Software Heritage at ZB MED colloquium
Archiving and Referencing all the source code towards recognizing software in academia

Morane Gruenpeter
Software engineer and metadata specialist
Inria, Software Heritage
morane@softwareheritage.org

December 10th, 2020
Introduction

The knowledge is in the source code!

Software Heritage: the universal source code archive

Data model and SWHID: the source code fingerprint

Recognizing software as a research output

The missing piece - the Metadata

Conclusion
Short Bio: Morane Gruenpeter

Goal: Building the Semantic Web of Free and Open Source Software

2008-2011  B.A in Musique (Harpist)
2012-2015  Licence (B.SC) in Computer Science @CNAM
2015-2017  Master in Software Engineering (R&D) @UPMC
            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 & FORCE11 Software Identification Working Group (SCID WG)
Software is all around us

Apollo 11 Guidance Computer (~60,000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."  
Margaret Hamilton

The World Wide Web, 1989, at CERN on a NeXT machine

“When somebody has learned how to program a computer … You’re joining a group of people who can do incredible things. They can make the computer do anything they can imagine.”

From An Insight, An Idea with Tim Berners-Lee (2013)
What is software?

Image taken from from reddit - ProgrammerHumor
Software definition

Encyclopædia Britannica

“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 as a concept
- software project / entity
- the creators and the community around it
- the software idea / algorithms / solutions

Software artifact
- the executable (or binary) of each version for a specific environment
- the software source code for each revision
### Much more complex than it seems

**Software is complex**

<table>
<thead>
<tr>
<th>Structure</th>
<th>monolithic/composite; self-contained/external dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime</td>
<td>one-shot/long term</td>
</tr>
<tr>
<td>Community</td>
<td>one man/one team/distributed community</td>
</tr>
<tr>
<td>Authorship</td>
<td>complex set of roles</td>
</tr>
<tr>
<td>Authority</td>
<td>institutions/organizations/communities/single person</td>
</tr>
</tbody>
</table>

### 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”.

---

*Morane Gruenpeter*

www.softwareheritage.org December 10th, 2020
Is this *software*?

What about *software source code*?
Outline

1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. Data model and SWHID: the source code fingerprint
5. Recognizing software as a research output
6. The missing piece - the Metadata
7. Conclusion
The knowledge is in the source code!

"The source code for a work means the preferred form of the work for making modifications to it."

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

Program (source code)

```
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}
```
Source code is *special*

**Executable and human readable knowledge**

“Programs must be written for people to read, and only incidentally for machines to execute.”
Harold Abelson

**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*
# THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAINED BY ADLER AND EYLES.

```
MONI SQT QUI MAL Y PENSE

----------------------------------------------------
TABLES FOR THE IGNITION ROUTINE
----------------------------------------------------
NOLI SE TANGERE
```

```
P12TABLE VN 0674  # (0)  
CF  UNGNUT  # (1)  
CF  COMFAIL3  # (2)  
CF  GOCUTOFF  # (3)  
CF  TASKOVER  # (4)  
CF  P12SPOT  # (5)  
DEC  D  # (6)  NO ULLAGE  
EBANK= WHICH  
2CDR= SERVEXIT  # (7)  
CF  DISPCHNG  # (11)  
CF .WaitIT  # (12)  
CF  P12IGN  # (13)  
P48TABLE VN 0648  # (0)  
```
Version control system (VCS)

- 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

Three decades of evolution
1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. Data model and SWHID: the source code fingerprint
5. Recognizing software as a research output
6. The missing piece- the Metadata
7. Conclusion
Collect, preserve and share all software source code

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

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
Our principles


Morane Gruenpeter

Software Heritage

As of today the archive already contains and keeps safe for you the following amount of objects:

- Source files: 8,846,381,610
- Commits: 1,880,663,008
- Projects: 140,348,311
- Directories: 7,506,954,410
- Authors: 38,603,337
- Releases: 15,051,940
Growing Support

Raising awareness: landmark agreement, 3/4/2017; grand opening, 7/6/2018

Sharing the vision

Sponsoring our work

Platinum sponsors
- Microsoft
- Intel
- Société Générale

Gold sponsors
- huawei
- Universidad de Paris
- Sorbonne Université

Silver sponsors
- openinventionnetwork
- AdaCore
- gandi.net
- Computer History Museum
- Software Freedom Conservancy
- Eclipse Foundation
- GitLab
- INRIA
- Linux Foundation
- OW2
- SIGSOFT
- Creative Commons
- Open Source Initiative
- Free Software Foundation
- Software Freedom Law Center
- AdaCore
- gandi.net
Under the hood: Automation, and storage

Global development history permanently archived in a uniform data model
- over 6 billion unique source files from over 90 million software projects
- ~400 TB (uncompressed) blobs, ~20 B nodes, ~280 B edges
Much more than an archive!

