

NLeSC Software Quality tools and some **personal** opinions.

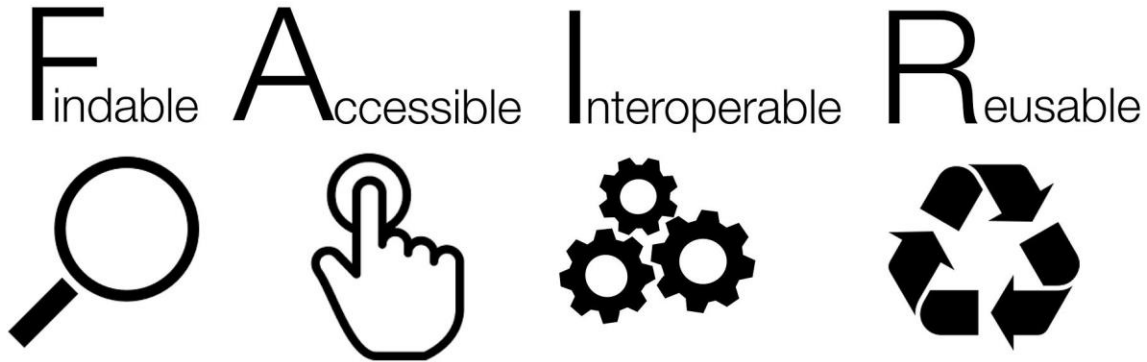
Faruk Diblen

10 October 2022

netherlands
eScience center

<https://doi.org/10.5281/zenodo.7193991>





FAIR Principles

<https://www.go-fair.org/fair-principles/>

The FAIR Guiding Principles for scientific data management and stewardship

<https://www.nature.com/articles/sdata201618>

WG **FAIR for Research Software (FAIR4RS) WG**
Taxonomy:

Posts | Create Wiki Index | Events | Repository | Outputs | Case Statements | Plenaries | Members | create new content

Group Status: WGs Maintaining deliverables (maintenance group) Join Group

Please make sure the group follows the new **RDA Groups Policy**, which came into effect on 1 April 2021. Please contact [enquiries\[at\]rd-alliance.org](mailto:enquiries[at]rd-alliance.org) if you have any questions.

Status: Recognised & Endorsed

Chair (s):

Michelle Barker, Paula Andrea Martinez, Leyla Jael Castro, Daniel S. Katz, Neil Chue Hong, Jennifer Harrow, Fotis Psomopoulos, Carlos Martinez-Ortiz, Morane Gruenpeter

Secretariat Liaison: Bridget Walker

TAB Liaison: Rob Quick

News:

May 24th, 2022. The RDA Council have endorsed the FAIR4RS Principles as an official output

Citation and download: 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., et al. (2022). FAIR Principles for Research Software version 1.0. (FAIR4RS Principles v1.0). Research Data Alliance. DOI: <https://doi.org/10.15497/RDA00068>

This Working Group is wrapping up! Thanks for your contributions!!!

The RDA Software Source Code Interest Group is the maintenance home for the principles. Concerns or queries about the principles can be raised at RDA plenary events organised by the SSC IG, where there may be opportunities for adopters to report back on progress.

FAIR data is different than FAIR software

FAIR for Research Software (FAIR4RS) WG

<https://www.rd-alliance.org/groups/fair-4-research-software-fair4rs-wg>

FAIR Principles for Research Software (FAIR4RS Principles)

<https://zenodo.org/record/6623556>

<https://doi.org/10.5281/zenodo.7193991>





FIVE RECOMMENDATIONS FOR FAIR SOFTWARE

ENDORSE

LET'S GO! →



WHAT'S FAIR

ABOUT US

<https://fair-software.nl>



#1 USE A PUBLICLY ACCESSIBLE REPOSITORY WITH VERSION CONTROL



WHY THIS IS IMPORTANT

HELP ME CHOOSE

<https://fair-software.nl>



#2 ADD A LICENSE



WHY THIS IS IMPORTANT

HELP ME CHOOSE

<https://fair-software.nl>

- @NLeSC we prefer Apache 2.0 for software
- No default for data but usually CC BY (<https://creativecommons.org/about/ccllicenses/>)
- We deviate only when it is necessary

Apache License 2.0

A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions

- Commercial use
- Distribution
- Modification
- Patent use
- Private use

Conditions

- License and copyright notice
- State changes

Limitations

- Liability
- Trademark use
- Warranty

Choose an open source license

An open source license protects contributors and users. Businesses and savvy developers won't touch a project without this protection.

