

Software Heritage and Guix

Software Heritage to the rescue of reproducible Science

Valentin Lorentz (@vlorentz) / Antoine R. Dumont (@ardumont)

Software Engineers, Software Heritage
Inria

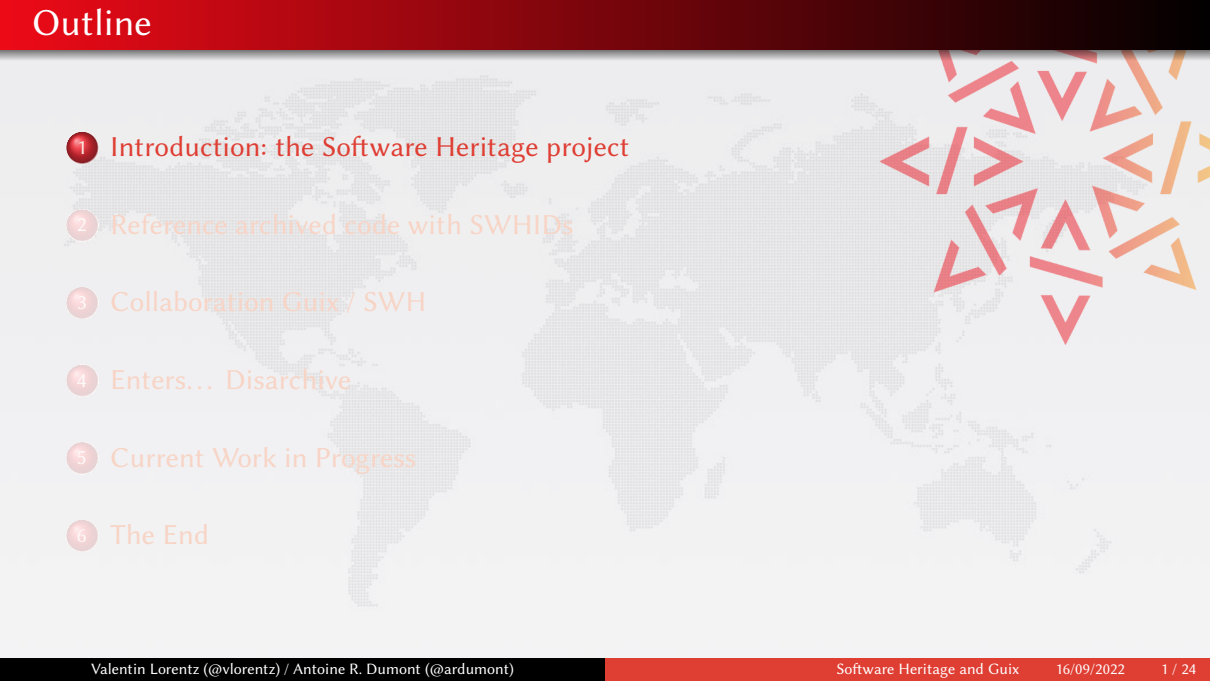
16/09/2022

Event 10 years of Guix, Paris 2022



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 
- 1 Introduction: the Software Heritage project
 - 2 Reference archived code with SWHIDs
 - 3 Collaboration Guix / SWH
 - 4 Enters... Disarchive
 - 5 Current Work in Progress
 - 6 The End

What is SoftwareHeritage?



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

The Universal Source Code Archive

Why an archive? Software is spread all around



Fashion victims

- many development platforms (popular forges: Guix, PyPI, npm, ...)
- various distribution places (standalone forges: gitlab, heptapod, cgit, gitea...)
- projects tend to migrate from one place to another over time

Why an archive? Software is spread all around



Fashion victims

- many development platforms (popular forges: Guix, PyPI, npm, ...)
- various distribution places (standalone forges: gitlab, heptapod, cgit, gitea...)
- projects tend to migrate from one place to another over time

One place...

... where can we find, track, search and recover *all* source code?

