Fulfilling Source Code Distribution Requirements

Alexios Zavras, Stefano Zacchiroli

Intel, alexios.zavras@intel.com
Software Heritage, zack@upsilon.cc

4 February 2018
CopyleftConf
Brussels, Belgium

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

The setup

- Intel is a *large* org which delivers a *lot* of software
- Software is a combination of own and FOSS components
- Many components have a legal source code distribution requirement
  - we also might deliver source in other cases

The legal requirement

For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. — *GPLv2*

Complete Corresponding Source (CCS)

Different terms used

- GPLv2: “complete corresponding machine-readable source code” / “accompany”
- GPLv3: “Corresponding Source” / “convey”
- MPLv2: “Source Code Form” / “made available”
- EPLv2: “Source Code” / “made available”

The problem

In an ideal world

- Fool-proof processes in place
  - collect, package, and provision
- Set it up once, always working

Practical considerations

- People change roles or leave
- Re-organizations happen
- Things get forgotten

Use cases

Trying to build an internal service

- Our delivery contains our own FOSS `sw.tar.gz`
- Our delivery contains `gcc-8.2`
- Our delivery contains `gcc` snapshot of revision `267928`
- Our delivery contains `gcc-8.2` patched with `patches.tar.gz`

Functional requirements

We need to be able to:

- provide our own software package
- refer to a “well-known” FOSS component
  - with release version or unique revision
- combine the two
  - well-known component with own patches

Functional requirements

We need to be able to:

- provide our own software package
- refer to a “well-known” FOSS component
  - with release version or unique revision
- combine the two
  - well-known component with own patches

Great Idea

- Can we *outsource* the fulfilment of these requirements?

The idea

Is it compliant?

GPL FAQ: Can I put the binaries on my Internet server and put the source on a different Internet site?

- *[v3] Yes. Section 6(d) allows this. However, you must provide clear instructions people can follow to obtain the source, and you must take care to make sure that the source remains available for as long as you distribute the object code.*
- *[v2] The GPL says you must offer access to copy the source code "from the same place"; that is, next to the binaries. However, if you make arrangements with another site to keep the necessary source code available, and put a link or cross-reference to the source code next to the binaries, we think that qualifies as "from the same place".*

The idea

Is it compliant?

GPL FAQ: Can I put the binaries on my Internet server and put the source on a different Internet site?

- *[v3] Yes. Section 6(d) allows this. However, you must provide clear instructions people can follow to obtain the source, and you must take care to make sure that the source remains available for as long as you distribute the object code.*
- *[v2] The GPL says you must offer access to copy the source code "from the same place"; that is, next to the binaries. However, if you make arrangements with another site to keep the necessary source code available, and put a link or cross-reference to the source code next to the binaries, we think that qualifies as "from the same place".*

Wouldn't it be great if *someone* could fulfill our requirements?

The Software Heritage Project

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Our mission

Collect, preserve and share the *source code* of *all the software* that is publicly available.

Past, present and future

Preserving the past, *enhancing* the present, *preparing* the future.

