Archive, identification, description and citation of source code for research

Morane Gruenpeter
Outline

- Introduction
- Preserving source code
- Software Heritage the universal source code archive
- Research Software: a first class research output
- Good practices for software curation
- Conclusion
Software engineer and metadata specialist

Timeline:

- **2008-2011** B.A in **Musique** (Harpist)
- **2012-2015** Licence (B.SC) in **Computer Science** @CNAM
- **2015-2017** Master (M.SC) in **Software Engineering** (R&D) @UPMC
- **2017** Internship **Software Heritage** (SWH)
- **2018-2019** European project EU2020 **CROSSMINER** (on SWH team)
- **2020-2022** European project **FAIRsFAIR** (on SWH team)

Working groups for Open Science and digital preservation

- the Research Data Alliance’s **Software Source Code** Interest Group (SSC IG),
- the FORCE11’s **Software Citation** Implementation Working Group (SCI WG),
- the joint RDA, ReSA & FORCE11 **FAIR for Research Software** Working Group (FAIR4RS WG)
- WikiData for **Digital Preservation** initiative (WikiDigi).
What is software?

“Software, instructions that tell a computer what to do. Software comprises the entire set of programs, procedures, and routines associated with the operation of a computer system. The term was coined to differentiate these instructions from hardware—i.e., the physical components of a computer system.”

Software digital artifacts
- created and used to operate on hardware

Software as a concept
- project or entity
- the community around the project
- the software idea / algorithms / solutions
What about **software source code**?
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}

Hello World
The Knowledge is in the Source Code

“Programs must be written for people to read, and only incidentally for machines to execute.”
Harold Abelson, 1985
Structure and Interpretation of Computer Programs (1st ed.),

“Source code provides a view into the mind of the designer.”
Len Shustek, 2006
Computer History Museum

“The source code for a work means the preferred form of the work for making modifications to it.”
GPL Licence
Why source code is fragile?
Source code can be destroyed

Google Kills Off Google Code

Natasha Lomas @riptari / 10:58 AM GMT+1 • March 13, 2015
Source: TechCrunch

1.4 million projects

When we started the Google Code project hosting service in 2006, the world of project hosting was limited. We were worried about reliability and stagnation, so we took action by giving the open source community another option to choose from. Since then, we’ve seen a wide variety of better project hosting services such as GitHub and Bitbucket bloom. Many projects moved away from Google Code to those other systems. To meet developers where they are, we ourselves migrated nearly a thousand of our own open source projects from Google Code to GitHub.

As developers migrated away from Google Code, a growing share of the remaining projects were spam or abuse. Lately, the administrative load has consisted almost exclusively of abuse management. After profiling non-abusive activity on Google Code, it has become clear to us that the service is now necessary no more.

Beginning today, we have disabled new project creation on Google Code. We will be shutting down the service about 10 months from now on January 25th, 2016. Below, we provide links to migration tools designed to help you move your projects off of Google Code. We will also make ourselves available over the next three months to those projects that need help migrating from Google Code to other hosts.

- March 12, 2015 - New project creation disabled.
- August 24, 2015 - The site goes read-only. You can still checkout/view project source, issues, and wikis.
- January 25, 2016 - The project hosting service is closed. You will be able to download a tarball of project source, issues, and wikis. These tarballs will be available throughout the rest of 2016.

Google will continue to provide Git and Gerrit hosting for certain projects like Android and Chrome. We will also continue maintaining our mirrors of projects like Eclipse, kernel.org and others.
In science, reproducibility requires long-term access to source code.

Source: BitBucket blog

Sunsetting Mercurial support in Bitbucket

April 21, 2020 | 3 min read

Denise Chan

[Update Aug 26, 2020] All hg repos have now been disabled and cannot be accessed.

[Update July 1, 2020] Today, mercurial repositories, snippets, and wikis will turn to read-only mode. After July 8th, 2020 they will no longer be accessible.