Why an archive? Software is fragile



A word cloud centered on the slide, featuring terms related to digital preservation challenges. The words are arranged in a circular pattern with varying font sizes and colors. The most prominent words are 'damage' (large, dark red), 'disaster' (large, purple), 'deletion' (medium, dark blue), 'malicious' (medium, brown), 'obsolete' (medium, purple), 'attack' (medium, blue), 'format' (medium, green), 'dependencies' (small, blue), 'dangling' (small, brown), 'wear' (small, brown), 'corruption' (small, brown), 'encryption' (small, blue), 'reference' (small, blue), 'storage' (small, brown), 'aging' (small, blue), 'media' (small, brown), and 'tear' (small, brown).

Like all digital information, FOSS is fragile

- link rot: projects are created, moved around, removed
- data rot: physical media with legacy software decay
- business-driven code loss (e.g. Gitorious, Google Code, Bitbucket, ...)

Why an archive? Software is fragile



A word cloud centered on the slide, featuring terms related to digital preservation challenges. The words are arranged in a circular pattern with varying font sizes and colors. The most prominent words are 'damage' (large, dark red), 'disaster' (large, purple), 'deletion' (large, dark blue), 'malicious' (medium, brown), 'obsolete' (medium, purple), 'attack' (medium, blue), 'format' (medium, green), 'dependencies' (small, blue), 'aging' (small, blue), 'tear' (small, blue), 'dangling' (small, blue), 'wear' (small, blue), 'corruption' (small, blue), 'encryption' (small, blue), 'reference' (small, blue), and 'storage' (small, blue). The background of the slide features a faint world map and a decorative pattern of colorful triangles on the right side.

Like all digital information, FOSS is fragile

- link rot: projects are created, moved around, removed
- data rot: physical media with legacy software decay
- business-driven code loss (e.g. Gitorious, Google Code, Bitbucket, ...)

If a website disappears you go to the Internet Archive...

where do you go if (a repository on) GitHub or GitLab goes away?



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Main Objectives

- Collect, Preserve and Share

Reference catalog



find and **reference** all
software source code

find and **reference** all
software source code



preserve forever archived
software source code



Research infrastructure



- enable analysis of software source code
- make every piece identifiable
- and freely available...

Research infrastructure



- enable analysis of software source code
- make every piece identifiable
- and freely available...

Reproducibility

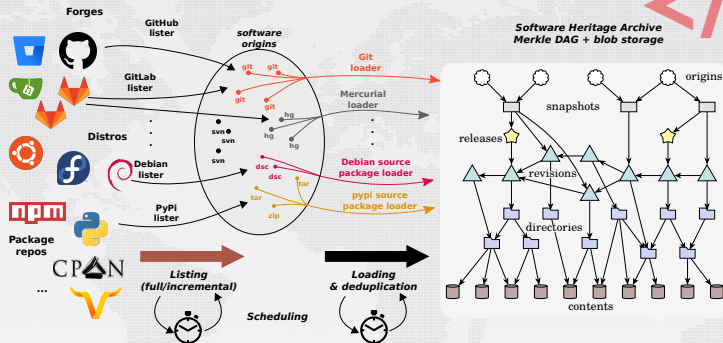


- ... exactly as it was when archived (as much as possible)
- for all research software artefacts