{ Which of the following best describes your situation? }



I need to work in a community.

Use the **license preferred by the community** you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to **add a license**.



I want it simple and permissive.

The **MIT License** is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

Babel, **.NET**, and **Rails** use the MIT License.



I care about sharing improvements.

The **GNU GPLv3** also lets people do almost anything they want with your project, *except* distributing closed source versions.

Ansible, **Bash**, and **GIMP** use the GNU GPLv3.

{ What if none of these work for me? }

My project isn't software.

[There are licenses for that.](#)

I want more choices.

[More licenses are available.](#)

I don't want to choose a license.

[Here's what happens if you don't.](#)

The content of this site is licensed under the Creative Commons Attribution 3.0 Unported License.

About Terms of Service Help improve this page
Curated with ❤️ by GitHub, Inc. and You!

<https://choosealicense.com/>

<https://doi.org/10.5281/zenodo.7193991>

License compliance is important!

The \$100 Million Court Case for Open Source License Compliance

- CoKinetic Systems Corporation, one of the major global players in the in-flight entertainment (IFE) market, has recently filed suit against Panasonic Avionics Corporation in a New York federal court, seeking damages of over \$100 million.
- CoKinetic claim that Panasonic, which holds a roughly 70% share of the embedded IFE hardware market, has willfully violated GPL v2 open source licensing requirements, in addition to a long list of other unlawful actions aimed at monopolizing the market for in-flight entertainment software and media services.

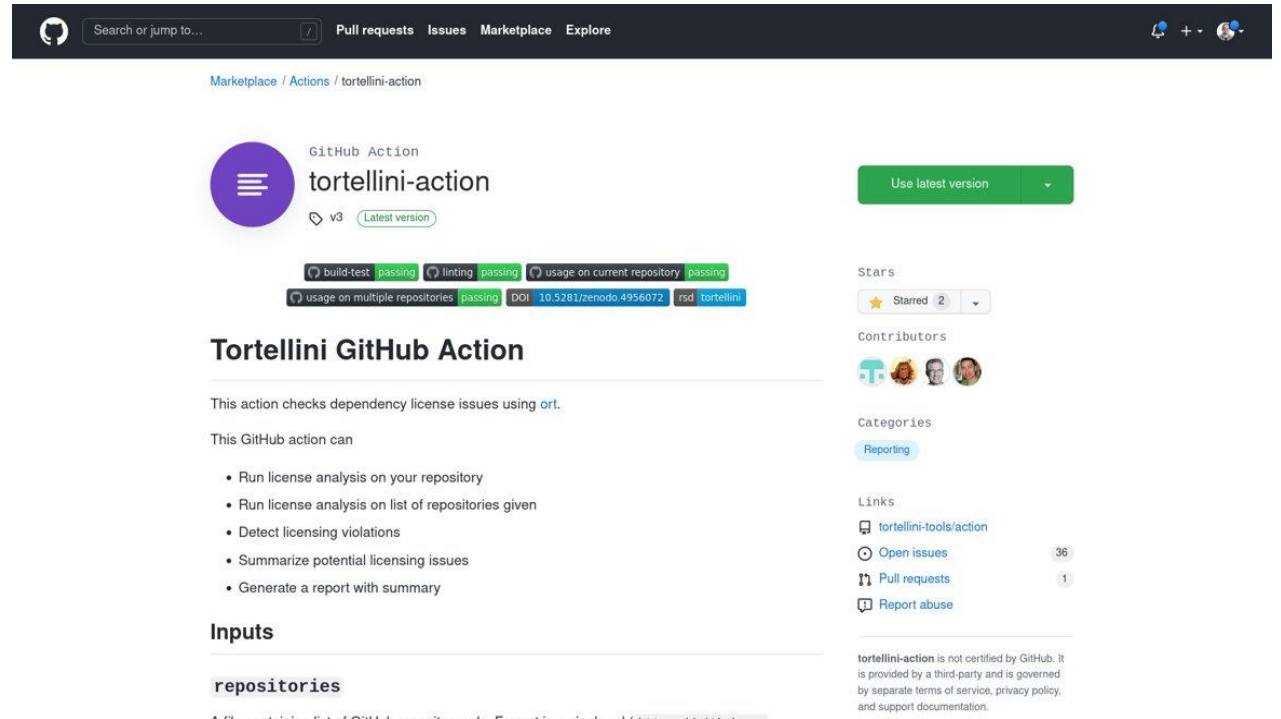
Vizio sued for breach of copyleft open-source software license

- On October 19, 2021, the Software Freedom Conservancy (SFC) sued Vizio, Inc. for alleged violations of the GNU General Public License covering software incorporated into certain Vizio smart TVs.

