The Journal of Open Source Software

FACILE-RS: archival and long-term preservation of research software repositories made easy

Marie Houillon[1], **Jochen Klar**[2], **Ziad Boutanios**[1], **Tomas Stary**[1], **Terry Cojean**[1,3], **Hartwig Anzt**[1,3], **and Axel Loewe**[1]

1 Karlsruhe Institute of Technology, Germany **2** Independent Software Developer, Germany **3** Technical University of Munich, Germany

DOI: 10.21105/joss.07330

Software

- Review
- Repository
- Archive

Editor: Ana Trisovic

Reviewers:

- @exaexa
- @NicolettaNinkovic1

Submitted: 20 June 2024
Published: 24 September 2025

Summary

The Python package FACILE-RS (Findability and Accessibility through Continuous Integration with Less Effort for Research Software) facilitates the archival and long-term preservation of research software repositories. It consists of a set of scripts that simplify the maintenance of software metadata by automating its generation and synchronization in various formats from a single manually maintained metadata file. FACILE-RS also makes it easier to publish and archive software releases according to the Open Science paradigm and the FAIR (Findable, Accessible, Interoperable, Reusable) principles for Research Software by offering tools to automate the creation of releases and the upload to persistent research data repositories.

In particular, FACILE-RS automates:

- Creating a DataCite record (DataCite Metadata Working Group, 2021) based on CodeMeta files (Boettiger, 2017) present in repositories
- Creating a CFF (Citation File Format) file (Druskat et al., 2021) from CodeMeta files
- Creating archive packages in the BagIt (Kunze et al., 2018) or the BagPack (RDA Research Data Repository Interoperability WG, 2019) formats
- Creating a release on the GitLab development platform using the GitLab API
- Archiving software releases persistently on Zenodo
- Archiving software releases persistently using the RADAR service (Kraft et al., 2016)
- Using content from Markdown files, BibTeX files, or Python docstrings to create web pages within the Grav CMS

The scripts can be run manually, but they have been designed to be used within workflow automation systems such as GitLab CI/CD or GitHub Actions, in order to reduce the need for manual intervention when maintaining metadata and creating persistent software releases.

Statement of need

Research software development is a fundamental aspect of academic research (Anzt et al., 2021), and it is now widely acknowledged that the FAIR principles (Wilkinson et al., 2016), established to improve the reusability of research data, should also be applied to research software. However, specific aspects of research software like executability or evolution over time require these guidelines to be adapted. Therefore, the FAIR principles for Research Software (FAIR4RS) have been introduced (Chue Hong et al., 2021).

In particular, reproducible research requires software and associated metadata to be easily findable by both machines and humans, and retrievable via standardised communication protocols. In this context, several metadata standards are widely used across the scientific community:

- The Citation File Format (CFF) is a human- and machine-readable format that indicates how to cite software.
- The DataCite Metadata Schema consists of core metadata properties selected for accurate and consistent identification of research outputs for citation and retrieval purposes, with instructions for recommended use.
- CodeMeta (Jones et al., 2017), an extension of Schema.org, is a JSON and XML metadata schema for scientific software that aims to standardize the exchange of software metadata across repositories and organizations. In particular, it provides mappings between metadata fields used by a large range of software registries and package managers.

All of these standards serve specific purposes, and several are required to cover the whole software lifecycle. However, maintaining multiple metadata files in different formats can be a significant burden for research software developers, and an obstacle to the adoption of good software publication practices. In addition, as the content of the different metadata files is largely overlapping, maintaining these files manually can pose a risk to data consistency.

Another requirement for FAIR scholarly software is that all software releases are published according to the FAIR4RS principles, and assigned a persistent identifier. This can be tedious and prone to errors without an automated process.

Some initiatives were taken in order to overcome these difficulties. For instance, the HERMES project (Druskat et al., 2022) allows to create automated workflows to gather metadata from various sources and publish software on repositories based on InvenioRDM (i.e. Zenodo). The metadata conversion utility Bolognese (Fenner, 2017) supports many metadata formats including CodeMeta and DataCite.

With FACILE-RS, we aim to provide a tool that makes it easy to automate both the creation and maintenance of different metadata formats associated to research software, as well as the publication of software releases according to the FAIR4RS principles on reputable research data repositories.

FACILE-RS has already been integrated in several research software projects such as the openCARP electrophysiology simulator (Augustin et al., 2024; Plank et al., 2021), the 3D visualizer meshalyzer (Vigmond et al., 2023) and DIVAID (Goetz et al., 2023), a tool to divide clinically important regions in bi-atrial geometries.