Our principles

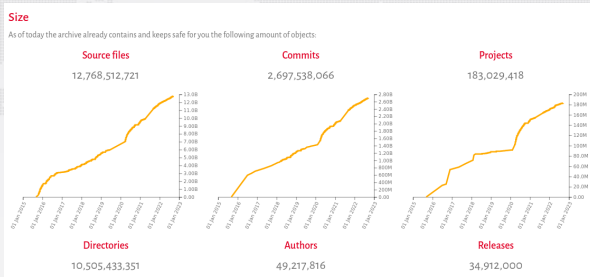


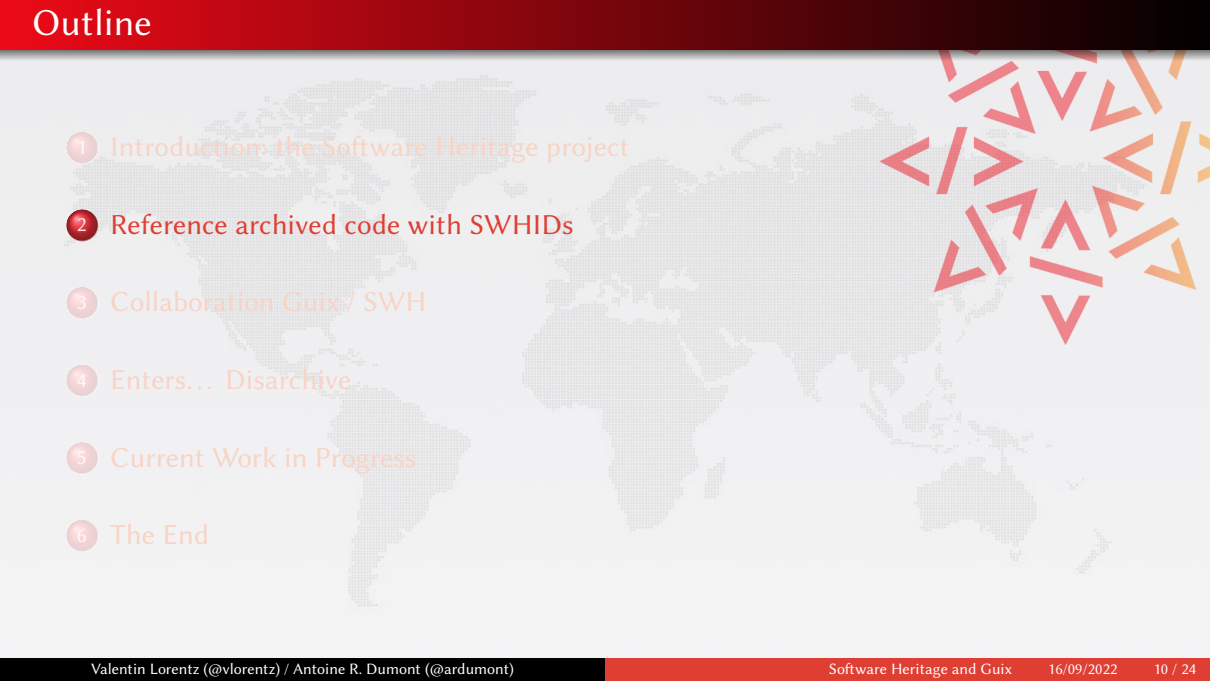
Under the hood: Automation and storage



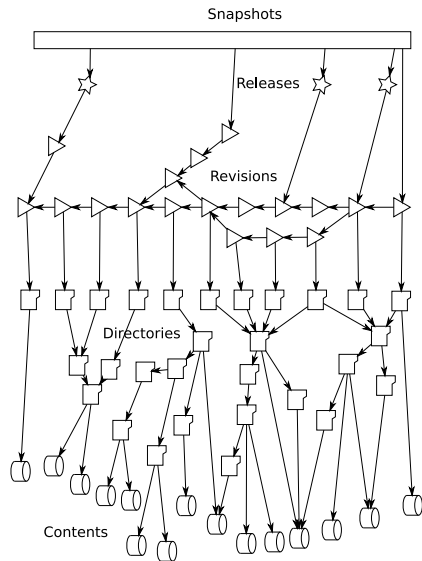
Global development history permanently archived in a uniform data model

- over 12 billion unique source files from over 180 million software projects
- ~900 TB (uncompressed) blobs, ~25 B nodes, ~300 B edges



- 
- 1 Introduction: the Software Heritage project
 - 2 Reference archived code with SWHIDs
 - 3 Collaboration Guix/ SWH
 - 4 Enters... Disarchive
 - 5 Current Work in Progress
 - 6 The End

Uniform Data Model



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.