Merkle tree (R. C. Merkle, Crypto 1979)

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
The archive in pictures
The archive in pictures

Contents

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom
to share and change all versions of a program--to make sure it remains
free software for all its users. We, the Free Software Foundation, use
the GNU General Public License for most of our software; it applies also
to any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to make restrictions that prevent
others from doing what will break these rights. These restrictions.

sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: 94a9ed024d385...
length: 35147
The archive in pictures
The archive in pictures

id: 515f00d44e92c65322aaa9bf3fa097c00dddb9c7d
The archive in pictures

Snapshots → Releases → Revisions → Directories → Contents
Revisions

<table>
<thead>
<tr>
<th>Details</th>
<th>Changes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA: 963634dca6ba5dc37e3e426ba091092c267f9f6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Committer: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject: provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parent: fc3a8b59ca1df424d860f2c29ab07fee4dc35d10</td>
<td>test...storage: properly pipeline origin and cont...</td>
<td>provenance.tasks: add the revision -&gt; origin cache task</td>
</tr>
</tbody>
</table>

```
515f00d44e92c65322aaa9bf3fa097c00dd9c7d
tree
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: 963634dca6ba5dc37e3e426ba091092c267f9f6
```
The archive in pictures
The archive in pictures

 Releases

 tag v0.0.51
  Tagger: Nicolas Danriment <nicolas@danriment.eu> 
  Date: Wed Aug 24 14:36:03 2023 +0200
  Release swh:storage v0.0.51
   - Add new metadata column to origin_visit
   - Update swh-add-directory script for updated API
     ---BEGIN PG SIGNATURE-----
     IQi8BAABCAAdBQJxZTNFhxuavnWVeGQGGRbWt/hyaWl1bnQuZUXUACgicQ7AWLMo2+ 
     neqprwh/aed50B5DaEw+KWNfWgYUc5+ikLvEvh1nWKeYwE6Y77txZK3AE2LED7Uf 
     ahp76p3q8lnp6acC1+Yy98rphc3Lj2yrthZeeWWWqZtwWNNMAEoQ7D0b8qphw8AD58t2 
     ICBl2juXXuCD9y3khPvwwZXh+hbo5MVmy3Dr6+w7Z7K4M+uPQggjyHP755yo 
     IGE/hWm0h7Y1Y3b1n5d7BseXqAq4+-bceqcdub7Z2xj+xj+pUQCBlyqY3nhr/FL 
     qZ2mut8y87Bh7Gh1+Pv1+hscrwbnPq3STHr0qjJ7UpKq92SPP9qUShZvICace 
     klje9qWy8UbMeb+nNkVjeibR93+yWBF3qSop5a1V1WoOTn6Ed1ALCMEmkEk0KMeKt 
     dgr/Maxa11jgDEfnqW67QG6DxKPKPhf9yQLQ3V3GxQR7NTr1tpMc0v066H97bawZC 
     Gq/K1PbHT4x2N646y9Py9ye0U2VXGudsvWQ9WZf4Z2KWiMn4S2Mdc40rdrjlSUQMn 
     RpPTTUt8sXeUxHxOOGpkm9xh5Yvnp1pQc76USTKoG6e4A2Zm1k0gMrwXCVClq9n 
     nhltb9sbMKhMywF8y75p0bY1K73P1fPRUOGWDozK9Wx2W2KZCkgkXy67fj0z9 
     gw1rZQMf9qWQC5O5dmAL2+hvPakYycXMejgh6zehEhivucx 
     =kOxP 
     ---END PG SIGNATURE-----