Source: BitBucket blog

250,000 repos
GitHub now belongs to Microsoft

Microsoft to acquire GitHub for $7.5 billion
June 4, 2018 | Microsoft News Center

Acquisition will empower developers, accelerate GitHub's growth and advance Microsoft services with new audiences

Source: Microsoft Blog

GitLab gains developers after Microsoft buys rival GitHub

By Vibhuti Sharma, Supantha Mukherjee

Source: Reuters
Hosting your open-source project

- on a free, publicly available platform is fine.
- But you have to prepare for the platform shutdown (you need a plan B).
Version control system (VCS) history

- records changes made to a (set of) source code file(s)
- allows to operate on versions: diff/merge/fork/recover etc.
- essential tool for software development
The development history is key

Software evolves over time

- projects may last decades
- the development history is key to its understanding

Complexity

- millions of lines of code
- large web of dependencies
- easy to break, difficult to maintain
- sophisticated developer communities
Outline

★ Introduction
★ Preserving source code
★ Software Heritage the universal source code archive
★ Research Software: a first class research output
★ Good practices for software curation
★ Conclusion
Collect, preserve and share all software source code

Preserving our heritage, enabling better software and better science for all

- Non-profit organization
- Launched in 2016 by INRIA (Roberto Di Cosmo & Stefano Zacchiroli)
- Archives publicly available code permanently and for free.
Software Heritage in a nutshell

**Reference catalog**
find and reference all software source code

**Universal archive**
preserve all software source code

**Research infrastructure**
enable analysis of all software source code
In numbers

Source files: 10545239307
Commits: 2215028000
Projects: 161168065

Directories: 8776445689
Authors: 43822776
Releases: 17972096
Rescuing software

Google Code content now safely collected

Rescuing 250000+ endangered Mercurial repositories
An international, non profit initiative

Platinum sponsors
- CNRS
- Intel
- Microsoft

Gold sponsors
- openInventionNetwork
- Université de Paris
- Sorbonne Université

Silver sponsors
- Cast
- GitHub
- Google
- Università di Pisa
- VMware

Bronze sponsors
- DANS
- FOSSID
- Other
How to archive the world's source code?
Archiving software
Crawling

The SWH archive **harvests actively** source code from different sources and converts all the source code into a single and universal data structure which is an enormous Merkle DAG [Merkle, 1987].

Crawling is separated into three phases:

1. **listing software sources**,  
2. **scheduling updates** and  
3. **loading the software artifacts into the archive**.
Data model

The data model adopted by Software Heritage to represent the information that it collects is centered around the notion of *software artifact*, using the following canonical names, from bottom to top:

- contents,
- directories,
- revisions and
- releases.

Using also *origins, visits* and *snapshots* to store provenance information. Read more in Software Heritage: Why and How to Preserve Software Source Code.

Merkle tree

Combination of
- tree
- hash function

Classical *cryptographic construction*
- fast, parallel signature of large data structures
- widely used (e.g., Git, blockchains, IPFS, . . . )
- built-in deduplication
Intrinsic identifier

SWH provides a Persistent IDentifier (PID) that can identify each and every source code artifact with integrity, called a SWHID.

SWHIDs are intrinsic identifiers which are intimately bound to the designated object, they do not need a register, only agreement on a standard.

Intrinsic vs. extrinsic blog post

Go to API endpoint
The complete workflow

Full development history permanently archived in a **uniform data model**
Revisions

Details
- SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6
- Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep  1 14:26:13 2016)
- Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep  1 14:26:13 2016)
- Subject: provenance.tasks: add the revision -> origin cache task
- Parent: fc3a8b59ca1df424d860f2c29ab07fee4dc35d10: test_storage: properly pipeline origin and cont...
  provenance.tasks: add the revision -> origin cache task

```
swh/storage/provenance/tasks.py
```

```
tree 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fee4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

provenance.tasks: add the revision -> origin cache task
```