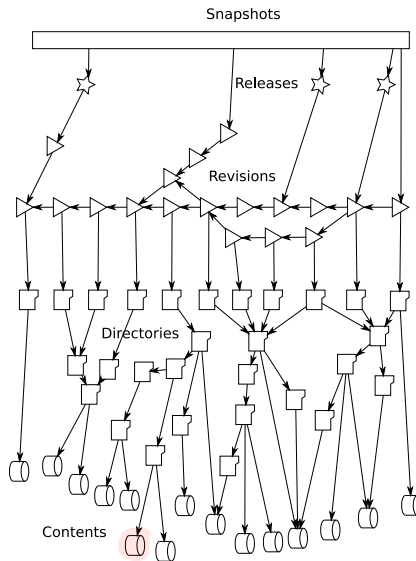
The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all 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 prevent anyone from denying you

sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: **94a9ed024d385...**
length: 35147

Data Model: A worked example



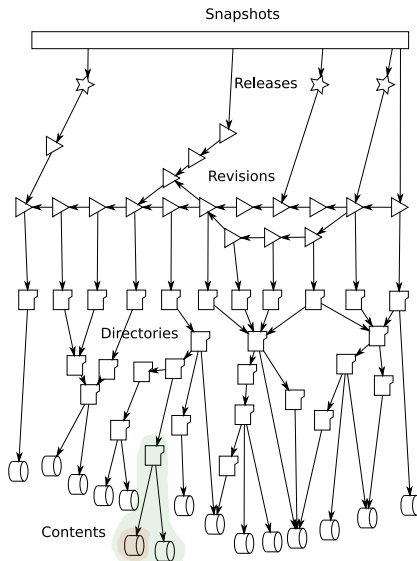


Directories

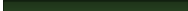
```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d0a34af6f52cf3cf6b0c2f7bd0648fbd255e77f AUTHORS
100644 blob 94a9ed024d3859793618152ea559a168bbcbb5e2 LICENSE
100644 blob d9b2665a435a43f8a79a84e0867751dfb095c7bb MANIFEST.in
100644 blob 524175c2bad0b35b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5bbddb038682ba7793f440492ed9678bb3 Makefile.local
100644 blob 8617980629cd24e6080404f09aa749b085b3e07b README.db_testing
100644 blob 76b29f94cf815e0869c414d38d78d7ce08ec514e README.dev
040000 tree e1e10ecf948af0b93adb0372afc89f12e92618a bin
040000 tree 83e56d0beaf7793c77a45a345c80fcb8af503013 debian
040000 tree a34c9c4ba213f0cedc67f9816348d27955577af5 docs
100644 blob f2a6d32c6135aa7287bbd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caf1bbc2d820da8dc026cb5b68180bc setup.py
040000 tree 224bb4c1f4c67fca1d160bfd2d06094e7e1abf3 sql
040000 tree 8631c9cd77bbe993168107ab5baf51f40c6300be swl
040000 tree 8fb905b56ba8ed692f1209b2773b474c6c1d66c1 utils
```

id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d

Data Model: A worked example



Revisions

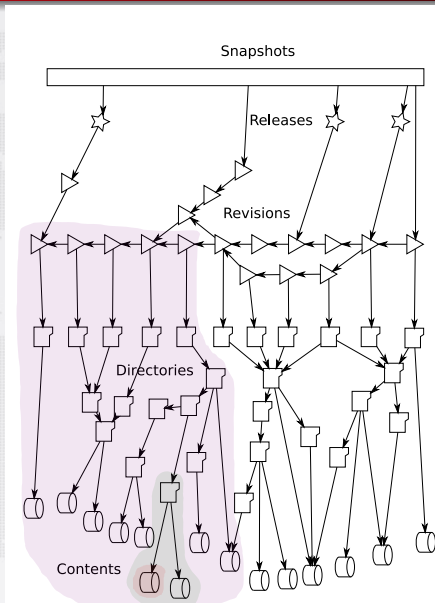
Details	Changes	Files
SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6		
Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Subject: provenance.tasks: add the revision -> origin cache task		
Parent: fc3a8b59ca1df424d860f2c29ab07fee4dc35d10 : test...storage: properly pipeline origin and cont...		
provenance.tasks: add the revision -> origin cache task		
swf/storage/provenance/tasks.py  77		