 id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb
The archive in pictures

Snapshots

commit 0fffeb25770109522eb3ce21691466c53a1d9158 refs/heads/atime
commit ba54a3a24e9f9fe32a3b46c292ccc4cfebe61c67eb refs/heads/directory-listing-arrays
commit dff0eda0f892338f5f69b07fba1c8567238269c5 refs/heads/foo
commit c7ff39eaaeab22b8f469f0f5a58610f672e6e80e88 refs/heads/master
commit 7eca197f7c6e6d2824047e54b1ed9e8b44361a8f2c refs/heads/myp-copy
commit 042a2285f153e5805a650427b533ee4f0225262e2 refs/heads/tmp/generic-releases
tag 291043cb1378cf7689f65597790f6d6907c72f77f55 refs/tags/v0.1

tag 72a21993a384e539996dbb86bf08beee72aee2cd refs/tags/v0.2.0

tag 3599e0caabeb677663767675fa2388b64f8a55c refs/tags/v0.2.1

tag 33378427a433b3a594a7777780b6e667df4c6555 refs/tags/v0.2.2

tag a6f7465273b327cf590311c2ba036cf3b4b35d refs/tags/v0.2.3

tag 5a6325fe686d85465b1e7442867d92a1e32ff3bbd refs/tags/v0.2.4

tag 586f68a45b8b47f5a9b5f9367643cb31a9cf7f refs/tags/v0.2.5

tag 8cd8bb85f488b5363177742bd289f6605be51c refs/tags/v0.2.6

tag a54244aee3f9febee326e287ee635b899abc7d6 refs/tags/v0.2.7

tag 228a7f15590d1222e55353462e1086fc4993d9 refs/tags/v0.2.8

tag 669797a9ca85d497fc0d24aad0d9c82536fe47c refs/tags/v0.2.9

tag 32b7f5a59fca2a3baed651f5a5a6332e275a67 refs/tags/v0.2.10

tag 3147f3d31e4c6f64927801e96b01237e8df2c7 refs/tags/v0.2.11

tag 215ea0ab1118ee2b07e27e6eb4b678a387080 refs/tags/v0.2.11

tag 3fb1e8c2872a5d82b52124257a1a5dfcf85f1a1df refs/tags/v0.2.12

tag 8cdbee8d4d73f54262789e46b16ac3c72aba4 refs/tags/v0.2.13

...
### Our challenges in the PID landscape

#### Typical properties of systems of identifiers
- uniqueness, non ambiguity, persistence, abstraction (opacity)

#### Key needed properties from our use cases
- **gratis** identifiers are free (billions of objects)
- **integrity** the associated object cannot be changed (sw dev, reproducibility)
- **no middle man** no central authority is needed (sw dev, reproducibility)

We could not find systems with both **integrity** and **no middle man**!

Intrinsic, decentralised, cryptographically strong identifiers = SWHIDs
The SWH-ID schema

```
swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
```

- `schema_version`
- `object_id`
- `prefix`
- `object_type`
The SWH-ID schema

```
swh:1:cnt:41dd823118f92d7218099a5e7a990cf58f1d07fa
```

- `prefix`:
  - `swh:1`
  - `cnt`

- `object_type`:
  - "snp" - snapshot
  - "rel" - release
  - "rev" - revision
  - "dir" - directory
  - "cnt" - content
The SWH-ID schema

```
swh:1:cnt:41d0b23118f92d7218099a5e7a99c58f1d07fa
```

- `schema_version`
- `object_id`
- `prefix`
- `object_type`
- `lines_ctx`
- `origin_ctx`

- "snp" - snapshot
- "rel" - release
- "rev" - revision
- "dir" - directory
- "cnt" - content

```
;lines=64-72
;origin=https://github.com/chrislgarry/Apollo-11
```
Let’s look at some famous excerpts of source code

Apollo 11 source code (excerpt)

```
P635POT3 CA BIT6 # IS THE LR ANTENNA IN POSITION 1 YET
EXTEND RAND CHAI33
EXTEND B2F P635POT4 # BRANCH IF ANTENNA ALREADY IN POSITION 1
CAF CODE508 # ASTRONAUT: PLEASE CRANK THE
TC BANKCALL # SILLY THING AROUND
CAND GOPERF1
TCF GOTOPOOH # TERMINATE
TCF P635POT3 # PROCEED SEE IF HE’S LYING
```

Quake III source code (excerpt)

```
float Q_sqrt(float number)
{
    long i;
    float x2, y;
    const float threehalves = 1.5F;
    x2 = number * 0.5F;
    y = number;
    i = * ( long *) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float *) &i;
    y = y * ( threehalves - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalves - ( x2 * y * y ) ); // 2nd iteration, this
    // can be removed
    return y;
}
```