Our principles

Our principles

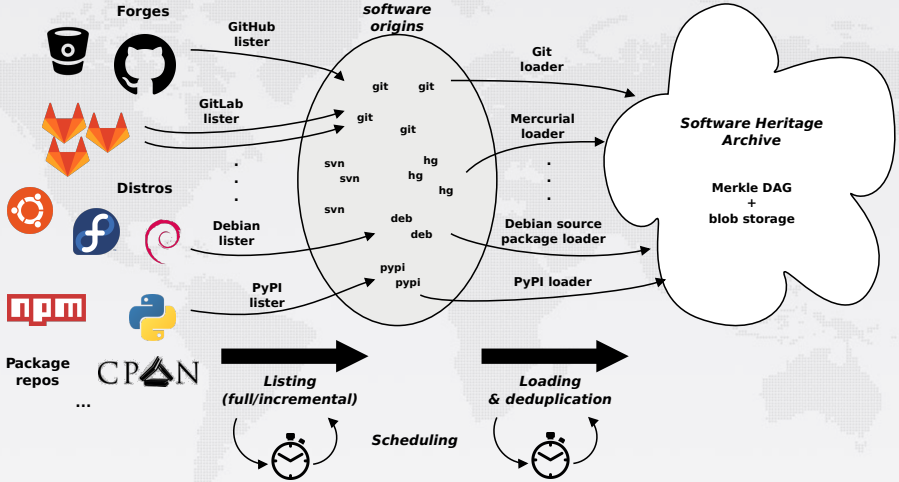
Open approach

- open source
- transparency

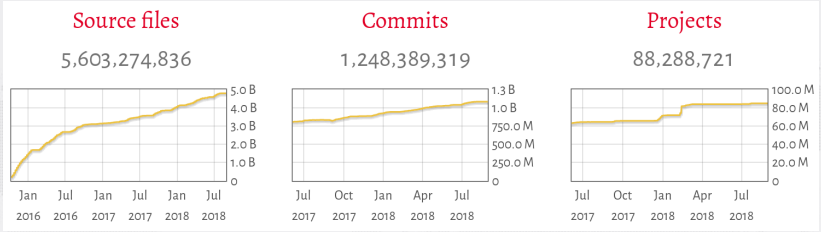
In for the long haul

- non profit
- replication & mirrors

Pull archiving — we crawl forges

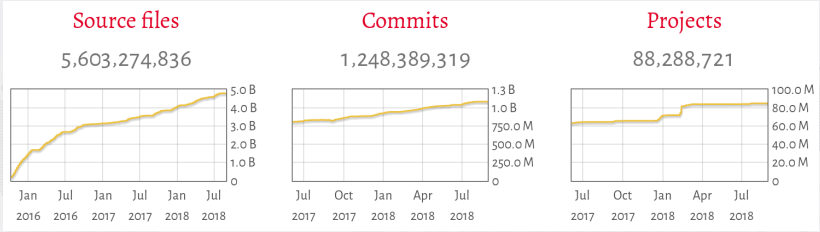


Archive coverage — archive.softwareheritage.org



GitHub · debian · GitLab · Google code · {P} · python Package Index · GITORIOUS · GNU · HAL archives-ouvertes.fr · Inria inventeurs du monde numérique

Archive coverage — archive.softwareheritage.org



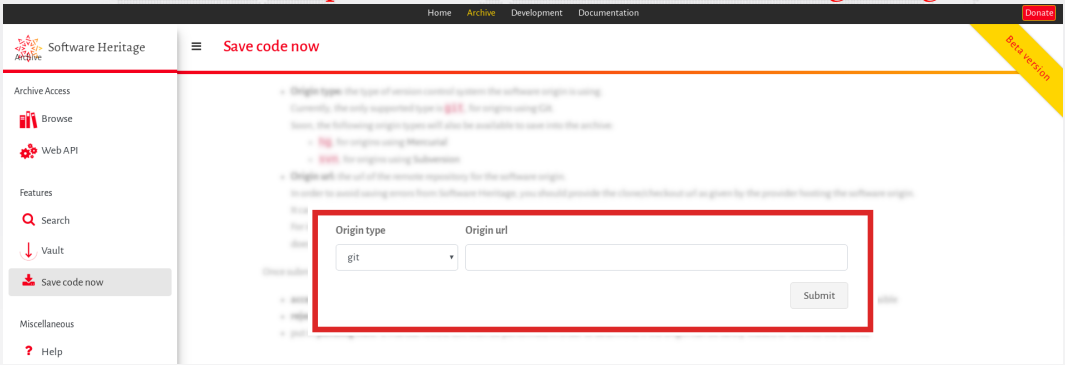
GitHub · debian · GitLab · Google code · {P} · python Package Index · GITORIOUS · GNU · HAL archives-ouvertes.fr · Inria inventeurs du monde numérique

- 200 TB (compressed) blobs, 6 TB database (as a graph: 10 B nodes + 100 B edges)
- The *richest* public source code archive, … and growing daily!

