Nine Best Practices for Research Software Registries and Repositories: A Concise Guide

Nine Best Practices for Research Software Registries and Repositories: A Concise Guide

Task Force on Best Practices for Software Registries[*]

23 December 2020

Scientific software registries and repositories serve various roles in their respective disciplines. These resources improve software discoverability and research transparency, provide information for software citations, and foster preservation of computational methods that might otherwise be lost over time, thereby supporting research reproducibility and replicability. However, developing these resources takes effort, and few guidelines are available to help prospective creators of registries and repositories. To address this need, we present a set of nine best practices that can help managers define the scope, practices, and rules that govern individual registries and repositories. These best practices were distilled from the experiences of the creators of existing resources, convened by a Task Force of the *FORCE11 Software Citation Implementation Working Group* during the years 2019–2020. We believe that putting in place specific policies such as those presented here will help scientific software registries and repositories better serve their users and their disciplines.

Contents

[*]Corresponding authors: Alice Allen, Mike Hucka, and Tom Morrell.
This Task Force was convened by the *FORCE11 Software Citation Implementation Working Group*.

Introduction

Scientific software registries and repositories serve various roles in their respective disciplines. *Registries* are typically indexes or catalogs of software stored elsewhere, while *repositories* are both indexes *and* places where software is stored. Both types of resource improve software discoverability and research transparency, provide information for software citations, and foster preservation of computational methods that might otherwise be lost over time, thereby supporting research reproducibility and replicability. Many provide or are integrated with other services, including indexing and archival services, which can be leveraged by librarians, digital archivists, journal editors and publishers, and researchers alike.

Having specific policies in place for software registries and repositories ensures that users and administrators have reference documents to help define a shared understanding of the scope, practices, and rules that govern these collections. These practices can prove useful in a variety of situations, including, but not limited to, presenting the contents in the resource to stakeholders and community members, reassuring potential contributors by clarifying sensitive issues such as attribution, and defining how content in a registry or repository can be (re)used by others.

The best practices presented here were proposed and developed by a Task Force of the *FORCE11 Software Citation Implementation Working Group*. The members of the Task Force were managers and editors of scientific software registries and repositories. Development of the best practices began with a series of monthly conference calls in 2019 and continued at the *Scientific Software Registry Collaboration Workshop*, a two-day workshop held at the University of Maryland in November, 2019, with generous funding from the Sloan Foundation. In 2020, the Task Force made additional refinements to the best practices during virtual meetings and through online collaborative writing. The Authors section lists the people who participated.

Each guideline is presented below with an explanation as to why we recommend the practice, what the practice describes or contains, and specific considerations to take into account. Our recommendations are partitioned into nine separate policies or statements, though there is inescapable overlap between some of them. In practice, the statements and policies are often combined into a smaller number of documents, as is evident in most of the real-world examples presented at the end of this document. To reduce repetition in the descriptions of the guidelines, we often refer to registries and repositories collectively as “resources” and “collections.”

These nine best practices, though not an exhaustive list, are applicable to the varied resources represented in the Task Force, so are likely to be broadly applicable to other scientific software repositories and registries. We believe that adopting these practices will help document, guide, and preserve these resources, and put them in a stronger position to serve their disciplines, users, and communities.

Best Practice: Provide a public scope statement

Why we recommend this: A scope statement clarifies the type of software contained in the repository or indexed in the registry. This manages the expectations of the potential depositor of metadata and/or software, as well as the resource seeker. It informs both of what the collection does and does not contain.

This should describe:

- What is accepted, and acceptable
- What is not accepted
- Exceptions to either/both of the above if necessary

What you might consider when writing a scope statement:

- Defining the community being served
- The types of software listed in the registry or stored in the repository, such as source code or compiled executables
- Criteria that must be satisfied by accepted software, such as whether certain software quality metrics must be fulfilled or whether the software must be used in published research
- Whether the code has to be in the public domain and/or have a license from a predefined set
- Whether software registered in another registry or repository will be accepted

Best Practice: Provide guidance for users

Why we recommend this: Different users of the registry or repository will benefit from having guidance on how to access the information they are interested in. For example, it is useful to describe how to search the collection, answer frequently asked questions (FAQs), provide tips and tricks, and to let users know who to contact for assistance.