Functionality

The main prerequisite for using FACILE-RS in a software repository is a CodeMeta metadata file, which can for example be generated using the CodeMeta generator. This tool can also be used to validate a CodeMeta file, as FACILE-RS does not offer a functionality for metadata validation.

FACILE-RS scripts can be run using the `facile-rs` command line tool. The available commands are detailed in Table 1. While each of these commands can be executed individually and manually, FACILE-RS was designed to be used within an automated workflow like GitLab CI/CD pipelines, used for automating software development workflow via a continuous and iterative process.

A typical GitLab CI/CD workflow for FACILE-RS is illustrated in Figure 1. In this example, each time a commit is published, the different metadata files are automatically updated from the CodeMeta file.

Table 1: Components of FACILE-RS.

Script	Functionality
`facile-rs cff create`	generates Citation File Format (CFF) metadata file
`facile-rs datacite create`	generates DataCite metadata file
`facile-rs bag create`	creates BagIt package
`facile-rs bagpack create`	adds DataCite XML to BagIt package
`facile-rs release prepare`	updates *version* and *dateModified* fields in metadata
`facile-rs gitlab publish`	creates release in GitLab
`facile-rs radar prepare`	reserves DOI on RADAR
`facile-rs radar upload`	creates archive and uploads it to RADAR
`facile-rs zenodo prepare`	reserves DOI on Zenodo
`facile-rs zenodo upload`	creates archive and uploads it to Zenodo
`facile-rs grav markdown`	updates Grav CMS website
`facile-rs grav bibtex`	converts BibTeX files and publishes references on Grav CMS website
`facile-rs grav docstring`	extracts docstrings from Python scripts and publishes them on Grav CMS website

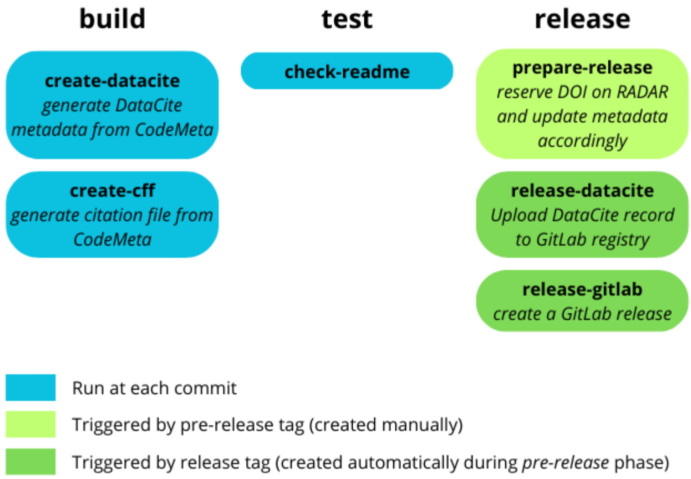
build
test
release
archive
create-datacite
generate DataCite metadata from CodeMeta
create-cff
generate citation file from CodeMeta
check-readme
prepare-release
reserve DOI on RADAR and update metadata accordingly
release-datacite
Upload DataCite record to GitLab registry
release-gitlab
create a GitLab release
archive-radar
Upload release to RADAR
archive-bagpack
Upload BagPack package to Gitlab registry
Run at each commit
Triggered by pre-release tag (created manually)
Triggered by release tag (created automatically during pre-release phase)

Figure 1: Typical structure of an automated FACILE-RS workflow.

This workflow also includes an automated process for creating software releases, both on GitLab and on the research repository RADAR, which is triggered by creating a *pre-release* tag (e.g. tag `pre-v0.1.0` for triggering the pipelines to create release `v0.1.0`). During the *pre-release* phase, a DOI is reserved on RADAR and the software metadata associated with the release is updated. Then, the proper release tag as well as the GitLab and RADAR releases are created automatically.

Such workflows can be integrated easily in existing projects by using the templates we provide for GitLab CI/CD or GitHub Actions. For more information on the implementation of FACILE-RS automated workflows, we refer to the tutorials provided in the FACILE-RS documentation.

Conclusion

In this paper, we present FACILE-RS, a tool to facilitate research software metadata management and archival. FACILE-RS helps researchers follow the FAIR principles for research software through a set of scripts, which can be easily deployed within CI/CD workflows.

Acknowledgements

This work was supported by KIT Strategy Fund Project "FAIR-RS" and the European High-Performance Computing Joint Undertaking EuroHPC under grant agreement No 955495 (MICROCARD) co-funded by the Horizon 2020 programme of the European Union (EU) and the German Federal Ministry of Education and Research.