More examples:

- https://en.wikipedia.org/wiki/Open_source_license_litigation
- <https://www.slashdata.co/blog/the-open-source-trials-hanging-in-the-legal-balance-of-copyright-and-copyleft>

Example: Akurat Font




OSS
Review Toolkit



- <https://github.com/oss-review-toolkit/ort>
- <https://github.com/tortellini-tools/action>
- <https://github.com/marketplace/actions/tortellini-action>

<https://doi.org/10.5281/zenodo.7193991>



#3 REGISTER YOUR CODE IN A COMMUNITY REGISTRY



WHY THIS IS IMPORTANT

HELP ME CHOOSE

<https://fair-software.nl>

research-software.nl

netherlands eScience center

Software Projects Metrics

This website is powered by the [Research Software Directory](#) – the content management system for research software.

Start typing here to search for software

Sort by: Last updated

Tags

- Big data (37)
- GPU (6)
- High performance computing (30)
- Image processing (5)
- Inter-operability & linked data (15)
- Machine learning (17)
- Multi-scale & multi model simulations (8)
- Optimized data handling (26)
- Real time data analysis (6)
- Text analysis & natural language processing (17)
- Visualization (27)
- Workflow technologies (28)

Organizations

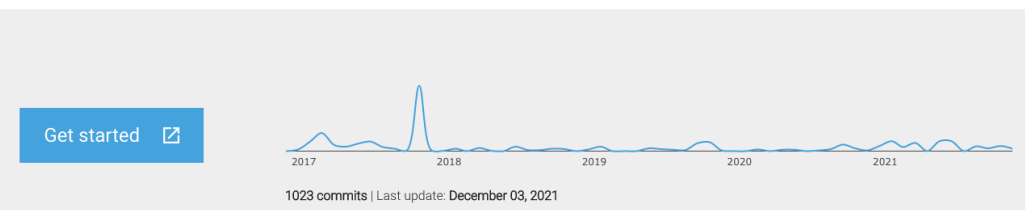
- Activinsights Ltd (1)
- Alfred Wegener Institute (2)
- ASTRON (6)
- Barcelona Supercomputing Center (2)

<p>Kernel Tuner Ke</p> <p>Kernel Tuner greatly simplifies the development of highly-optimized and auto-tuned CUDA, OpenCL, and C code, supporting many advanced use-cases and optimization strategies that speed up the auto-tuning process.</p> <p>3 days ago ★ Featured</p>	<p>GGIR GG</p> <p>Converts raw data from wearables into insightful reports for researchers investigating human daily physical activity and sleep.</p> <p>12 days ago ★ Featured</p>
<p>Xenon Xe</p> <p>If you are using remote machines to do your computations, and don't feel like learning and implementing many different APIs, Xenon is the tool for you.</p> <p>13 days ago ★ Featured</p>	<p>mcfly mc</p> <p>Helps you find a suitable neural network configuration for deep learning on time series.</p> <p>28 days ago ★ Featured</p>
<p>Netherlands eScience Center Python Template Ne</p> <p>Generic template for Python packages, so you can spend less time setting up and configuring, and comply with the Netherlands eScience Center Software</p>	<p>Noodles No</p> <p>Task-based parallel programming model in Python that offers the same intuitive interface when running complex workflows on your laptop or on large computer clusters.</p>

Kernel Tuner

28 mentions
11 contributors

Kernel Tuner greatly simplifies the development of highly-optimized and auto-tuned CUDA, OpenCL, and C code, supporting many advanced use-cases and optimization strategies that speed up the auto-tuning process.



Get started

Cite this software

DOI: 10.5281/zenodo.5500166 [Copy to clipboard](#)

Choose a version: 0.4.1

Choose a reference manager file format: BibTeX [Download file](#)

What Kernel Tuner can do for you

- Allows developers to easily unit test and auto-tune GPU code
- Generic auto-tuning of user-defined parameters for CUDA, OpenCL, and C kernels
- Supports more than 20 different search optimization methods to speedup tuning
- Successfully used in 10+ different eScience projects, across various disciplines

Kernel Tuner simplifies the development of efficient GPU programs, or *kernels*. It does so by making kernels written in C/C++, OpenCL, or CUDA accessible from Python, while taking care of the required synchronization between data kept in host memory and data kept in device memory.

