# Track 2 Lightning Talk: Should `CITATION` Files Be Standardized?

Stephan Druskat
Department of German Studies and Linguistics
Humboldt-Universität zu Berlin
Berlin, Germany
stephan.druskat@hu-berlin.de

***Abstract*—This paper proposes and discusses the standardization of files that contain reference information for research software: `CITATION` files. While some research software provides files with information on how to reference the software for the purposes of citation, these files vary enormously in their syntax, format, contents, names and extensions. A standardization of `CITATION` files could boost the visibility of the contained information through machine-readability, and consequently possibilities for re-use. A standardization that would preserve human-readability at the same time would provide a compromise between free form text files and more elaborate systems for transitive credit.**

***Index Terms*—Credit, attribution, research software, standardization, format, CITATION files.**

## I. INTRODUCTION

`CITATION` files [1] are computer files containing information on how to correctly cite a (research) software. They often include a bibliographic reference as well as a message outlining some basic terms ("If you use this software in your work, please cite ..."). They may be distributed together with the software's executables, or provided with the its source code in a code repository. In general, they constitute a central strategy for directing, and rendering possible, attribution and credit for research software, as outlined, e.g., by Katz and Smith [2]. Similar to `LICENSE` files, they rely on their discovery and due application. Unlike software licenses, however, which are legally binding and actionable, the intended use of reference information in a `CITATION` file depends on the application of Good Scientific Practice by, e.g., the researcher who uses a software to produce a paper. In this context, it is essential that the information in a `CITATION` file is highly visible for users of a software.

A brief survey of citation files on GitHub (https://github.com/search?q=filename%3Acitation&type=Code&utf8=%E2%9C%93) reveals great variety in syntax, format, contents, names and extensions of `CITATION` files. Some files, most prominently R [3] package citation files, use a syntax, while most are simply free form plain or marked-up text, partly containing BibTeX entries for re-use in TeX bibliographies.

DOI: 10.6084/m9.figshare.3827058.

## II. READABILITY

It seems that in order to create greater visibility for files containing reference information, they should follow a defined syntax and be machine-readable. Machine-readability would allow for their re-use in a number of contexts:

- They can be read at runtime, and their contents formatted and provided to users in the GUI, or in log messages, of the research software itself.
- Software repositories can re-use them to prominently provide citation information to users similar to, e.g., the way GitHub displays license information.[1]
- They can be used as a source for a collection of metadata for a software, e.g., a CodeMeta [4] JSON-LD file.
- They can be used as a source for collecting transitive credit information [5] for a product.
- They can be used as a source for creating citation files for other systems, such as the R package citation system.

While the machine-readability of `CITATION` files is essential to provide possibilities for greater visibility of reference information for research software, they should still remain as human-readable as possible, as they will often be packaged with shipped products and are thus targeted at end-users of differing technical skill sets.

## III. FORMAT

The readability constraints could be fulfilled by using a simple format similar to RIS [6]. RIS is a line-based text format which defines a set of tags for reference information, such as

- an obligatory *type* tag (`TY`) – simultaneously marking the start of a reference entry – for a finite set of defined reference types[2], including `COMP` (computer program);
- the (optional) standard tags for describing the referenced resource, including author(s), title, etc.;
- a single processing tag (`ER`) for marking the end of a reference entry.

---

[1] Depending on where a `CITATION` file is located in the directory tree, reference information could thus be provided granularly, for different levels of the software, e.g., files or packages implementing a citable algorithm, applications and libraries, or containers bundling several applications and libraries.

[2] Thus, RIS could be described as "strongly typed", in contrast to the less strictly typed BibTeX format.

The list of reference tags could be extended, e.g.,

- with an (optionally empty) tag for messages (e.g., `ME`) as often found in `CITATION` files ("If you use this software in your work, please cite …") – this tag could be used for ordering purposes within the file, cf. below;
- with tags specifically targeting the purpose of `CITATION` files, such as a tag for versions (`VE`);
- with tags supporting open science standards, such as ORCiDs instead of – or complementary to – author names;
- even with a tag for a plain-text version of the citation file contents, including pre-formatted references.