A separate section in these guidelines on the *Conditions of use policy* covers terms of use of the collection, including data and API, and how best to cite records in the resource and the resource itself. Guidance for users who wish to contribute software is covered in the next section, *Provide guidance to software contributors*.

This should describe:

- How to perform common user tasks
- Answers to questions that are often asked or can be anticipated
- Whom to contact for questions or help

What you might consider when writing guidance for users:

- Identifying the types of users your resource has or could potentially have, and corresponding use cases
- Offering multiple forms of guidance, such as in-field prompts, linked explanations, and completed examples
- If there is an API, including a description specifying the interface or a pointer to the official documentation for the interface
- If content negotiation is enabled, stating what formats, such as JSON-LD or XML, are supported

Best Practice: Provide guidance to software contributors

Why we recommend this: People interested in contributing software entries to the registry or repository need to know what the process entails. The scope statement will already have explained *what* is accepted and what is not; the contributor policy addresses *who* can add or change software entries, and the processes involved.

This should describe:

- Who can or cannot submit entries and/or metadata
- Required and optional metadata expected from software contributors
- Review process, if any
- Curation process, if any
- Procedures for updates (e.g., who can do it, when it is done, how is it done)

What you might consider when writing a contributor policy:

- Defining who can submit and/or update entries
- Whether the author(s) of the software will be contacted if the contributor is not also an author, and whether contact is a condition or side-effect of the submission
- Stating how persistent identifiers are assigned (if they are used)
- Including a statement that depositors must comply with all applicable laws and not be intentionally malicious

Best Practice: Establish an authorship policy

Why we recommend this: Establishing a policy dedicated to authorship ensures that people are given due credit for their work. It also serves as a document that administrators can turn to in case authorial disputes arise and allows for proactive problem mitigation, rather than having to resort to reactive interpretation. Further, having an authorship policy is in keeping with similar policies by journals and publishers. Having such explicit authorship policies is thus part of a larger trend. Note that the authorship policy will be communicated at least partially to users through guidance provided to software contributors.

This should describe:

- Who should be listed as an author of the software
- Policies around making changes to authorship
- How authorship disputes are handled
- What the resource will do in case of conflict

What you might consider when writing an authorship policy:

- Taking into consideration whether those who are not coders, such as software testers or documentation maintainers, will be identified or credited as authors, as well as criteria for ordering the list of authors in cases of multiple authors
- How the resource handles large numbers of authors and group or consortium authorship
- Including guidelines about how changes to authorship are handled
- What role the registry will play, if any, in authorship disputes, and if so, how they are handled
- Maintaining consistency with the citation policies for the registry/repository
- Using a credit ontology (*e.g.*, https://casrai.org/credit/) to describe authors' contributions

Best Practice: Share your metadata schema

Why we recommend this: For individual and organizational users interested in the information in registries and repositories, revealing the metadata schema used for the entries helps users understand the structure and properties of the deposited information. The metadata structure helps to inform users how they might interact with or ingest records in the collection. A metadata schema mapped to other schemas and an API specification can improve the interoperability between registries and repositories.