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](#)

Data Model: A worked example



Releases

tag v0.0.51
Tagger: Nicolas Dandrimont <nicolas@dandrimont.eu>
Date: Wed Aug 24 14:36:03 2016 +0200

Release sw.h.storage v0.0.51

- Add new metadata column to origin_visit
- Update sw.h-add-directory script for updated API [...]

commit c0c9f16b1e134f593e7567570a1761b156e6eb1d

object c0c9f16b1e134f593e7567570a1761b156e6eb1d
type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> 1472042163 +0200

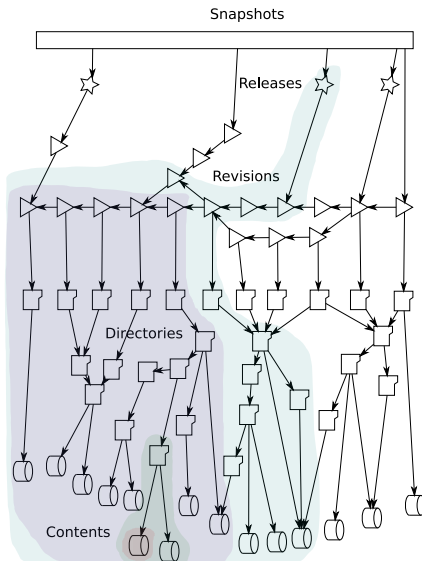
Release sw.h.storage v0.0.51

- Add new metadata column to origin_visit
- Update sw.h-add-directory script for updated API
- BEGIN PGP SIGNATURE-----

iQIzBAABCAAdBQJXvZTNFhxuWNvbGFzQGRhbmRyaW1vbnQuZXUACgkQ7AWLMo2+
neqorw//aq6SOb5DijzEa+kWN3rXgVS+1K1vEVh1wnKAwx8eKJ7aX2kEiLDtt7uf
ahpZ6pz3q8nqs6aC1+YrxBfcih3L2YtrdZeWXXWqr8xWNMaEoYDb8qaphwh8AD5t2
ICBlit2ujtXuCrDt93eKPKPwvZXg+hB0sMWy35Dr6jW7Z7K4Mu/PgGlyLHPY55yo
IGEndWno7VfH1Vm6t1n5qB7l5mXRaQA+becqddubTZ2xij+jpIUqC8cyqN3hm/fL
qsJ2mu8kyz3t8tG/H1/pV+I5OwBlnPoS5TH0tuoJEvGPK/dHSP79QuHDHZFkCao
klj6kAWyU80Mxb+nKVjleLbrR3+yWBFj3Qp5a1/V8oOTh6E1dAlcNMpEaKCoKtMt
d/gMRax1l1/g0EDfnsW67G6sDwKPKPHngfVLQ3nV3GaQQTnu1RpMz006H9/tAwzC
Gg/K1PdHT4hz0i46wYPZyje0U2VXGFu6vVU9vFQ4ZR/Wjn+0zMzdcRdrJl5UOMn
RpTTFUbsXUeXHGOpgXhSYTnvp1gdPc76U5TsK0aGe84AZm1lk0mGrwXCvFPqlyo
nhhibB5HBNMoqyF6yTSOpUbyK70tpYRRUGKWDeRK0wKSxkWUzGtKzy6jYqJJo29
gulwgZQif5qWQC80ontAL2+HvPfaVycKMejUhg62cP/+EHLvUk=
=kOxP
-----END PGP SIGNATURE-----