This has a number of advantages. First, it simplifies *auto-tuning* of the kernel parameters. In fact, Kernel Tuner comes standard with a variety of strategies for efficiently searching the parameter space, leading to greatly improved performance of tuned kernels. Second, it allows for unit testing of GPU code from within Python.

Kernel Tuner does not add any additional dependencies to the kernel code, and does not require extensive code changes. Furthermore, it is noteworthy that kernels tuned by Kernel Tuner do not require any changes after tuning to make them production ready—tuned kernels can be used as-is from any host programming language.

[Read less](#)

Tags

GPU High performance computing

Multi-scale & multi model simulations

Real time data analysis

Optimized data handling Big data

Programming Language

Python
CUDA
OpenCL

License
Apache-2.0

Source code

KB LAB Datasets Tools News & events Blog About us LIGHT DARK NL


Join us and explore the KB's digital treasure trove

The KB Lab hosts all experimental tools and data sets based on the KB's digitised collection.

5.482 lines of code 56.330 MB files 60 events

Datasets

more datasets

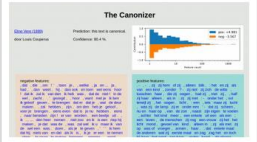


Dutch Novels 1800-2000

Dataset that contains a corpus of 1346 novels from DBNL.


Tools

more tools



Canonizer

The Canonizer demonstration shows how well canonicity can be classified based on a novel.



Timelapse

View a hundred years' worth of newspapers within a minute: the KB visualized its digitized newspapers.

<https://lab.kb.nl>


Help Sponsors Log in Register

Find, install and publish Python packages with the Python Package Index

Search projects

Or [browse projects](#)

368,796 projects 3,361,250 releases 5,855,870 files 583,735 users



The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#).

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#).

<https://pypi.org>

ELIXIR bio.tools Explore Login Sign-up

This site uses cookies. By continuing to browse the site you are agreeing to our use of cookies. Find out more here.

bio.tools

Essential scientific and technical information about software tools, databases and services for bioinformatics and the life sciences.

Search bio.tools

Communities in bio.tools

Explore **communities** in bio.tools and see what resources are used by different scientific communities.

Experts from scientific communities are key in identifying what are the methods and software resources used in their research activities. We rely upon scientific communities to improve the terminology and description of resources in different domains of the Life Sciences. This includes the communities being fostered by ELIXIR, and any other group of scientists with an interest to help out.

We strongly encourage everyone to take part in the community effort and contribute. [Email us](#) for more info.

Communities and domains

In bio.tools we try, with the help of the community, to create content domains or "slices" relevant to a specific community, research area or task (e.g. [Proteomics](#), [Rare diseases](#), [COVID-19](#)).

Explore and search bio.tools [domains](#) and [manage](#) your own domains. Read more about bio.tools domains [here](#).

<https://bio.tools>

README.md awesome

Awesome Research Software Registries

A list of research software registries (also known as catalog, index, warehouse, repository, hub, platform, and other terms).

We define a registry as any publicly accessible website where research software can be registered, and that employs Search Engine Optimization such as the addition of metadata to help promote discovery of the research software that is published on the registry. Websites do not necessarily have to retain their own copies of the research software itself in order to be included in the list below.

Contents

- By country
- By organization
- By programming language
- By domain
- Generic

By country

- France - HAL is an open archive where authors can deposit scholarly documents from all academic fields.
- Netherlands - NARCIS aims to record all scholarly outputs (papers, data, as well as software) produced at Dutch institutions.

By organization

- caltech.edu - California Institute of Technology overview of software packages.
- combiomed.eu - Software catalogue from H2020 project CompBioMed.

Contributors 7

<https://github.com/NLeSC/awesome-research-software-registries>

Awesome Research Software Registries

- Collection of research software registries organized by
 - Country
 - Organization
 - Programming language
 - Scientific domain
- Collective effort
- Please contribute!

<https://github.com/NLeSC/awesome-research-software-registries>



<https://doi.org/10.5281/zenodo.7193991>

Awesome Research Software Registries

A list of research software registries (also known as catalog, index, warehouse, repository, hub, platform, and other terms).