It works!

we have **intrinsic** identifiers for all 20+ billion objects in the archive
1 Introduction
2 The knowledge is in the source code!
3 Software Heritage: the universal source code archive
4 Data model and SWHID: the source code fingerprint
5 Recognizing software as a research output
6 The missing piece - the Metadata
7 Conclusion
Software is a *forgotten* pillar of Open Science

**Lack of recognition**

- not (yet) a first class output
  - in the EOSC plan
  - in the scholarly world

*Sometimes, if you don’t have the software, you don’t have the data*

Christine Borgman, Paris, 2018

---

**Reproducibility is the key**

*non-reproducible single occurrences are of no significance to science*

Karl Popper, *The Logic of Scientific Discovery*, 1934
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

Laboratories/teams
- track software contributions
- produce reports
- maintain web page

Research Organization
- know its software assets for technology transfer, impact metrics and strategy.
Software in research has different roles

Multiple facets, it can be seen as:

- a tool
- a research outcome or result
- the object of research

By identifying the software role, we can decide how to treat it
## What is at stake

<table>
<thead>
<tr>
<th>Archival</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research software artifacts must be properly archived</td>
<td>Research software artifacts must be properly referenced</td>
</tr>
<tr>
<td>make it sure we can retrieve them (<em>reproducibility</em>)</td>
<td>make it sure we can identify them (<em>reproducibility</em>)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research software artifacts must be properly described</td>
<td>Research software artifacts must be properly cited (<em>not the same as referenced!</em>)</td>
</tr>
<tr>
<td>make it easy to discover them (<em>visibility</em>)</td>
<td>to give <em>credit</em> to authors (<em>evaluation!</em>)</td>
</tr>
</tbody>
</table>

The research software (deposit) use case

Deposit software in HAL poster

Generic mechanism:
- SWORD based
- review process
- versioning

How to do it: (guide)
- deposit .zip or .tar.gz file with metadata

Timeline:
- March 2018: test phase on HAL-Inria
- September 2018: open to all HAL
- December 2019:
  - 80 complete source code deposits
  - 98 software records
Submit your source code
The deposit view

HAL-Inria

CGAL 3D Periodic Mesh Generation

Michael Botsch 1, Aymeric Pelle 1, Moïs Rouxel-Labbé 2, Monique Teillaud 1

1 Geometric Computing, Inria Saclay, France; 2 University of Lorraine, France

Abstract: This package provides a 3D mesh generator for periodic domains. The domain to be meshed is a region of the three-dimensional flat torus with cubic fundamental domain.

METADATA

version
CGAL 4.13

Software License
GNU General Public License v3.0 or later

Programming Language
C++

Code Repository
https://github.com/CGAL成果转化/Periodic_Mesh

Platform/OS
Multiplatform

COLLECTIONS

UNIV/LORRAINE | LORIA | LORIA-ALGO | CNRS | INRIA | UNIV/PARIS-SACLAY

CITATION


SHARE

www.softwareheritage.org December 10th, 2020 28 / 46
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)
  connect the dots...

![Metadata Record Diagram]

- **Authors**: Gruenpeter M.
- **Date of production**: 2015
- **Source code identifier (swh-id)**: swh1:dir:fb13b51abbcfd13de65d9ba8d670a23679576cd7
- **Title**: The assignment problem
- **Deposit type**: software
- **hal-id**: hal-01243573
Save and reference research software

Prepare your public repository with:
- README, LICENSE, AUTHORS & codemeta.json files

What’s a good README

extracted from Eric Steven Raymond and Make a 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
Save and reference research software

Save code now on https://archive.softwareheritage.org/save/

- git, svn or mercurial
- intrinsic metadata files
- complete history
Choose the granularity level for the reference:

**file (with code fragment)**

```
swh:1:cnt:c60366bc03936e03936e6509b23307321f0f1035e23;lines=473-537
... and add ;origin=https://github.com/sagemath/sage/
```

James McCaffrey’s **algorithm** in sageMath

**directory**

```
swh:1:dir:c6f07c2173a458d98de45d4c459a8f1916d900f
... and add ;origin=https://github.com/id-Software/Quake-III-Arena/
```

source code of **Quake-III Arena** from id-Software
Save and reference research software

