematical Software, Notre Dame, United States, 24/07/18. <https://doi.org/10.1007/978-3-319-96418-8>
6. Druskat, Stephan. (2018). Research software citation for researchers. <https://research-software.org/citation/researchers/>

Appendices

Appendix 1: Software Terminology

Software is the set of instructions to tell a computer how to operate. Software is often composed of reusable modules. Terminological confusion can arise because different communities use different terms for these modules and the frameworks that are used to build them. While other definitions are reasonable, we use the following definitions when referring to software and software components in these recommendations.

- **Program:** a digital object that can be executed by a computer operating system, typically consisting of multiple modules. In a research context, a program is often a standalone application, such as SPSS or Excel, that provides a range of functionality or a tool to do a specific task, such as

- **Module:** a reusable unit of software that can be executed as part of a program
- **Library:** a mechanism for representing a module, usually as a file containing compiled code and data that can be used by programs and other libraries. In a research context, libraries are often used to enable the reusability of algorithm implementations (such as Fast Fourier Transforms) or to
- **Package:** a mechanism for representing a module, usually as a collection of files containing compiled code and data that can be used by programs and other packages
- **Dependency:** a module that is required for execution of some software program
- **Code:** textual representation of a program or module following the syntactic constraints of a programming language
- **Script:** a list of commands, often in the form of a text file, that are executed by another program or framework. In a research context, scripts are often small pieces of code used to automate parts of a workflow, or process data.
- **Framework:** a software environment that provides shared functionality for building software applications through specific extension points. In a research context, frameworks are often
- **Software:** a program, module, library, package, dependency, code, script or framework, i.e. software encompasses all of the above.