We define a registry as any publicly accessible website where research software can be registered, and that employs Search Engine Optimization such as the addition of metadata to help promote discovery of the research software that is published on the registry. Websites do not necessarily have to retain their own copies of the research software itself in order to be included in the list below.

Contents

- [By country](#)
- [By organization](#)
- [By programming language](#)
- [By domain](#)
- [Generic](#)

By country

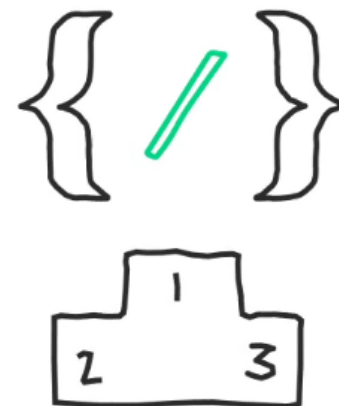
- [France](#) - HAL is an open archive where authors can deposit scholarly documents from all academic fields.
- [Netherlands](#) - NARCIS aims to record all scholarly outputs (papers, data, as well as software) produced at Dutch institutions.

By organization

- [caltech.edu](#) - California Institute of Technology overview of software packages.
- [compbiomed.eu](#) - Software catalogue from H2020 project CompBioMed.
- [darpa.mil](#) - US Defense Advanced Research Projects Agency (DARPA).
- [b2find.eudat.eu](#) - Data and software discovery service based on metadata harvested from research data collections from EUDAT data centres and other repositories.
- [kb.nl](#) - Dutch Royal Library.
- [lbl.gov](#) - Lawrence Berkeley National Lab.
 - [Applied Mathematics Software](#)
 - [Computational Science Software](#)
 - [Computer Science Software](#)
 - [Data Science & Technology Software](#)
- [nasa.gov](#) - NASA software catalog.
- [ontosoft.org](#) - Software from a variety of disciplines, annotated and searchable using the [OntoSoft vocabulary](#) for describing software.



#4 ENABLE CITATION OF THE SOFTWARE



WHY THIS IS IMPORTANT

HELP ME CHOOSE

<https://fair-software.nl>



<https://blog.esciencecenter.nl/cffinit-now-is-the-time-to-get-more-recognition-for-your-software-e2e6ef617f8e>

<https://doi.org/10.5281/zenodo.7193991>



Citation File Format (CFF) [About](#) [Events](#) [Documentation](#) [Tutorials](#) [Create CFF file](#)

4 minute read

What is a CITATION.cff file?

CITATION.cff files are plain text files with human- and machine-readable citation information for software (and datasets). Code developers can include them in their repositories to let others know how to correctly cite their software.

This is an example of a simple CITATION.cff file:

```

cff-version: 1.2.0
message: "If you use this software, please cite it as below."
authors:
  - family-names: Druskat
    given-names: Stephan
    orcid: https://orcid.org/0000-0003-4925-7248
title: "My Research Software"
version: 2.0.4
doi: 10.5281/zenodo.1234
date-released: 2021-08-11
  
```

The format of CITATION.cff files is the Citation File Format (CFF).

Why you should add a CITATION.cff file to your repository!

It is very easy to *correctly* cite a paper: all the necessary information (*metadata*) can be found on the title page or the article website. Software and datasets have no title page, the relevant information is often less obvious.

People who want to cite your software may ask questions like:

citation-file-format.github.io

attribution format citation cred
research-software-engineering citation
software-sustainability wssipe

Readme
CC-BY-4.0 License
Code of conduct
Cite this repository

Rename schema_poc to validator 3 months ago
Make error message more useful 3 months ago
Bump to version 1.2.0 3 months ago
Ignore VSCodium metadata 3 months ago
refs #243; local links are not checked reliably, skipping helps avoid... 4 months ago
Update Zenodo metadata with related identifiers 3 months ago
Add missing version header for 1.2.0 3 months ago
Fix #318: Duplicate identifiers[1] in doi field 2 months ago

CT.md Move CoC to root
Inform users which branch to fork from for which case
Fix #87: Change license file format
Add Bibliography.jl to tooling table
added validator for snippets in schea-guide.md and added it to the
added bumpversion requirement
Add start to the field list of reference
Remove unneeded hash from schema 3 months ago

Cite this repository

If you use this software in your work, please cite it using the following metadata. [Learn more](#)

APA BibTeX

Druskat, S., Spaaks, J. H., Chue Hong, N., H

View citation file



<https://citation-file-format.github.io/>


<https://doi.org/10.5281/zenodo.7193991>

CFF INIT

Generate your citation metadata files with ease

CITATION . c f f files are plain text files with human- and machine-readable citation information for software and datasets.