This should describe:

- What schema is used (e.g., *CodeMeta*, *Schema.org*) and its version number if a published standard schema is used, or, if a custom schema is used, a description of the schema and/or a data dictionary
- Where the metadata documentation or its official site can be found
- What metadata is expected when submitting software, including which fields are required and which are optional, and the format of the content in each field

What you might consider when stating your metadata schema:

- Using a metadata schema that is mapped ("cross-walked") to published standard schemas, or providing a cross-walk between your schema and other schemas
- Providing an example of the metadata schema with a complete entry in your repository that illustrates all the fields of the schema

Best Practice: Stipulate conditions of use

Why we recommend this: A conditions of use policy lets users of your resource know how the metadata of the registry or repository can be used, attributed, and/or cited. It provides information about licensing and forestalls potential liabilities and difficulties that may arise, such as claims of damage for misinterpretation or misapplication of metadata. In turn, it clearly states how the metadata can and cannot be used, including for commercial purposes and in aggregate form.

This should describe:

- Legal disclaimers about the responsibility and liability borne by the registry or repository
- License and copyright information, both for individual entries and for the registry or repository as a whole
- Conditions for the use of the metadata, including prohibitions, if any
- Preferred format for citing software entries
- Preferred format for attributing or citing the resource itself

What you might consider when writing a conditions of use policy:

- What license governs your metadata, and whether there are licensing requirements for findings and/or derivatives of the resource
- Whether there are differences in the terms and license for commercial versus noncommercial use
- Conditions for the use of the API if one is available
- Restrictions on use of the metadata
- Including a statement to the effect that the registry or repository makes no guarantees about completeness and is not liable for any damages that may arise from the use of the information

Best Practice: State a privacy policy

Why we recommend this: Having a privacy policy demonstrates a strong commitment to the privacy of users of the registry or repository, and allows the resource to comply with the legal requirement of many countries in addition to those a home institution and/or funding agencies may impose. A privacy policy discloses what information, analytics, and metrics a registry collects and/or retains about its users and why.

This should describe:

- What information is collected and how long it is retained
- How the information, especially any personal data, is used
- Whether tracking is done, what is tracked, and how (e.g., Google Analytics)
- Whether cookies are used

What you might consider when writing a privacy policy:

- Detailing the specific data collected, why it is collected, and whether it is shared or sold
- Being explicit about third party tools used to collect analytic information and potentially referencing their privacy policies
- Stating whether users will receive email as a result of visiting or downloading content
- Explaining the measures taken to protect users' privacy, and whether the resource complies with the *European Union Directive on General Data Protection Regulation* (GDPR) or other local laws, if applicable
- Reserving the right to make changes to the Privacy Policy
- Defining a mechanism by which users can request information be removed

Best Practice: Provide a retention policy

Why we recommend this: Software registries and repositories make an implicit promise to retain records for some period of time, but for various reasons may have to remove records. Common examples include removing entries that are outdated or no longer meet the scope of the registry or are found to be in violation of policies. The collection should document retention goals so that users and depositors are aware of them.

This should describe:

- The length of time metadata and/or files are expected to be retained
- Under what conditions metadata and/or files are removed
- Who has the responsibility and ability to remove information
- Procedures to request that metadata and/or files be removed

What you might consider when writing a retention policy:

- If assigning identifiers, whether best practices for persistent identifiers are followed, including resolvability, retention, and non-reuse of those identifiers
- Making sure the length of time is not too prescriptive (e.g., “for the next 10 years”), but rather fits within the context of the underlying organization(s) and its funding
- Stating who is allowed to edit metadata, delete records, or delete files, and if so, how these changes are documented and consistent with the registry broadly
- Explaining the process by which data may be taken offline and archived as well as the process for its possible retrieval

Best Practice: Disclose your end-of-life policy

Why we recommend this: Sharing a clear end-of-life policy increases trust in the community served by your registry or repository. It demonstrates a thoughtful commitment to users by informing them that provisions for the artifacts contained in the resource have been considered should the resource close or otherwise end its services for these artifacts. Such a policy sets expectations and provides reassurance as to how long the records within the resource will be findable and accessible in the future.