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6
Releases

id: 85083a5cc441a89dea73f5bd67c3f9c6afdb
Snapshots

commit 08ffe2b25770109525eb3ce21691466c53a1d9158 refs/heads/ati
time
commits ba5443a24e3f9e3e32a46c2923ec47cbe61ce673eb refs/heads/directory-listing-arrays
commits d69e0dbaf92383f165806b21fbeb1c85d72738db9c5 refs/heads/food
commits cff7f4e9ee0eb272f0946008f5a0819f67de468e08 refs/heads/master
commits 5eca19f7c662d824047e5a4b9e9b8a4436a136f2c2 refs/heads/master-add
commits 642a265f37de5805a854d27b53ee4efa2b225e82e refs/heads/tmp/generic-releases
tag 2df03b1379c7f68d6659799f9fd4980c75771755 refs/tags/v0.0.1
tag 72a1991a38643e359996db86876f00bee72ee2c2d refs/tags/v0.0.10
tag 3590e8ca86eb70e05b376795f2a230bbfa4fafa5c refs/tags/v0.0.11
tag 33378427a493b369a6777b7d858f6f674f6c6556 refs/tags/v0.0.12
tag 067f46527553b27c7f590311c2ba036c3f3b35d refs/tags/v0.0.13
tag 5a6252fe88ab8585d7442667d92a1e3213bd refs/tags/v0.0.14
tag 586feb4de58084f5fa05f399367643cbcb1a9c7f refs/tags/v0.0.15
tag 8c0bb80f4909bf363177742b2089f666e0be51c refs/tags/v0.0.16
tag a5422444ee3f0f6ed35e8bf292f83e835c899abc7d6 refs/tags/v0.0.17
tag 228a2f1658dd12222e556559462e1e06fc4993d9 refs/tags/v0.0.18
tag 6069779a4ca05d497fc0d24aad00dce8263e6f47c refs/tags/v0.0.19
tag 32b5a591fc2a323bbad651f5e6ad5382e275a67 refs/tags/v0.0.2
tag 3147c3d3ec46cf642f881e908b1237edbf22c7 refs/tags/v0.0.20
tag 215e5a05aba131e082e0b72e76e6bf4b6073a87908 refs/tags/v0.0.21
tag 3f1b168c2672a5d6252124257a1e5d50fe5f5f0e7e7e7 refs/tags/v0.0.22
tag 8c0eb88a4d73f5d5262789e460a16ac372aba4 refs/tags/v0.0.23

id: b464cad1b66ff266a37b46ea6e7a04b545e904b
The complete workflow

Full development history permanently archived in a uniform data model
Outline

Introduction

Preserving source code

Software Heritage the universal source code archive

Research Software: a first class research output

Good practices for software curation

Conclusion
Multiple facets, it can be seen as:
- a tool
- a research **outcome** or result
- the **object** of research
Much more complex than it seems

- **Structure**
  - monolithic/
  - composite;
  - self-contained/
  - external dependencies

- **Lifetime**
  - One-shot / long term

- **Community**
  - one person / one team/distributed community

- **Authorship**
  - complex set of roles

- **Authority**
  - institutions/organizations/communities/single person

Various granularities

- **Exact status** of the source code for **reproducibility**, e.g.
  - “you can find at
    swh:1:cnt:cdf19c4487c43c76f3612557d4dc61f9131790a4;lines=146-187 the core algorithm used in this article”

- **(Major) release** “This functionality is available in **OCaml version 4**”

- **Project** “Inria has created OCaml and Scikit-Learn”
A plurality of needs

Researchers

• **archive and reference** software used and created in articles
• **find** useful software
• **get credit** for developed software
• **verify/reproduce/improve** results

Research Organization

know its **software assets** for:

• technology transfer,
• impact metrics,
• strategy

Laboratories/teams

• **track** software contributions
• **produce** reports
• **maintain** web page
Important policy tool in Open Science

- 9 infrastructures
  - 3 archives
  - 3 open access publishers
  - 3 aggregators
- recommendations
  - archive in Software Heritage,
  - use SWHID
  - open non profit
  - default to open source for research software