References

Anzt, H., Bach, F., Druskat, S., Löffler, F., Loewe, A., Renard, B., Seemann, G., Struck, A., Achhammer, E., Aggarwal, P., Appel, F., Bader, M., Brusch, L., Busse, C., Chourdakis, G., Dabrowski, P., Ebert, P., Flemisch, B., Friedl, S., … Weeber, R. (2021). An environment for sustainable research software in Germany and beyond: Current state, open challenges, and call for action. *F1000Research*, *9*(295). https://doi.org/10.12688/f1000research.23224.2

Augustin, C., Azzolin, L., Bayer, J., Bishop, M., Boyle, P. M., Caforio, F., Campos, F., Colin, R., Costa, C. M., Crozier, A., Fastl, T., Gillette, K., Gsell, M., Houillon, M., Huang, Y.-L. (Cary), Huppé, A., Hustad, K. G., Karabelas, E., Krauß, J., … Wülfers, E. M. (2024). *openCARP (v15.0)*. Karlsruhe Institute of Technology (KIT). https://doi.org/10.35097/1979

Boettiger, C. (2017). *Codemeta: A Rosetta Stone for Software Metadata*. https://doi.org/10.6084/m9.figshare.4490588.v1

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. (2021). *FAIR principles for research software (FAIR4RS principles)*. https://doi.org/10.15497/RDA00068

DataCite Metadata Working Group. (2021). *DataCite metadata schema documentation for the publication and citation of research data and other research outputs*. https://doi.org/10.14454/3w3z-sa82

Druskat, S., Bertuch, O., Juckeland, G., Knodel, O., & Schlauch, T. (2022). *Software publications with rich metadata: State of the art, automated workflows and HERMES concept*. https://doi.org/10.48550/arXiv.2201.09015

Druskat, S., Spaaks, J. H., Chue Hong, N., Haines, R., Baker, J., Bliven, S., Willighagen, E., Pérez-Suárez, D., & Konovalov, O. (2021). *Citation File Format* (Version 1.2.0). https://doi.org/10.5281/zenodo.5171937

Fenner, M. (2017). *Bolognese: A ruby library for conversion of DOI metadata*. DataCite. https://doi.org/10.5438/n138-z3mk

Goetz, C., Loewe, A., & Martínez Díaz, P. (2023). *DIVAID (v1.0)*. Karlsruhe Institute of Technology (KIT). https://doi.org/10.35097/1846

Jones, M. B., Boettjiger, C., Mayes, A. C., Smith, A., Slaughter, P., Niemeyer, K., Gil, Y. G., Fenner, M., Nowak, K., Hahnel, M., Coy, L., Allen, A., Crosas, M., Sands, A., Hong, N. C., Cruse, P., Katz, D., & Goble, C. (2017). *CodeMeta: An exchange schema for software metadata. Version 2.0.* https://doi.org/10.5063/schema/codemeta-2.0

Kraft, A., Razum, M., Potthoff, J., Porzel, A., Engel, T., Lange, F., Van den Broek, K., & Furtado, F. (2016). The RADAR project—a service for research data archival and publication. *ISPRS International Journal of Geo-Information*, *5*(3). https://doi.org/10.3390/ijgi5030028

Kunze, J. A., Littman, J., Madden, L., Scancella, J., & Adams, C. (2018). *The BagIt File Packaging Format (V1.0)* (No. 8493). RFC 8493; RFC Editor. https://doi.org/10.17487/RFC8493

Plank, G., Loewe, A., Neic, A., Augustin, C., Huang, Y.-L., Gsell, M. A. F., Karabelas, E., Nothstein, M., Prassl, A. J., Sánchez, J., Seemann, G., & Vigmond, E. J. (2021). The openCARP simulation environment for cardiac electrophysiology. *Computer Methods and Programs in Biomedicine*, *208*, 106223. https://doi.org/10.1016/j.cmpb.2021.106223

RDA Research Data Repository Interoperability WG. (2019). *Research Data Repository Interoperability WG Final Recommendations*. Zenodo. https://doi.org/10.15497/RDA00025

Vigmond, E., Francesco, G. de, Neic, A., Huang, Y.-L. (Cary), & Loewe, A. (2023). *Meshalyzer (5.2)*. Karlsruhe Institute of Technology (KIT). https://doi.org/10.35097/1799

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., & others. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 1–9. https://doi.org/10.1038/sdata.2016.18