This should describe:

- Under what circumstances the resource might end its services
- What consequences would result from closure
- What will happen to the metadata and/or the software artifacts contained in the resource in the event of closure
- If long-term preservation is expected, where metadata and/or software artifacts will be migrated for preservation
- How a migration will be funded

What you might consider when writing a end-of-life policy:

- Whether the records will remain available, and if so, how and for whom, and under which conditions, such as archived status or “read only”, should the collection close
- What restrictions, if any, may apply
- Establishing a formal agreement or MOU with another registry, repository, or institution to receive and preserve the data or project, if applicable

Policy examples

Scope Statement

- Astrophysics Source Code Library. (n.d.). *Editorial policy.* https://ascl.net/wordpress/submissions/editiorial-policy/

- bio.tools. (n.d.). *Curators Guide.* https://biotools.readthedocs.io/en/latest/curators_guide.html

- Caltech Library. (2017). *Terms of Deposit.* https://data.caltech.edu/terms

- Caltech Library. (2019). *CaltechDATA FAQ.* https://www.library.caltech.edu/caltechdata/faq

- Computational Infrastructure for Geodynamics. (n.d.). *Code Donation.* https://geodynamics.org/cig/dev/code-donation/

- CoMSES Net Computational Model Library. (n.d.). *Frequently Asked Questions.* https://www.comses.net/about/faq/#model-library

- ORNL DAAC for Biogeochemical Dynamics. (n.d.). *Data Scope and Acceptance Policy.* https://daac.ornl.gov/submit/

- RDA Registry and Research Data Australia. (2018). *Collection.* ARDC Intranet. https://intranet.ands.org.au/display/DOC/Collection

- Remote Sensing Code Library. (n.d.). *Submit.* https://rscl-grss.org/submit.php

- SciCrunch. (n.d.). *Curation Guide for SciCrunch Registry.* https://scicrunch.org/page/Curation%20Guidelines

- U.S. Department of Energy: Office of Scientific and Technical Information. (n.d.-a). *DOE CODE: Software Policy.* https://www.osti.gov/doecode/policy

- U.S. Department of Energy: Office of Scientific and Technical Information. (n.d.-b). *FAQs.* OSTI.GOV. https://www.osti.gov/faqs

Authorship

- CASRAI. (n.d.). CRediT - Contributor Roles Taxonomy. https://casrai.org/credit/

- Committee on Publication Ethics: COPE. (2020a). *Authorship and contributorship.* https://publicationethics.org/authorship

- Committee on Publication Ethics: COPE. (2020b). *Core practices.* https://publicationethics.org/core-practices

- Dagstuhl EAS Specification Draft. (2016). *The Software Credit Ontology.* https://dagstuhleas.github.io/SoftwareCreditRoles/doc/index-en.html#

- Journal of Open Source Software. (n.d.). *Ethics Guidelines.* https://joss.theoj.org/about#ethics

- ORNL DAAC (n.d) *Authorship Policy.* https://daac.ornl.gov/submit/
- PeerJ Journals. (n.d.-a). *Author Policies.* https://peerj.com/about/policies-and-procedures/#author-policies
- PeerJ Journals. (n.d.-b). *Publication Ethics.* https://peerj.com/about/policies-and-procedures/#publication-ethics
- PLOS ONE. (n.d.). *Authorship.* https://journals.plos.org/plosone/s/authorship
- National Center for Data to Health. (2019). The Contributor Role Ontology. https://github.com/data2health/contributor-role-ontology