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

Data Model: A worked example



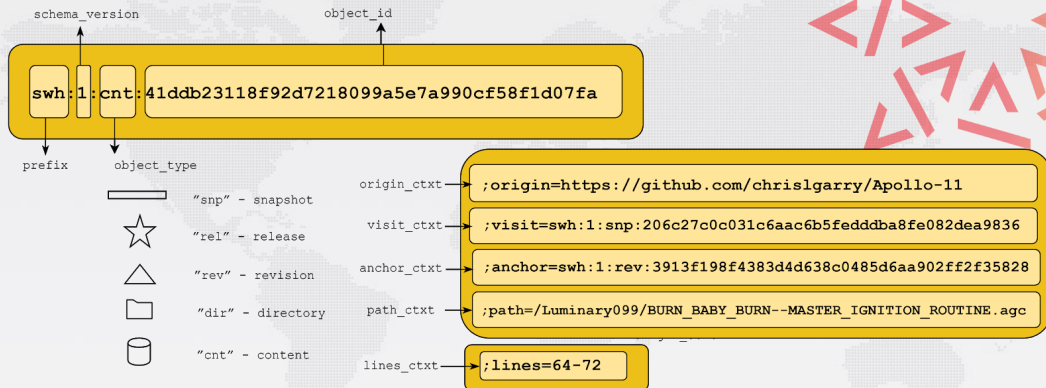
git show-refs

Snapshots

```
commit 08ffeb25770109525eb3ce21691466c53a1d9158 refs/heads/atime
commit ba5443a24e3f9fe323a46c292cec4fcbe61c67eb refs/heads/directory-listing-arrays
commit d69e0dbf892383ff6589b27f1e1c05d27238d9c5 refs/heads/foo
commit cf7ff9eea0eb22f8946908f5a8019f67de468e08 refs/heads/master
commit 7eca197fc66d2024047e54b1ed9e8b44361a0fc2 refs/heads/tmp-directory-add
commit 642a205f37de85005a85d427b53ee4fb2252e82e refs/heads/tmp/generic-releases
tag 20f043b1379cf768d966597799fd4907c757f755 refs/tags/v0.0.1
tag 72a21991a384e539996dbb867bfb0bee72aee2cd refs/tags/v0.0.10
tag 3590e0ca0ebb070e5b376705fa230bbfa4ffa5cc refs/tags/v0.0.11
tag 33378427a403ba569a67777b8d58f6674fbc6556 refs/tags/v0.0.12
tag 06f74652755b327cf590311c2bfa036cf3b4b35d refs/tags/v0.0.13
tag 5a6325fe86ab854b581d7442667d92a11e32f3bd refs/tags/v0.0.14
tag 586fba4e580b4f5fab05f599367643cbcb1a9c7f refs/tags/v0.0.15
tag 8cd8b885f4098bf363177742bd289f660e5be51c refs/tags/v0.0.16
tag a542444ee3f0fbed35efb202fee035c809abc7d6 refs/tags/v0.0.17
tag 228a2f1650dd12222e556559462e1e06fc4993d9 refs/tags/v0.0.18
tag 606979a4ca05d497fc0d24aad0dc82636ef47c refs/tags/v0.0.19
tag 32bf5a59fc2a323baa6d5f15a6ad5382ec275a67 refs/tags/v0.0.2
tag 3147c3d31ec46cf6492f881e908b1237ebdff2c7 refs/tags/v0.0.20
tag 215ea50daball1e082e0b72e76eb4b6073a87908 refs/tags/v0.0.21
tag 3fb168c2072a5d6252124257a1e5dfc0f5ffa1df refs/tags/v0.0.22
tag 8cdbee8da4d73fc5d262789e460a16ac3c72aba4 refs/tags/v0.0.23
...
```

