Archiving, referencing and attributing research software towards software as a first class citizen

Roberto Di Cosmo
Seminaire LIRIS, Lyon

November 21st, 2019
Outline

1. Software Source Code: a (forgotten) pillar of Science
2. Software Heritage
3. Archive and reference *all* the source code
4. Describe and cite *research* source code
5. The road ahead
Software source code: a precious part of our heritage

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.) 1985

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

Apollo 11 source code (excerpt)

```
P63SPOT3
CA BIT6    # IS THE LR ANTENNA IN POSITION 1 YET
EXTEND RAND CHAN33
EXTEND BZF P63SPOT4 # BRANCH IF ANTENNA ALREADY IN POSITION 1
CAF CODE500 # ASTRONAUT: PLEASE CRANK THE
TC BANKCALL # SILLY THING AROUND
CADR GOPERF1
TCF GOTOP00M # TERMINATE
TCF P63SPOT3 # PROCEED SEE IF HE’S LYING

P63SPOT4
TC BANKCALL # ENTER INITIALIZE LANDING RADAR
CADR SETPOS1
TC POSTJUMP # OFF TO SEE THE WIZARD ...
```

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”

Quake III source code (excerpt)

```
float Q_rsqrt( 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 = 0xf53759df - ( 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;
}
```
Source code is a *special* and endangered heritage

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*

Precious, endangererd *Executable* and *human readable* knowledge

key people are passing away …

no organised effort to catalog and archive it (more later)
Software is everywhere in modern research

[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you dont have the software, you dont have the data

Christine Borgman, Paris, 2018

Open Science: three pillars

The links in the picture are essential
The state of the art (in CS!) is far from ideal

<table>
<thead>
<tr>
<th>ICSE (Zannier, Melrik, Maurer, 2006)</th>
<th>complete absence of replication studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM TOSEM 2001 to 2006</td>
<td>60% of all papers have tools: only 20% installable</td>
</tr>
<tr>
<td>Collberg’s 2015 study</td>
<td><a href="http://reproducibility.cs.arizona.edu/">http://reproducibility.cs.arizona.edu/</a></td>
</tr>
<tr>
<td></td>
<td>601 mainstream papers: 508 with tools, only 40% installable</td>
</tr>
</tbody>
</table>

Main reasons

source code (or the right version of it) cannot be found
Where we stand

A wealth of initiatives!

- Policies: ACM Artifact Review and Badging, …
- Working groups: FORCE11, RDA, SPSO, …
- Metrics: Open Science Monitor (Elsevier!), …
- Repositories: FigShare, Zenodo, …

but …

Lack of recognition

- not (yet) a first class citizen
  - in the EOSC plan
  - in the scholarly works

Lack of proper guidance on how to

- *archive* and *reference* software
- choose a license
- *cite* a software project
<table>
<thead>
<tr>
<th>What is at stake</th>
<th>Research software artifacts must be properly...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archival</td>
<td>archived make it sure we can <strong>retrieve</strong> them (<strong>reproducibility</strong>)</td>
</tr>
<tr>
<td>Identification</td>
<td>referenced make it sure we can <strong>identify</strong> them (<strong>reproducibility</strong>)</td>
</tr>
<tr>
<td>Metadata</td>
<td>described make it easy to <strong>discover</strong> them (<strong>visibility</strong>)</td>
</tr>
<tr>
<td>Citation</td>
<td>cited (not the same as referenced!) to give <strong>credit</strong> to authors (<strong>evaluation</strong>)</td>
</tr>
</tbody>
</table>
A plurality of needs

**Researcher**
- archive and reference sw used in articles
- get credit for the software they develop
- verify/reproduce/improve results

**Laboratory/team**
- track software contributions
- produce up-to date report / web page

**University/Research Organization**
- central view of research software assets
- tech transfer
- impact metrics
Outline

1. Software Source Code: a (forgotten) pillar of Science
2. Software Heritage
3. Archive and reference *all* the source code
4. Describe and cite *research* source code
5. The road ahead
Collect, preserve and share the source code of all the software

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

Reference catalog
find and reference all the source code

Universal archive
preserve all the source code

damage, disaster, malicious, obsolete, deletion, attack, buggy, redundant, broken, outdated, obsolete, insecure, dangerous, broken

Research infrastructure
enable analysis of all the source code
The largest software archive, a shared infrastructure

Software Heritage

Source files
6,197,000,081

Commits
1,379,380,527

Projects
90,231,104
A peek under the hood

Global development history permanently archived in a unique git-like Merkle DAG

- ~400 TB (uncompressed) blobs, ~20 B nodes, ~280 B edges
So/f_tware Heritage for Research and Innovation

Reference platform for Big Code

- unique observatory of all software development
- big data, machine learning paradise: classification, trends, coding patterns, code completion...