"all research software should be made available under an Open Source license by default, and all deviations from this default practice should be properly motivated"

See [https://doi.org/10.2777/28598](https://doi.org/10.2777/28598)
L’ouverture des codes sources des logiciels est un enjeu majeur de reproductibilité des résultats scientifiques.
What is at stake? In order of difficulty

**Archive**
- Research software artifacts must be properly archived
- make it sure we can retrieve them (reproducibility)

**Reference**
- Research software artifacts must be properly referenced
- make it sure we can identify them (reproducibility)

**Describe**
- Research software artifacts must be properly described
- make it easy to discover them (visibility)

**Cite (for credit)**
- Research software artifacts must be properly cited (not the same as referenced!)
- to give credit to authors (evaluation!)
Submit code to HAL

- A scholarly repository
- An archive
- Software is more findable
- Transferring code to SWH

Save code now to SWH

- Easy (only submit repo’s url)
- All dev history archived
- different vcs are accepted
- PID to reference specific pieces of code (even algorithms)

Classic deposit
.zip, .tar.gz

SWHID deposit

Save code now
Git, svn, hg
HAL and SWH collaboration

Key dates

★ 2017 - Collaboration launch
★ Mars 2018 - Beta-test on HAL-Inria
★ Septembre 2018 - Launch on all HAL instances
★ Avril 2020 - BibLaTeX @software export

Coming soon:

★ end 2021 - Deposit with SWHID
Advantages:

★ Metadata moderation by the digital archivist team
★ Export formats available on the software record - to cite software

Deposit guide:
Save your any code now!

https://save.softwareheritage.org/

You can contribute to extend the content of the Software Heritage archive by submitting an origin save request. To do so, fill the required info in the form below:

- **Origin type**: git
- **Origin url**: [input field]

Submit

A "Save code now" request takes the following parameters:
SWHID deposit

Advantages:

★ Metadata moderation by the digital archivist team
★ Export formats available on the software record - to cite software
★ Retrieve metadata with codemeta.json to complete form
Reference vs. citation

Credit & Attribution
a metadata record
all authors & contributors

Reuse & Reproducibility
a specific artifact
with complementary information (docs)

Archive & Index
metadata record (HAL)
artifact itself (SWH)
**Software concept / project / collection**
Description in registry, a homepage or any other form of metadata record
- Project versions (for example Python2 and Python3)
- Modules
- Sub-modules

**Software artifact**
- Executable (download link)
- Software source code
  - Dynamic artifact - current development code
  - Archived copy
    - Snapshot (all branches, all dev history)
    - Release / Package
    - Commit- a specific point in development history
- Directory
- File
- Algorithm

---

Research Data Alliance/FORCE11 Software Source Code Identification WG et al. (2020). Use cases and identifier schemes for persistent software source code identification (V1.1). Research Data Alliance. [https://doi.org/10.15497/RDA00053](https://doi.org/10.15497/RDA00053)
Reference source code in SWH

Choose the granularity level for the reference:

code fragment
- swh:1:cnt:c60366bc03936eede6509b23307321faf1035e23;lines=473-537
- James McCaffrey’s algorithm in sageMath

specific version - release
- swh:1:rel:22ece559cc7cc2364edc5e5593d63ae8bd229f9f
- release 2.3.0 of Darktable, dated 24 December 2016

full repository - snapshot
- swh:1:snp:c7c108084bc0bf3d81436bf980b46e98bd338453
- a snapshot of the entire Darktable repository (4 May 2017, GitHub)
Citation export in HAL

→ Citation accessible on the HAL record
→ Export BibTeX using the format BibLaTeX @software or @softwareversion (if a version property was submitted)
→ Export used in activity reports for scientific outputs at Inria since 2020.
Citation with the biblatex-software package

BibLaTex style extension for software


Describe - mandatory files for a HAL deposit

Prepare your code with the following files:

These files are verified by moderators

- README ([https://readme.so/fr/editor](https://readme.so/fr/editor))
- AUTHORS (containing list of authors)
- LICENSE
  - Open-source SPDX compliant license
    [https://choosealicense.com/](https://choosealicense.com/)
    [https://reuse.software/](https://reuse.software/)
- codemeta.json (not mandatory but useful)

See also: [HOWTO archive and reference your code](#)
Describe: What’s a good README

★ **MUST** include:
  ○ Name and a description of the software.