Code developers can include such files in their source code repositories to let others know how to correctly cite their software.

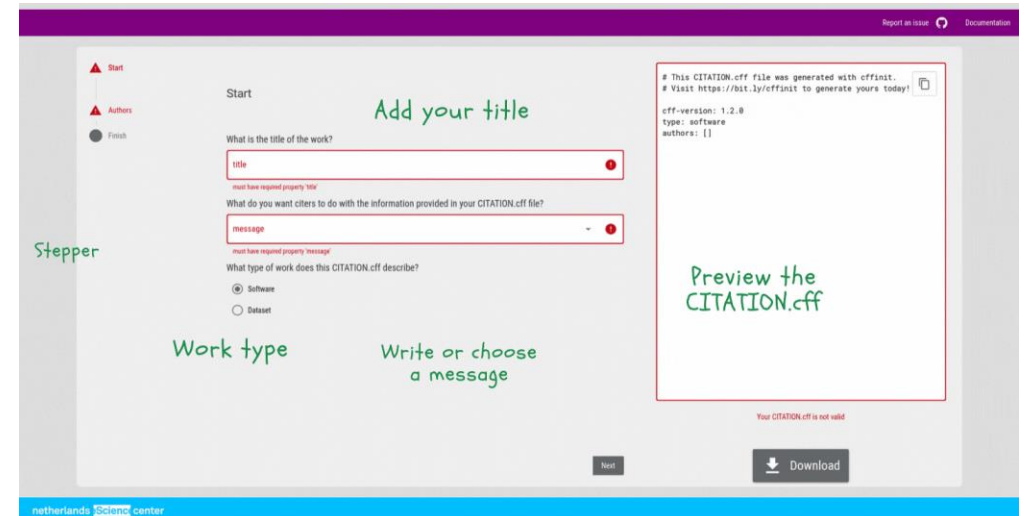


Generate your Citation File Format file

+ Create

<http://bit.ly/cffinit>

<https://citation-file-format.github.io/cff-initializer-javascript>



Start

What is the title of the work?

title

What do you want citers to do with the information provided in your CITATION.cff file?

message

What type of work does this CITATION.cff describe?