First datasets are available!

- full graph of software development (~20Bn nodes, ~200Bn edges) see Pietri, Spinellis, Zacchirolì, MSR 2019
  https://dx.doi.org/10.1109/MSR.2019.00030
- MSR 2020 mining competition see https://2020.msrconf.org/track/msr-2020-mining-challenge#Call-for-Papers
An international, non profit initiative built for the long term

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Platinum sponsors

Golden sponsor

Silver sponsors

Bronze sponsors

Roberto Di Cosmo

(CC-BY 4.0) Research Software November 21st, 2019
Outline

1. Software Source Code: a (forgotten) pillar of Science
2. Software Heritage
3. Archive and reference *all* the source code
4. Describe and cite research source code
5. The road ahead
Software Heritage: a revolutionary infrastructure

- **universal archive** of all source code
- we archive *all* software: both research and non research
- we *proactively collect software* in a systematic way

- **intrinsic identifiers for reproducibility**
- identify software artefacts *without any third party*
- cryptographically strong, compatible with git hashes

Full guidelines available!  
https://www.softwareheritage.org/save-and-reference-research-software/

Save code now … in just a few clicks

Demo

My 2012 Parmap paper **before** and **after**; other links: Apollo 11 (and blog), Quake III Arena

Roberto Di Cosmo
1. Software Source Code: a (forgotten) pillar of Science
2. Software Heritage
3. Archive and reference *all* the source code
4. Describe and cite *research* source code
5. The road ahead
Many articles/guidelines

- reproducibility
- archival
- credit and evaluation

Most common limitations

- software is ’just data’
- citation = reference = DOIs
- citation produced by automated tools

A few remarkable exceptions

- ASCL (since 1999): metadata only, carefully curated
- geodynamics.org: source, documentation, metadata
- swmath.org: software catalog via articles

Software Citation WG at Inria (since 10/2018)

- leverage a 50 year experience, make recommendations
- read more https://hal.archives-ouvertes.fr/hal-02135891
Why it is not simple

Software is complex

- Structure: monolithic/composite; self-contained/external dependencies
- Lifetime: one-shot/long term
- Community: one man/one team/distributed community
- Authorship: complex set of roles (more later)
- 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”
Proposals for the scholarly world

Refined ontology for contributors
- Design, Architecture,
- Coding, Testing, Debugging,
- Documentation, Maintenance, Support,
- Management

see also CRediT, Geodynamics

Reference is distinct from citation
- **Reference** is for reproduction
- **Citation** is for credit

They must not be conflated

Beware of the numbers game: … do we really want an *s*-index?

Keep the human in the loop

When *credit* is at stake, automation/crowdsourcing is not enough!

Humans *are needed* to get quality information
First steps with HAL / Software Heritage

How it works, what is special

- **Generic mechanism:**
  - SWORD based
  - review process
  - versioning

**Today:** deposit .zip or .tar.gz file *(guide)*

**Tomorrow:** just provide the *SWH id*

Deposit/describe research software in HAL

- **author:** [https://hal.archives-ouvertes.fr/hal-01872189](https://hal.archives-ouvertes.fr/hal-01872189)
- **moderator:** [https://hal.archives-ouvertes.fr/hal-01876705](https://hal.archives-ouvertes.fr/hal-01876705)

Examples

LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …

Roberto Di Cosmo

(CC-BY 4.0) Research Software  November 21st, 2019  17 / 19
The swmath.org approach

Article based citation

See for example:

- SemiPar on swmath.org
Outline

1. Software Source Code: a (forgotten) pillar of Science
2. Software Heritage
3. Archive and reference all the source code
4. Describe and cite research source code
5. The road ahead

Roberto Di Cosmo (CC-BY 4.0) Research Software November 21st, 2019 19 / 19
Conclusion

Research software

- pillar of open science
- finally in the limelight

Doing it right is not easy

- simplistic approaches, "just data", ...
- soon part of research evaluation

You can help make a change

- leverage Software Heritage in conferences and journals for archival and reference
- join the conversation on software citation and software evaluation criteria
- tackle the scientific problems: big code, classification, infrastructure, etc.

Thank you!

Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchiroli
Building the Universal Archive of Source Code
Communications of the ACM, October 2018

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli
Identifiers for Digital Objects: the Case of Software Source Code Preservation
iPRES 2018: Intl. Conf. on Digital Preservation

Roberto Di Cosmo (CC-BY 4.0) Research Software November 21st, 2019 19 / 19
Appendix
Outline

6. Big Code
7. Milestones and breaking news
8. Under the hood
9. Inria’s commitment
10. Identifiers are not easy
11. Looking for the right identifiers
So/f_tware Heritage for Research and Innovation

Reference platform for *Big Code*

- unique observatory of all software development
- big data, machine learning paradise: classification, trends, coding patterns, code completion…

First datasets are available!

- full graph of software development (~20Bn nodes, ~200Bn edges) see Pietri, Spinellis, Zacchiolo, MSR 2019
  
  https://dx.doi.org/10.1109/MSR.2019.00030

- MSR 2020 mining competition see https://2020.msrconf.org/track/msr-2020-mining-challenge#Call-for-Papers

Roberto Di Cosmo
6 Big Code
7 Milestones and breaking news
8 Under the hood
9 Inria’s commitment
10 Identifiers are not easy
11 Looking for the right identifiers
Milestones

Summer 2015
The collection starts: first server, (very) early prototype

April 3rd 2017
Unesco - Inria agreement on software access and preservation.

December 7th 2018
Starting the mirror network

June 30th 2016
Public unveiling, with the first sponsors: Microsoft and DANS

June 7th 2018
Opening the archive to the world

February 26th 2019
Publication of the expert meeting Paris Call on Software Source Code
Breaking news: archiving public code

https://code.etalab.gouv.fr
Breaking news: SWHAP

Paris Call on Software Source Code

“We call to support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

SWHAP: an important step forward

- detailed guidelines to curate landmark legacy source code and archive it on Software Heritage
- intense cooperation with Università di Pisa and UNESCO
- open to all, we’ll promote it worldwide

https://www.softwareheritage.org/swhap
Thomas Jefferson, February 18, 1791

…let us save what remains: not by vaults and locks which fence them from the public eye and use in consigning them to the waste of time, but by such a multiplication of copies, as shall place them beyond the reach of accident.

Welcoming ENEA

- first institutional mirror
- increased resilience
- AI infrastructure for researchers
- stepping stone to an European joint effort
Inria’s ongoing contributions

Software Heritage
universal archive (research) software source code archived and referenced

Reproducibility

tools Guix (now with Software Heritage)
training/research RR workshops, MOOC

Research software curation
HAL - SWH bridge curation of metadata, and deposit in Software Heritage
URL decay disrupts the web of reference

Web links are not permanent (even permalinks)

there is no general guarantee that a URL… which at one time points to a given object continues to do so


URLs used in articles decay!

Analysis of IEEE Computer (Computer), and the Communications of the ACM (CACM): 1995–1999

- the half-life of a referenced URL is approximately 4 years from its publication date


An example from Astronomy

How Do Astronomers Share Data?

Pepe, Goodman, Muench, Crosas, Erdmann

dx.doi.org/10.1371/journal.pone.0104798
**DOI limitations**

Example: doi:10.1109/MSR.2015.10

- to find what 10.1109/MSR.2015.10 is, go to a resolver (e.g. doi.org)
- at this URL we find ...

**Architecture of the DOI infrastructure**

- DOI resolution *can change*
- content at URL *can change*
- no *intrinsic* way of noticing
- persistence based on *good will of multiple parties*
A system of identifiers is:

- a set of labels (the identifiers)
- mechanisms to perform:
  
<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>create a new label</td>
</tr>
<tr>
<td>Assignment</td>
<td>associate label to object</td>
</tr>
<tr>
<td>Retrieval</td>
<td>get object from a label</td>
</tr>
</tbody>
</table>

- optionally, mechanisms to perform:
  
<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification</td>
<td>check label and object</td>
</tr>
<tr>
<td>Reverse Lookup</td>
<td>get label from an object</td>
</tr>
<tr>
<td>Description</td>
<td>get metadata of an object</td>
</tr>
</tbody>
</table>
Mechanisms offered in some systems of identifiers

<table>
<thead>
<tr>
<th>Mechanism / System</th>
<th>Handle</th>
<th>DOI</th>
<th>Ark</th>
<th>PURL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Assignment</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Retrieval</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Verification</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>Reverse Lookup</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>Description</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
</tbody>
</table>
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!
An important distinction: DIOs vs. IDOs

The term “Digital Object Identifier” is construed as “digital identifier of an object,” rather than “identifier of a digital object” — Norman Paskin, 2010

DIO (Digital Identifier of an Object)
- digital identifiers for (potentially) non digital objects
- epistemic complexity (manifestations, versions, locations, etc.)
- need an authority to ensure persistence and uniqueness

IDO (Identifier of a Digital Object)
- digital identifiers (only) for digital objects
- can provide both integrity and no middle man
- broadly used in modern software development (git, etc.)