Re-using and adapting RIS and its concept in this way, standardized `CITATION` files could contain one or more reference blocks, each starting with a message line, i.e., a line starting with the message tag (`ME`) and providing basic terms.

Each reference block in turn could contain one or more reference entries in the RIS-based format. Each line within an entry, as well as the message line, starts with the (two-character) tag, followed by two whitespaces, followed by a dash, followed by a single whitespace, followed by the value allocated to the tag. This format arguably preserves a passable degree of human-readability, and could even be authored manually:

```
ME  - If you use URPS for your work, please cite:

TY  - COMP
A1  - Stephan Druskat
O1  - 0000-0003-4925-7248
T1  - URPS: Universal Research Problem-Solving Software
VE  - 1.0.42
UR  - https://github.com/sdruskat/urps
DO  - 10.5281/zenodo.840573
ER  -

TY  - JOUR
T1  - A Universal Research Problem-Solving Algorithm
A1  - Brown, Emmett
A1  - La Forge, Geordi
A1  - Druskat, Stephan
PY  - 2017
JF  - Journal of Really Sound Science
VL  - 42
IS  - 11
SP  - 48
EP  - 52
ER  -
```

In addition to the proposed format, files containing reference information should also be consistenly named, e.g., `CITATION` (no file extension, in analogy to `LICENSE` and `NOTICE` files).

## IV. ACCESSIBILITY AND TOOLS

In order to promote a unified name and format for `CITATION` files, communities such as the WSSSPE or RSE community could collaborate in providing tooling for creators of and contributors to research software, in order to smooth the way for wide adoption of standardized `CITATION` files.

In order to advance accessibility of the standardized format, such tooling could include, for example, a web service or platform for an easy creation of standardized `CITATION` files. The community could also provide software libraries for reading `CITATION` file contents in different programming languages for provision via GUIs or logs.

And ideally, third-party services such as repository providers would integrate this or similar tooling to boost visibility for reference information, and thus further promote attribution of and credit for research software.

All in all, standardized `CITATION` files as described above represent a compromise between the commonly used free form citation files as found in code repositories, and a more comprehensive, but also much more elaborate transitive credit system implemented in JSON-LD, which is given as an alternative to `CITATION` files by Smith et al. [2].

## REFERENCES

[1] R. Wilson, "Encouraging citation of software–introducing citation files," 2013, (accessed 17 July 2017). [Online]. Available: https://www.software.ac.uk/blog/2013-09-02-encouraging-citation-software-introducing-citation-files

[2] A. M. Smith, D. S. Katz, K. E. Niemeyer, and FORCE11 Software Citation Working Group, "Software citation principles," *PeerJ Computer Science*, vol. 2, no. e86, 2016. [Online]. Available: https://doi.org/10.7717/peerj-cs.86

[3] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[4] M. B. Jones, C. Boettiger, A. C. Mayes, A. Smith, P. Slaughter, K. Niemeyer, Y. Gil, M. Fenner, K. Nowak, M. Hahnel, L. Coy, A. Allen, M. Crosas, A. Sands, N. C. Hong, P. Cruse, D. Katz, and C. Goble, "CodeMeta: an exchange schema for software metadata. Version 2.0," KNB Data Repository, 2017. [Online]. Available: https://doi.org/10.5063/schema/codemeta-2.0

[5] D. S. Katz and A. M. Smith, "Implementing transitive credit with JSON-LD," *Journal of Open Research Software*, vol. 3, no. e7, 2015. [Online]. Available: https://doi.org/10.5334/jors.by

[6] Thomson-Reuters, *"RIS" format documentation*, 2009, (accessed 17 July 2017). [Online]. Available: http://endnote.com/sites/rm/files/m/direct_export_ris.pdf