Metadata Schema

- ANDS: Australian National Data Service. (n.d.). *Metadata*. ANDS. https://www.ands.org.au/working-with-data/metadata
- ANDS: Australian National Data Service. (2016). *ANDS Guide: Metadata.* https://www.ands.org.au/__data/assets/pdf_file/0004/728041/Metadata-Workinglevel.pdf
- Bernal, I. (2019). *Metadata for Data Repositories.* https://doi.org/10.5281/zenodo.3233486
- bio.tools. (2020). *Bio-tools/biotoolsSchema* [HTML]. https://github.com/bio-tools/biotoolsSchema (Original work published 2015)
- bio.tools. (2019). *BiotoolsSchema documentation.* https://biotoolsschema.readthedocs.io/en/latest/
- The CodeMeta crosswalks. (n.d.) https://codemeta.github.io/crosswalk/
- Citation File Format (CFF). (n.d.) https://doi.org/10.5281/zenodo.1003149
- The DataVerse Project. (2020). DataVerse 4.0+ Metadata Crosswalk. https://docs.google.com/spreadsheets/d/10Luzti7svVTVKTA-px27oq3RxCUM-QbiTkm8iMd5C54
- OntoSoft. (2015). *OntoSoft Ontology.* https://ontosoft.org/ontology/software/
- OpenAPI Specification. (2020). http://spec.openapis.org/oas/v3.0.3
- Zenodo. (n.d.-a). *Schema for Depositing.* https://zenodo.org/schemas/records/record-v1.0.0.json
- Zenodo. (n.d.-b). *Schema for Published Record.* https://zenodo.org/schemas/deposits/records/legacyrecord.json

Conditions of use policy

- Allen Institute. (n.d.). *Terms of Use.*
  https://alleninstitute.org/legal/terms-use/

- Europeana. (n.d.). *Usage Guidelines for Metadata.* Europeana Collections.
  https://www.europeana.eu/portal/en/rights/metadata.html

- U.S. Department of Energy: Office of Scientific and Technical Information. (n.d.). *DOE CODE FAQ: Are there restrictions on the use of the material in DOE CODE?*
  https://www.osti.gov/doecode/faq#are-there-restrictions

- Zenodo. (n.d.). *Terms of Use.*
  https://about.zenodo.org/terms/

Privacy policy

- Allen Institute. (n.d.). *Privacy Policy.*
  https://alleninstitute.org/legal/privacy-policy/

- CoMSES Net. (n.d.). *Data Privacy Policy.*
  https://www.comses.net/about/data-privacy/

- Nature. (2020). *Privacy Policy.*
  https://www.nature.com/info/privacy

- Research Data Australia. (n.d.). *Privacy Policy.*
  https://researchdata.ands.org.au/page/privacy

- SciCrunch. (2018). *Privacy Policy.* SciCrunch.
  https://scicrunch.org/page/privacy

- Science Repository. (n.d.). *Privacy Policies.*
  https://www.sciencerepository.org/privacy

- Zenodo. (n.d.). *Privacy policy.*
  https://about.zenodo.org/privacy-policy/

Retention Policy

- Caltech Library. (n.d.). *CaltechDATA FAQ.*
  https://www.library.caltech.edu/caltechdata/faq

- CoMSES Net Computational Model Library. (n.d.). *How long will models be stored in the Computational Model Library?*
  https://www.comses.net/about/faq/

- Dryad. (2020). *Dryad FAQ - Publish and Preserve your Data.*
  https://datadryad.org/stash/faq#preserved

- Software Heritage. (n.d.). *Content policy.*
  https://www.softwareheritage.org/legal/content-policy/

- Zenodo. (n.d.). *General Policies v1.0.*
  https://about.zenodo.org/policies/

- Bioconductor. (2020). *Package End of Life Policy.*
  https://bioconductor.org/developers/package-end-of-life/

End-of-life policy

- Figshare. (n.d.). *Preservation and Continuity of Access Policy.* https://knowledge.figshare.com/articles/item/preservation-and-continuity-of-access-policy

- Open Science Framework. (2019). *FAQs.* OSF Guides. http://help.osf.io/hc/en-us/articles/360019737894-FAQs

- NASA Earth Science Data Preservation Content Specification (n.d.) https://earthdata.nasa.gov/esdis/eso/standards-and-references/preservation-content-spec

- Zenodo. (n.d.). *Frequently Asked Questions.* https://help.zenodo.org/

Additional useful sites

In addition to the links to sites and information embedded in this Concise Guide, the following sites are directly applicable to the best practices we have listed.