**specific release**

```
swh:1:rel:22ece559cc7cc2364edc5e5593d63ae8bd229f9f
... and add ;origin=https://github.com/darktable-org/darktable/
```

release 2.3.0 of Darktable, dated 24 December 2016

**full snapshot (including all branches and all releases)**

```
swh:1:snp:c7c108084bc0bf3d81436bf980b46e98bd338453
... and add ;origin=https://github.com/darktable-org/darktable/
```

a snapshot of the entire Darktable repository (4 May 2017, GitHub)
1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. Data model and SWHID: the source code fingerprint
5. Recognizing software as a research output
6. The missing piece- the Metadata
7. Conclusion
“Ontologies are agreements, made in a social context, to accomplish some objectives. It’s important to understand those objectives, and be guided by them.”


What do we want to describe?

- a software project?
- a software artifact? a collection of artifacts?
- With what terms or vocabulary?
The Software Ontology *Touchstone*

Software Citation Principles (Smith et al. 2016)

- **Importance**: first class citizen in the scholarly ecosystem
- **Credit and attribution**: authors, maintainer
- **Unique identification**: points to a unique, specific software version (DOI, Git SHA1 hash, etc.)
- **Persistence**: identification beyond the lifespan of the software (swh-id)
- **Accessibility**: url, publisher
- **Specificity**: version, environment
The metadata landscape

Software schemes

General schemes

- Dublin Core
- PRONOM
- Digital Preservation
- PREMIS
The metadata landscape

Software schemes

- DOAP
- ADMS.SW
- Dublin Core
- PRONOM
- Digital Preservation
- PREMIS

General schemes

- Pypi
- NPM
- Maven
- Package Management
The metadata landscape

Software schemes

DOAP
Q7397-software
Q341-free software
Dublin Core
PRONOM

Wikiidata
Digital Preservation
Linked Data

ADMS.SW
SoftwareSourceCode
SoftwareApplication

FSF directory
swMATH
librairies.io
Pypi
NPM
Maven
Package Management

General schemes
The metadata landscape

Software schemes

- catalogs / registries
- DOAP
- Dublin Core
- PRONOM
- ADMS.SW
- Q7397 - software
- Q341 - free software
- Wikidata
- Digital Preservation
- SoftwareSourceCode
- SoftwareApplication
- schema.org
- Linked Data
- DBPedia
- PREMIS
- Package Management
- Pypi
- NPM
- Maven
- Scholarly Ecosystem
- Datacite
- libraries.io
- FSF directory
- swMATH
- OntoSoft
- Linked Data
Software schemes

DOAP

Q7397- software
Q341- free software

Dublin Core

PRONOM

Wikidata

Digital Preservation

PREMIS

Linked Data

Morane Gruenpeter
## Software Metadata Terms

### Identify
- identifier
- name
- author(s)
- contributor(s)
- version
- applicationCategory
- codeRepository

### Administrate
- maintainers (contact*)
- citation
- funder(s)
- license
- editor / publisher
- dates (created, modified, published)
- developmentStatus
Software Metadata Terms

**execute**
- buildInstructions
- issueTracker
- operatingSystem
- softwareRequirements
- runtimePlatform
- downloadUrl
- (memory, processor, storage)

**classify**
- description
- keywords
- supportingData
- referencePublication
- algorithms*
- readme (docs*)
The CodeMeta Generator link to online tool

CodeMeta generator

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
Outline

1 Introduction
2 The knowledge is in the source code!
3 Software Heritage: the universal source code archive
4 Data model and SWHID: the source code fingerprint
5 Recognizing software as a research output
6 The missing piece - the Metadata
7 Conclusion
Come in, we’re open!

Software Heritage

- universal source code archive
- intrinsic identifiers (SWHIDs)
- open, non profit, long term
- infrastructure for Open Science

You can help improve science!

- use SWH and save *relevant* source code
- build on SWH (see swmath.org and ipol.im)
- contribute to SWH- it is *open source*
- spread the word
Thank you! Any questions?

contact: morane@softwareheritage.org

P. Alliez, R. Di Cosmo, B. Guedj, A. Girault, M. Hacid, A. Legrand, N. Rougier
Attributing and Referencing (Research) Software: Best Practices and Outlook From Inria

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli
Referencing Source Code Artifacts: a Separate Concern in Software Citation
Computing in Science & Engineering, 2020, ISSN: 1521-9615