...for the core Software Heritage archive, IDOs are enough
Merkle tree (R. C. Merkle, Crypto 1979)

Combination of
- tree
- hash function

Classical cryptographic construction

- fast, parallel signature of large data structures, built-in deduplication
- satisfies all three criteria: gratis, integrity, no middle man!
- widely used in industry (e.g., Git, nix, blockchains, IPFS, …)
IDOs in Software Heritage: a worked example
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 would otherwise inhibit your freedom. This license is designed to protect users' freedom to share and change it. It is not a warranty that you can rely on. Instead, it gives you nothing for free; it gives you freedom to share and change the program.

Sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: 94a9ed024d385...
length: 35147
IDOs in Software Heritage: a worked example
IDOs in Software Heritage: a worked example

Directories

```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d6a34af6f52cf3c6fb0c2f7bd0648fdb255e7f AUTHORS
100644 blob 94a9ed824d38597993618152ea559a1688cbb5e2 LICENSE
100644 blob d9b2665a435a43f8a79a84e0867751dfb985c7bb MANIFEST.in
100644 blob 524175c2badb035b975f9f284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5b5bbdb038682b7793f4409492ed96788bb3 Makefile.local
100644 blob 8617980629cd24e080404f99f74b98685b3e67b README.db_testing
100644 blob 76b29f94c815e0869c414d38d7b7ce08ec514e README.dev
040000 tree e1e10eef948af6b93a0db8372af89f2e92618a bin
040000 tree 83e56d8beaf7793c77a45a345c80fcb8af503813 debian
040000 tree a34c9c5a213f8ec6c3f79816348d27955577af5 docs
100644 blob 2a6d32c6135a7287bd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caflbcb2d820da8dc0265b5b68180c setup.py
040000 tree 224bb4c1f4c67fca1d160b0fd2d06b94e7e1abf3 sql
040000 tree 8631c8c77bb69931681187ab52a5f51f48c6300be swh
040000 tree 8fb958b56ba8ed692f1209b2773b474c61d66c1 utils
```

```
id: 515f00d44e92c65322aa9bf3fa097c00db9c7d
```
IDOs in Software Heritage: a worked example

[Diagram showing the relationship between snapshots, releases, revisions, directories, and contents]
Revisions

<table>
<thead>
<tr>
<th>Details</th>
<th>Changes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author:</td>
<td>Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
</tr>
<tr>
<td>Committer:</td>
<td>Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
</tr>
<tr>
<td>Subject:</td>
<td>provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
</tr>
<tr>
<td>Parent:</td>
<td>fc3a8b59ca1df424d860f2c29ab07ee4dc35d10</td>
<td></td>
</tr>
</tbody>
</table>

```bash
tree 515f00d44e92c65322aaa9bf3fa097c00dddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07ee4dc35d10
```

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

IDOs in Software Heritage: a worked example

Releases

object c0c9f16b1e134f593e7567570a1761b156e6eb1
with type commit
and tag v0.0.51
tagged by Nicolas Dandrimont <nicolas@dandrimont.eu> on 2019-11-21 17:03:02 +0200

Release swh.storage v0.0.51

- Add new metadata column to origin_visit
- Update swh-add-directory script for updated API