- (Authorship) Citation File Format: https://citation-file-format.github.io/
- (Authorship) CiteAs: https://citeas.org/
- (Metadata Schema) Software Heritage Metadata workflow: https://docs.softwareheritage.org/devel/swh-indexer/metadata-workflow.html
- (Metadata Schema) W3C data profile definition: https://www.w3.org/TR/dx-prof-conneg/#dfn-data-profile

You may also be interested in the https://www.coretrustseal.org/why-certification/requirements/, which are intended to reflect the characteristics of trustworthy repositories.

Glossary

API: Application Programming Interface

Collection: Used in this document as a synonym for *registries and repositories*

Depositor: A user who submits information and/or software to a registry or repository; synonymous with *software contributor*

Entry: Information about and/or software for a particular holding in a registry or repository; synonymous with *record*

JSON-LD: JavaScript Object Notation for Linked Data

Metadata: Information about a code or software package

Record: Information about and/or software for a particular holding in a registry or repository; synonymous with *entry*

Registry: Typically an index or catalog of software stored elsewhere

Repository: Typically a site that both indexes and stores software

Resource: Used in this document as a synonym for *registries and repositories*

Software author: A person who is credited as an author of a software package; this may include not only one who writes code, but also one who tests, documents, maintains, or otherwise contributes effort to the software package

Software contributor: A user who submits information and/or software to a registry or repository; synonymous with *depositor*

XML: Extensible Markup Language

Authors

Alain Monteil, INRIA, HAL/Software Heritage

Alejandra Gonzalez-Beltran, Science and Technology Facilities Council, UK Research and Innovation

Alexandros Ioannidis, CERN, Zenodo

Alice Allen, University of Maryland, Astrophysics Source Code Library

Allen Lee, Arizona State University, CoMSES Net

Anita Bandrowski, UCSD, SciCrunch

Bruce E. Wilson, Oak Ridge National Laboratory, ORNL Distributed Active Archive Center for Biogeochemical Dynamics

Bryce Mecum, NCEAS, UC Santa Barbara, CodeMeta

Cai Fan Du, iSchool, University of Texas at Austin, CiteAs

Carly Robinson, DOE-OSTI

Daniel Garijo, Information Sciences Institute, University of Southern California, Ontosoft

Daniel S. Katz, University of Illinois at Urbana-Champaign, Associate EiC for JOSS, FORCE11 Software Citation Implementation Working Group co-chair

David Long, Brigham Young University, IEEE GRS Remote Sensing Code Library

Genevieve Milliken, NYU Bobst Library, IASGE

Hervé Ménager, Institut Pasteur, ELIXIR bio.tools

Jessica Hausman, Jet Propulsion Laboratory, PO.DAAC

Jurriaan H. Spaaks, Netherlands eScience Center, Research Software Directory

Katrina Fenlon, University of Maryland, iSchool

Kristin Vanderbilt, Environmental Data Initiative, IMCR

Lorraine Hwang, Computational Infrastructure for Geodynamics, UC Davis

Lynn Davis, DOE-OSTI

Martin Fenner, DataCite, FORCE11 Software Citation Implementation Working Group co-chair

Michael R. Crusoe, CWL, Debian-Med

Mike Hucka, Caltech, SBML, COMBINE

Mingfang Wu, Australian Research Data Commons

Neil Chue Hong, Software Sustainability Institute, University of Edinburgh, FORCE11 Software Citation Implementation Working Group co-chair

Peter Teuben, University of Maryland

Shelley Stall, American Geophysical Union, AGU Data Services

Stephan Druskat, German Aerospace Center (DLR)/University Jena/Humboldt-Universität zu Berlin, Citation File Format

Ted Carnevale, Neuroscience Department, Yale University, ModelDB

Tom Morrell, Caltech, CaltechDATA