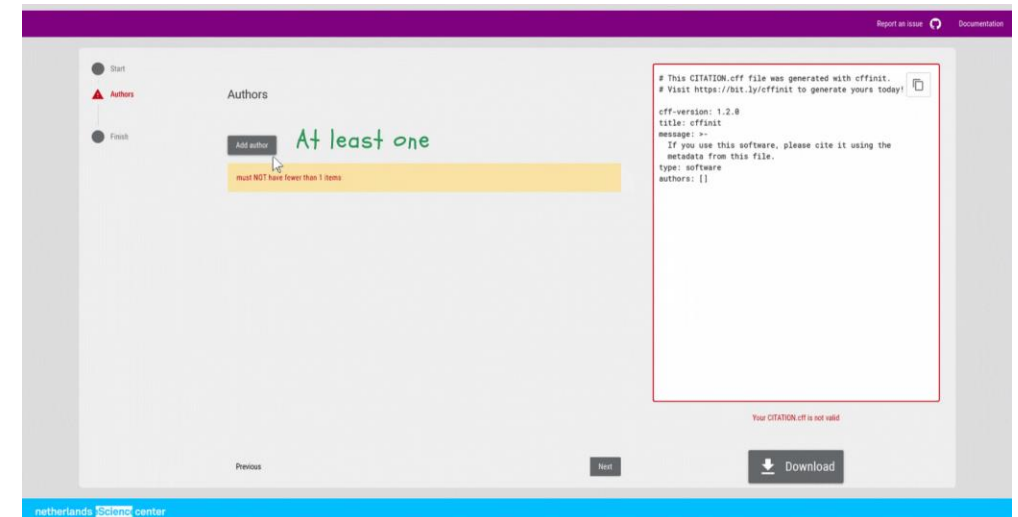
Software

Dataset

Preview the CITATION.cff

Your CITATION.cff is not valid

Download



Authors

Add author

At least one

must NOT have fewer than 1 items

Preview the CITATION.cff

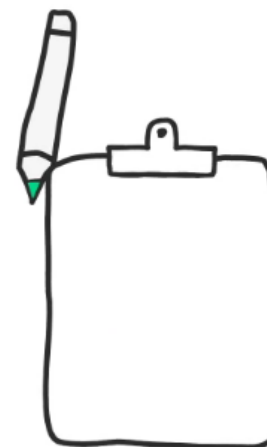
Your CITATION.cff is not valid

Download

<https://doi.org/10.5281/zenodo.7193991>



#5 USE A SOFTWARE QUALITY CHECKLIST



WHY THIS IS IMPORTANT

HELP ME CHOOSE

<https://fair-software.nl>

OpenSSF Best Practices Badge Program

Get Your Badge Now!

The [Open Source Security Foundation \(OpenSSF\)](#) Best Practices badge is a way for Free/Libre and Open Source Software (FLOSS) projects to show that they follow best practices. Projects can voluntarily self-certify, at no cost, by using this web application to explain how they follow each best practice. The OpenSSF Best Practices Badge is inspired by the many badges available to projects on GitHub. Consumers of the badge can quickly assess which FLOSS projects are following best practices and as a result are more likely to produce higher-quality secure software.

You can easily see the [criteria for the passing badge](#). More information on the OpenSSF Best Practices Badging program is [available on GitHub](#). [Project statistics](#) and [criteria statistics](#) are available. The [projects page](#) shows participating projects and supports queries (e.g., you can see [projects that have a passing badge](#)). You can also see [an example \(where we try to earn our own badge\)](#). This project was formerly known as the Core Infrastructure Initiative (CII) Best Practices badge, and was originally developed under the CII. It is now part of the [OpenSSF Best Practices Working Group \(WG\)](#). The OpenSSF is a foundation of the [Linux Foundation \(LF\)](#).

Privacy and legal issues: Please see our [privacy policy](#), [about cookies](#), and [terms of use](#). The code for the badging application itself is released under the [MIT license](#) (projects pursuing a badge are under their respective licenses). All publicly-available non-code content managed by the badging application is released under at least the [Creative Commons Attribution License version 3.0 \(CC-BY-3.0\)](#); newer non-code content is released under CC-BY version 3.0 or later (CC-BY-3.0+). If referencing collectively or not otherwise noted, please credit the OpenSSF Best Practices badge contributors.

Please share this:



Some badge earners:





The Turing Way

Search this book...

- Welcome
- Guide for Reproducible Research
- Guide for Project Design
- Guide for Communication
- Guide for Collaboration
- Guide for Ethical Research
- Community Handbook
- Afterword

Visit our [GitHub Repository](#)
This book is powered by [Jupyter Book](#)



- Contents
- Our Community
- History
- Citing *The Turing Way*

Welcome

Welcome to *The Turing Way* handbook to reproducible, ethical and collaborative data science.

The Turing Way project is open source, open collaboration, and community-driven. We involve and support a diverse community of contributors to make data science accessible, comprehensible and effective for everyone. Our goal is to provide all the information that researchers and data scientists in academia, industry and the public sector need to ensure that the projects they work on are easy to reproduce and reuse.

Top Tip

The Turing Way is not meant to be read from start to finish. Start with a concept, tool or method that you need now, in your current work. Browse the different guides that make up the book, or use the search box to search for whatever you would like to learn about first.

All stakeholders, including researchers, software engineers, project leaders and funding teams, are encouraged to use *The Turing Way* to understand their roles and responsibility of reproducibility in data science. You can inspect our resources on [GitHub](#), contribute to the project as described in our [contribution guidelines](#) and re-use all materials ([see the License](#)).



TESTING.md	Replace Docker images from nlesc/xenon- * to ...	3 years ago
build.gradle	Merge pull request #688 from xenon-middlew...	2 months ago
gradlew	Update of gradle to 4.1. Removed println	5 years ago
gradlew.bat	Update of gradle to 4.1. Removed println	5 years ago
jitpack.yml	Require openjdk11	3 years ago
runAllChecks.sh	Fixed javadoc devel errors	4 years ago
sonar-project.properties	Update sonarcloud config	4 years ago

☰ README.md ✎

Xenon

build passing build failing codecov 64% quality gate passed license Apache-2.0

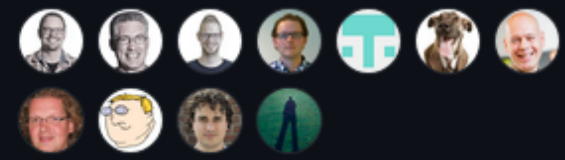
DOI 10.5281/zenodo.597993 rsd xenon openssf best practices passing fair-software.eu ●●●●●

Copyright 2013-2021 The Netherlands eScience Center

What problem does Xenon solve?