----BEGIN PGP SIGNATURE-----
IQJZBABBACAdqBOjXv27bfxuanVWogQ0GRh/bn9bg0b1bqXUACgqG7AWLWo2+
neqR4y/a+b6S0g5R0f6h2a+xnVt2nXQ0+S+1KvEv/1b/5Kxaw8Ek7a/XKZ0t17f
a7pZ6p3q3Qbn6aC1-Yx9BfchzJL2yftrZkZVWXWq8txWVNAEmvJxT0h8q5phwhhAAoStz
ICBI2u/jXuXGR93eK4Pw2zZGxg+hBq5MWy3S5Dr6W7ZT7K7MjupPOLgy3pHP75yo
IGEEnWnoYH1Y1vm8Clh5b75mXJagB+becgjde6ubj2Z3jy+jpU9G4qBcysqY3hnr/FL
qs/2mulkYzy/zkhGj1jpv+hsbBlnhPsStH0zuojlYr9pKh2hSPF9u1h2h3kCinOx
kJ/jekAyWU8bM0exh+nK/Vj3lbrR3+yWBF:3q5pa1i/Vhoa1Y7m1Ed1ALCMnEp6KoKm7
qMXa1ltJ3g/EDf6q67G6JsDwePKPo/khmg9LQqQm3VJ4GmQ7U5n1qP3c40kZTV5xKoA
Ghzh1kPdhF4zuoWId4wyPZyv0e0U2XG63Vw9vO2ZkX3W+jm5ZMzdc9drj5UOBl
RvTTFusXEUxKGpokgXh5ySTvps1pdPc76USTKDo6eA4M1k0mGrwXCVQgPBo
mhhb5S4YMM0vS6FqyTS0plb1YK70t1PKRXUGXo9kKGwK5sKXZGzkx7jY7d/jo29
gu1g9gZOFqWQ2D0m1sL5UHFaTVsCMeqU80zJzFEJnlHvUk=
=K0xP
----END PGP SIGNATURE-----

id: 85083a5cc14a441c89dea73f5badf67c3f9c6afdb
Snapshots

commit 08fbeb2577010952eb3ceive1b69146c530a1d9158 refs/heads/atime
commit bs44a3a2e4e3f9e32e8d2a6c792cc4e0c6e1c7eb refs/heads/directory-listing-arrays
commit df9ebd0a89fc3356f86927b1ac0567273869c5 refs/heads/foo
commit c777f3e06eb9c2b78f46908f5a386f0df2de458ec88 refs/heads/master
commit e7ca1977f6c66d22d9047f5a5b317d9a8b44361a8fc2 refs/heads/tmp/copydwn-
commit 6422515f1565805a5650427b3554e4f52252e82e refs/heads/tmp/generic-releases

tag 201904313780c2f7e69665577f90f4df0f097c57f77f55 refs/tags/v0.0.1

tag 72a2191a384e359995edbd87bf00db872aeec9cd refs/tags/v0.0.10

tag 35996ca8abebd8b3e531b78f5a23b8bf4ffaf5cc refs/tags/v0.0.11

tag 33737827443ba95a53a7778b9c66720f0d5556 refs/tags/v0.0.12

tag 65f7465275253322c5f9031c12baa363cf3b42d5 refs/tags/v0.0.13

tag 5465267f566a4b5c61e444268d92a1e32f3bd refs/tags/v0.0.14

tag 586fa8605b3d45a9f5f9307613c3b3a19c7f refs/tags/v0.0.15

tag 8cd8bb5f4f990f363177742db289f660e5be51c refs/tags/v0.0.16

tag a54244e3f9f9ebe3e2f9ee3de8389a8c77d9 refs/tags/v0.0.17

tag 228af715590d1222e55855462a1896f9c493d9 ref/tags/v0.0.18

tag 6696794a405d497f8c0d24aad9d0e82536eef47c refs/tags/v0.0.19

tag 3c12f55a39fca32aa6ad5155052437342 refs/tags/v0.0.2

tag 3147c3d3e46c6f4927f0a10e90b1237e6f2727c refs/tags/v0.0.20

tag 2155e05dab118e02e0726e6b6b783a87908 refs/tags/v0.0.21

tag 3fbeb8c2872a5d8252124257afa5d8f85ffad1df refs/tags/v0.0.22

tag 8c0b8e8da4473f5a726789ae46ba16ac3c72ab3a4 refs/tags/v0.0.23

...
<table>
<thead>
<tr>
<th>IDO ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbbb5e2</td>
<td>full text of the GPL3 license</td>
</tr>
<tr>
<td>swh:1:dir:d198bc9d7a6bcf6db04f476d29314f157507d505</td>
<td>Darktable source code</td>
</tr>
<tr>
<td>swh:1:rev:309cf2674ee7a0749978cf8265ab91a60aea0f7d</td>
<td>a revision in the development history of Darktable</td>
</tr>
<tr>
<td>swh:1:rel:22ece559cc7cc2364edc5e5593d63ae8bd229f9f</td>
<td>release 2.3.0 of Darktable, dated 24 December 2016</td>
</tr>
<tr>
<td>swh:1:snp:c7c108084bc0bf3d81436bf980b46e98bd338453</td>
<td>a snapshot of the entire Darktable repository (4 May 2017, GitHub)</td>
</tr>
</tbody>
</table>

Current resolvers: [archive.softwareheritage.org](http://archive.softwareheritage.org) and [n2t.org](http://n2t.org)