Push archiving — save code now

- on demand, prioritized crawling of your software origin of choice
- also available as "take a new snapshot" for already archived origins

available at `https://archive.softwareheritage.org`



Push archiving — direct deposit

Deposit service

- complement regular (pull) crawling of forges and distributions
- restricted access (i.e., not a warez dumpster!)
- `deposit.softwareheritage.org`

Tech bits

- SWORD 2.0 compliant server, for digital repositories interoperability
- RESTful API for deposit and monitoring, with CLI wrapper

Accessing archived code

After ingestion a deposit becomes an integral, permanent part of the archive.

- it has a persistent identifier
  - e.g., `swh:1:rev:a86747d201ab8f8657d145df4376676d5e47cf9f`
- it can be browsed online at `archive.softwareheritage.org`
  - e.g., `https://archive.softwareheritage.org/browse/swh:1:rev:a86747d201ab8f8657d145df4376676d5e47cf9f`

Accessing archived code

After ingestion a deposit becomes an integral, permanent part of the archive.

- it has a persistent identifier
  - e.g., `swh:1:rev:a86747d201ab8f8657d145df4376676d5e47cf9f`
- it can be browsed online at `archive.softwareheritage.org`
  - e.g., `https://archive.softwareheritage.org/browse/swh:1:rev:a86747d201ab8f8657d145df4376676d5e47cf9f`

Software Heritage Persistent Identifiers (PIDs)

- adoption: Wikidata, SPDX, Identifiers.org, US NTIA (under consideration), IANA (ongoing submission)
- Di Cosmo, Gruenpeter, Zacchiroli. *Identifiers for Digital Objects: the Case of Software Source Code Preservation.* In proceedings of iPRES 2018.
- `https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html`

Downloading archived code

- source code is thoroughly deduplicated within the Software Heritage archive
- bulk download of large artefacts (e.g., a Linux kernel release) requires collecting millions of objects
- the Software Heritage Vault cooks and caches source code bundles for bulk download needs

Downloading archived code

- source code is thoroughly deduplicated within the Software Heritage archive
- bulk download of large artefacts (e.g., a Linux kernel release) requires collecting millions of objects
- the Software Heritage Vault cooks and caches source code bundles for bulk download needs

```
$ curl -X POST /api/1/vault/revision/a86747d2.../gitfast
{
  'fetch_url': '/api/1/vault/revision/a86747d2.../gitfast/raw/',
  'progress_message': None,
  'status': 'new',
  'id': 4,
  'obj_id': 'a86747d201ab8f8657d145df4376676d5e47cf9f',
  'obj_type': 'revision_gitfast'
}
```

Downloading archived code

- source code is thoroughly deduplicated within the Software Heritage archive
- bulk download of large artefacts (e.g., a Linux kernel release) requires collecting millions of objects
- the Software Heritage Vault cooks and caches source code bundles for bulk download needs

```
$ curl -X POST /api/1/vault/revision/a86747d2.../gitfast
{
  'fetch_url': '/api/1/vault/revision/a86747d2.../gitfast/raw/',
  'progress_message': None,
  'status': 'new',
  'id': 4,
  'obj_id': 'a86747d201ab8f8657d145df4376676d5e47cf9f',
  'obj_type': 'revision_gitfast'
}

$ curl -O dump.gz /api/1/vault/revision/a86747d2.../gitfast/raw/
$ git init
$ zcat dump.gz | git fast-import
$ git checkout HEAD
```

Next steps @Intel

- Gradual deployment
- Generation of SwHer pointers
- Integration with sw production and delivery
- Fine-tuning
  - check whether already available vs. upload everything
  - software optimization and customization

Future ideas for copyleft

- No requirement for physical transmission
  - of CCS, license text, etc.
- Allow provision of a pointer to required material

Wrapping up

- long-term hosting of CCS archives can be onerous in the real-world
- it is A-OK to outsource that responsibility to third parties
- Software Heritage crawls (pull) all FOSS and also accepts push deposits
- Intel and Software Heritage are working together on practical FOSS tooling to outsource CCS hosting to the Software Heritage archive

Come and join us!

- alexios.zavras@intel.com, zack@upsilon.cc
- https://www.softwareheritage.org
- https://archive.softwareheritage.org
- https://deposit.softwareheritage.org