Many applications use remote storage and compute resources. To do so, they need to include code to interact with the scheduling systems and file transfer protocols used on

Contributors 11



Environments 1

🚀 github-pages Active

Languages



ARDC FAIR-software checklist

ARDC FAIR self-assessment checklists

Choose [software](#) or [data](#).

ARDC FAIR for software self-assessment checklist

Answer the 18 questions below to assess your software's FAIRness.

Findable

1. Does the software have any identifier assigned?

- (+0) No identifier
- (+1) Local identifier or reasonably unique name
- (+2) Web address (URL)
- (+3) Globally unique and persistent identifier (e.g. DOI, PURL, or Handle)

Compliance with 5 recommendations?



howfairis 0.14.1

✓ Latest version
Released: Mar 9, 2021

Python package to analyze compliance with fair-software.eu recommendations

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Statistics

GitHub statistics:

- ★ Stars: 27
- 🔗 Forks: 7
- 📄 Open issues/PRs: 69

View statistics for this project via [Libraries.io](#), or by using our [public dataset on Google BigQuery](#)

Meta

License: Apache Software License (Apache Software License 2.0)

Author: <https://github.com/jspaaks>

📄 howfairis

Maintainers

- howfairis

Project description

Python package to analyze a GitHub or GitLab repository's compliance with the [fair-software.eu](#) recommendations.

Badges

fair-software.nl recommendations	
(1/5) code repository	
(2/5) license	
(3/5) community registry	
(4/5) citation	
(5/5) checklist	
overall	

Other best practices

Documentation	
Supported Python versions	
Code quality	
Code coverage of unit tests	
DockerHub	
GitHub Actions	
Citation metadata consistency	
Unit tests	
Live tests (triggered manually)	



Badges

fair-software.nl recommendations	
(1/5) code repository	
(2/5) license	
(3/5) community registry	
(4/5) citation	
(5/5) checklist	
overall	

Install

```
pip3 install --user howfairis
```

Verify that the install directory is on the `PATH` environment variable. If so, you should be able to call the executable, like so:

```
howfairis https://github.com/<owner>/<repo>
```

`howfairis` supports URLs from the following code repository platforms:

1. `https://github.com`
2. `https://gitlab.com` (not including self-hosted instances)

<https://pypi.org/project/howfairis>

<https://github.com/fair-software/howfairis>

<https://doi.org/10.5281/zenodo.7193991>

Assess compliance with fair-software.eu

To enable this checker, add the following snippet as `.github/workflows/fair-software.yml` in your GitHub repository.

```
name: fair-software

on: push

jobs:
  verify:
    name: "fair-software"
    runs-on: ubuntu-latest
    steps:
      - uses: fair-software/howfairis-github-action@0.2.1
        name: Measure compliance with fair-software.eu recommendations
        env:
          PYCHARM_HOSTED: "Trick colorama into displaying colored output"
        with:
          MY_REPO_URL: "https://github.com/${{ github.repository }}"
```

<https://github.com/fair-software/howfairis-github-action>

<https://doi.org/10.5281/zenodo.7193991>

Compliance of multiple repositories

fairtally

Python application to analyze multiple GitHub and GitLab repositories compliance recommendations.

fair-software.nl recommendations	Badges
1. Code repository	
2. License	
3. Community Registry	
4. Enable Citation	
5. Checklist	
Other best practices	
Continuous integration	
DockerHub	

<https://pypi.org/project/fairtally/>
<https://github.com/fair-software/fairtally>

- Checks compliance of multiple repositories using howfairis
- Generates a report that includes compliance of each repository
- Can be automated to keep track of the changes

Fairtally results							Search url	
url	repository	license	registry	citation	checklist	count	badge	
https://github.com/fair-software/fairtally	✓	✓	✓	✓	✗	4		
https://github.com/fair-software/howfairis	✓	✓	✓	✓	✓	5		

Records per page: All 1-2 of 2

<https://doi.org/10.5281/zenodo.7193991>

Questions?

netherlands
eScience center