★ **SHOULD** include:
  ○ how to run and use the source code
  ○ build environment, installation, requirements

★ **CAN** include:
  ○ project website or documentation pointer and recent news
  ○ visuals

extracted from Eric Steven Raymond and Make a README
Describe - choose a vocabulary?

Software schemes

General schemes

Software ontologies landscape from Pathways for Discovery of Free Software (slide deck from LibrePlanet 2018).

(Gruenpeter & Thornton, 2018) CC-by-4
An initiative
  ○ An academic community discussing software metadata

A vocabulary
  ○ A subset of schema.org

A crosswalk table - mapping the metadata landscape

An open source tool to create codemeta.json files

Contributed to the community by Software Heritage

Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.

The software itself

Name
My Software
the software title

Description
My Software computes ephemerides and orbit propagation. It has been developed from early `80.

Creation date
YYYY-MM-DD

First release date
YYYY-MM-DD

Use it directly on the CodeMeta hosted version

Contributions are welcome on the code repository
Describe: Software metadata terms

<table>
<thead>
<tr>
<th>Identify</th>
<th>administrate</th>
<th>classify</th>
<th>execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>● identifier</td>
<td>● maintainer</td>
<td>● description</td>
<td>● buildInstructions</td>
</tr>
<tr>
<td>● name</td>
<td>● citation</td>
<td>● keywords</td>
<td>● issueTracker</td>
</tr>
<tr>
<td>● author(s)</td>
<td>● funder(s)</td>
<td>● supportingData</td>
<td>● operatingSystem</td>
</tr>
<tr>
<td>● contributor(s)</td>
<td>● license</td>
<td>● referencePublication</td>
<td>● softwareRequirements</td>
</tr>
<tr>
<td>● version</td>
<td>● editor</td>
<td>● algorithms*</td>
<td>● runtimePlatform</td>
</tr>
<tr>
<td>● applicationCategory</td>
<td>● publisher</td>
<td>● readme (docs*)</td>
<td>● downloadUrl</td>
</tr>
<tr>
<td>● codeRepository</td>
<td>● dates</td>
<td></td>
<td>● memory, processor, storage</td>
</tr>
<tr>
<td></td>
<td>○ created,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ modified</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>○ published</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>● developmentStatus</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New actor: CITATION.cff

Advantages

- BibTeX export with type `@software`
- Citation.cff are also indexed in the SWH archive

Source: https://github.com/lanl/PRISM
Outline

★ Introduction
★ Preserving source code
★ Software Heritage: the universal source code archive
★ Research Software: a first class research output
★ Good practices for software curation
★ Conclusion
Wrap up

★ 📦 Archive your code!
https://save.softwareheritage.org/

★ 👓 Describe your code with metadata
README, LICENSE, AUTHORS, codemeta.json

★ 🗺️ Reference your code
SWHID over DOI, context

★ 🎓 Cite your code
Version, release, file, lines
The SWH ambassadors program

Who can be an ambassador?

★ **Enthusiastic individuals** who wish to volunteer as ambassadors to help grow the [Software Heritage community](https://softwareheritage.org)

★ An ambassador can come from **different areas**: academia, cultural heritage, industry and public administration.

Do I need a technical background to become an ambassador?

★ Absolutely not! Source code archival is a cross-domain, cross-expertise concern.

★ you can have a technical background which might be helpful to read technical documentation and trying some of the more advanced features.
Thank you for your attention!

Keep in touch: morane@softwareheritage.org
@moraneottilia, @SWHeritage
https://www.fairsfair.eu/fairsfair-newletters/
https://www.softwareheritage.org/newsletter/