id: b464cad1b66fff266a37b46ea6e7a04b545e904b

Meet the SWHID intrinsic identifiers

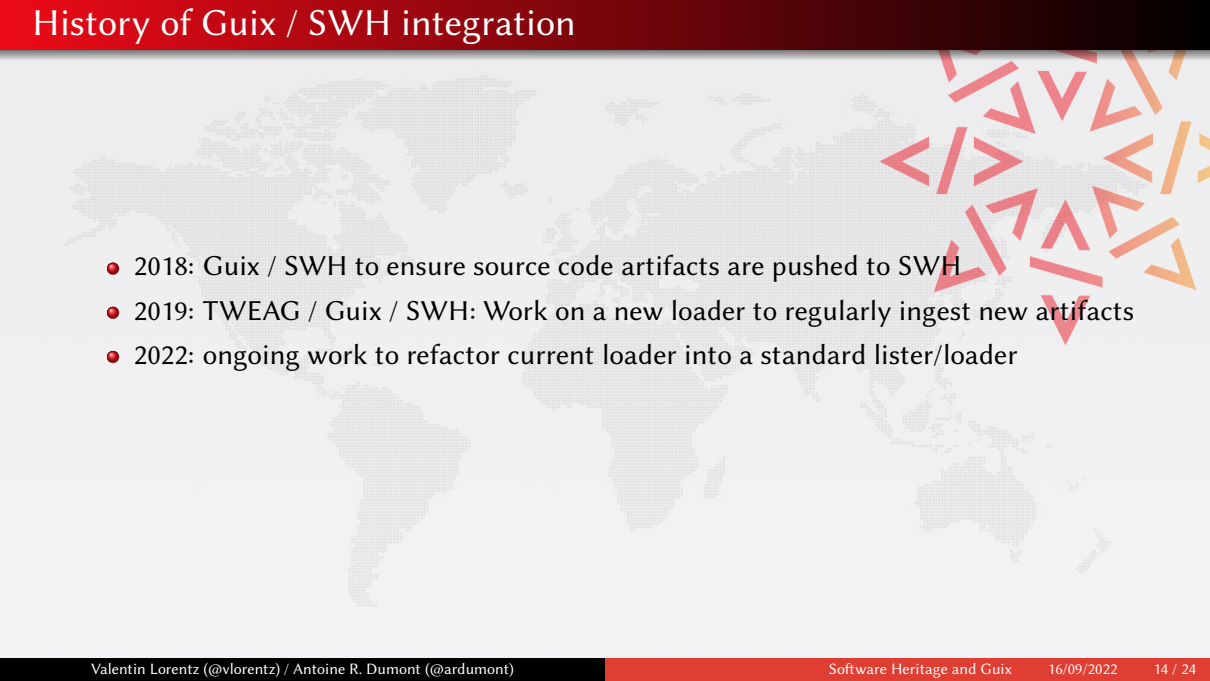


- 
- 1 Introduction: the Software Heritage project
 - 2 Reference archived code with SWHIDs
 - 3 **Collaboration Guix / SWH**
 - 4 Enters... Disarchive
 - 5 Current Work in Progress
 - 6 The End

How does this relate to Guix?

- Nothing is eternal, source code (in all forms) disappears
- Hopefully, SWH keeps a copy of everything
- **Guix ensures source code is archived in SWH when building** ("Save Code Now")
- After source code actually disappears, falls back to SWH when rebuilding ("Software Heritage Vault")

History of Guix / SWH integration

- 
- 2018: Guix / SWH to ensure source code artifacts are pushed to SWH
 - 2019: TWEAG / Guix / SWH: Work on a new loader to regularly ingest new artifacts
 - 2022: ongoing work to refactor current loader into a standard lister/loader

Reproducibility is of the essence!

Current situation

- Persistent intrinsic identifiers (SWHID) are not (yet?) package manager standard
- Guix (and other) package managers reference tarball hashes, instead of hashes of the content

```
(define-public ...  
  (package  
    ...  
    (source (origin (method url-fetch)  
                    (uri (string-append  
                        "https://..." version ".tar.gz"))  
                    (sha256 (base32 "03mwi1l3354x52nar...")))))  
    ...
```

Reproducibility is of the essence!

Current situation

- Persistent intrinsic identifiers (SWHID) are not (yet?) package manager standard
- Guix (and other) package managers reference tarball hashes, instead of hashes of the content

```
(define-public ...  
  (package  
    ...  
    (source (origin (method url-fetch)  
                    (uri (string-append  
                        "https://..." version ".tar.gz"))  
                    (sha256 (base32 "03mwi1l3354x52nar..."))))  
    ...
```

Solutions

- make (non-specific swh) SWHID standard or rebuild original bit-for-bit tarball

How to rebuild original tarballs?

pristine-tar

- <https://manpages.debian.org/bullseye/pristine-tar/pristine-tar.1.en.html>
- `xdelta`: binary diffs of tar headers' content and order
- `zgz`: guessing compression parameters

Limitations

- it is brittle (relies on reference Tar implementation)
- it produces large diffs

- 
- 1 Introduction: the Software Heritage project
 - 2 Reference archived code with SWHIDs
 - 3 Collaboration Guix/ SWH
 - 4 Enters... Disarchive**
 - 5 Current Work in Progress
 - 6 The End

Discussions

- "gforge.inria.fr to be taken off-line in Dec. 2020"
<https://issues.guix.gnu.org/42162>
- "lookup ingested tarballs by container checksum"
<https://forge.softwareheritage.org/T2430>

New software

- Disarchive by Timothy Sample <https://git.ngyro.com/disarchive/>

Principles

- Manifest of tarball fields: entry order, PAX headers, ...
- References the root directory in SWH
- WIP: guessing compression parameters/implementations (using `zgz`)
- -> rebuild original `.tar`, then original `.tar.{gz,xz}`

Example manifest (1/2)

Example manifest (1/2)

```
(disarchive
  (version 0)
  (tarball
    (name "test-archive.tar")
    (digest (sha256 "0da9fa3e7b360533678338871d9dd36f3..."))
    (default-header
      (chksum (trailer " "))
      (magic "ustar ")
      (version " \x00")
      (devmajor 0 (source "" (trailer "")))
      (devminor 0 (source "" (trailer "")))
      (data-padding ""))
    ...
```

Example manifest (2/2)

Example manifest (2/2)

```
(disarchive
...
  (headers
    ("test-archive/" (mode 493) (checksum 4291) (typeflag 53)
    ("test-archive/file-a" (size 15) (checksum 4849))
    ("test-archive/file-b" (size 15) (checksum 4850)))
  (padding 6656)
  (input (directory-ref
    (version 0)
    (name "test-archive")
    (addresses
      (swhid "swh:1:dir:902b1e94f0f5efdde6..."))
      (digest (sha256 "277dec b2666f4832ef64a...")))))
```

Planned integration of SWH with Disarchive

Currently

- SWH does not store Disarchive manifests yet

Plan

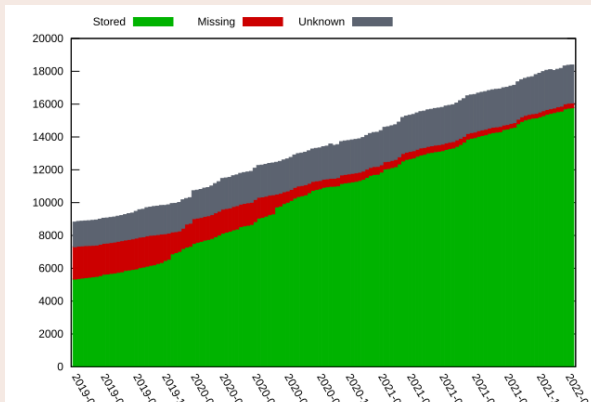
- Run Disarchive every time SWH loads a tarball
- Add the manifest to the Archive
- when someone requests `tarball-hash`, rebuild from the manifest

- 
- 1 Introduction: the Software Heritage project
 - 2 Reference archived code with SWHIDs
 - 3 Collaboration Guix / SWH
 - 4 Enters... Disarchive
 - 5 Current Work in Progress**
 - 6 The End

NixGuix manifests coverage in SWH

goal: 100% coverage

- currently missing sources due to technical limitations: bare files, directories, patches
- Redesign in progress to deal with such limitations



Integration

- code dump at <https://git.ngyro.com/swh/>
- needs to be reviewed and merged

- 
- 1 Introduction: the Software Heritage project
 - 2 Reference archived code with SWHIDs
 - 3 Collaboration Guix / SWH
 - 4 Enters... Disarchive
 - 5 Current Work in Progress
 - 6 The End

Links

- [Software Heritage website](#)
- [SWH's Archive](#)
- [Development forge](#)
- [guix lint](#)

We are hiring

[We are hiring devs and sysadmins](#)

Questions?

And thanks for your time!