FOSSology - OSS for Open Source License Compliance

Anupam Ghosh (anupam.ghosh@siemens.com)
Overview: Contents

1. Introduction FOSSology
2. What is FOSSology
3. Motivation
4. What FOSSology needs
5. Introduction Project Builder
6. The ProjectBuilder Project
7. Build Native Fossology Packages
8. To get container running in the continuous build
9. Conclusion
10. Where to see it
The Problem Actually

You know these examples

Distributing open source software requires to

- Provide licenses of involved software
- Provide copyright statements of involved authors
- Provide disclaimers
- ... and much more
It is about finding licenses

Finding Licenses

- License texts
- References to licenses
- Written texts explaining licensing
- License relevant statements

![License text example](image-url)
Problem of many Licenses (“Proliferation”)

Open Source and Reuse

- It is natural that an OSS project reuses available
  https://github.com/fossology/fossology
- Likely OSS from other projects is found
- For example, FOSSology will find 25 other licensing relevant text occurrences in Apache thrift
How does FOSSology work?

See more details the Basic Workflow Description: https://www.fossology.org/get-started/basic-workflow

- Upload an open source package to the server
- Select scan agents that analyze the software

- Review what scanners have found
- Review license occurrences and correct findings if necessary

- Generate report output
- For example list of licenses or SPDX

Upload OSS Package

Review and Adjust ("Clearing")

Generate
What is the point of FOSSology?

See more details the Basic Workflow Description: https://www.fossology.org/get-started/basic-workflow

- Upload an open source package to the server
- Select scan agents that analyze the software
- Review what scanners have found
- Review license occurrences and correct findings if necessary
- Generate report output
- For example list of licenses or SPDX
**Key facts**

- Linux Foundation collaboration project
- GPLv2 licensed
- Linux, and only Linux application
- Mostly C/C++ and PHP
- Frontend runs on Apache httpd
- Provides also a Command Line Interface
- Backend schedules multiple Agents in parallel
- PostgreSQL as database
- Provides scripts for Docker and Vagrant
New Features
Feature: SPDX Import

SPDX Import allows for applying SPDX license analysis information to uploaded source code packages

Use Case

- Licensing information in SPDX files require also to see the original source code
- If you receive an SPDX file from another (unknown) organisation, review is maybe necessary
- How can I review SPDX license information from other source?

Solution

- FOSSology allows for uploading SPDX files
- Select “Report import” from the “Upload menu”
- Select different options for importing
- Actually it is a little bit difficult
  - How to deal with found in file and concluded licenses?
  - How to deal with new licenses -> create candidates
  - How to deal with conclusions -> take over or stage them?

Licensing information in SPDX files require also to see the original source code

If you receive an SPDX file from another (unknown) organisation, review is maybe necessary

How can I review SPDX license information from other source?
1. Select the folder that contains the upload: [Software Repository]
2. Select the upload you wish to edit: [zlib128.zip from 2017-10-22 13:59]
3. Select report to upload: [Browse... SPDX2_zlib128.zip_1508719996-spdx.rdf]
4. Select how the information should be imported:
   - Create new licenses as
     - license candidate
       - Note: license candidates as scanner findings are currently not handled correctly in the UI
     - new license
   - Add the License Info as findings from
     - SPDX tag of type licenseInfoInFile
     - SPDX tag of type licenseConcluded
   - Add concluded licenses as decisions
     - also overwrite existing decisions
     - import as "to be discussed"
   - Add the copyright information as textfindings

Upload and Import
(Note: Importing SPDX and reuse of licensing analysis information)
**Feature: Analysis Documentation**

**SPDX Import** allows for applying SPDX license analysis information to uploaded source code packages.

**Use Case**

- Exchanging licensing documentation with SPDX is fine, but ...
- Can I have documentation of my analysis?
- Can I provide comprehensive reporting what was analysed?
- **How can tell others what needs to be done?**

**Solution**

- Now generate a report
- Same as with SPDX output
- Contains rich set of elements
- License listings, copyright listings, ECC listing, Bulk phrase listing, ignored files listing, remarks listing
- Trying to summarise all information out of FOSSology for a component
# OSS Component Clearing report

<table>
<thead>
<tr>
<th>Clearing Information</th>
<th>FOSSology Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department</td>
<td>FOSSology Generation</td>
</tr>
<tr>
<td>Prepared by</td>
<td>2017/10/22 fossy</td>
</tr>
<tr>
<td>Reviewed by (opt.)</td>
<td>NA</td>
</tr>
<tr>
<td>Report release date</td>
<td>NA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component Information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Community</td>
<td>NA</td>
</tr>
<tr>
<td>Component</td>
<td>NA</td>
</tr>
<tr>
<td>Version</td>
<td>NA</td>
</tr>
<tr>
<td>Component hash (SHA-1)</td>
<td>6CD0FD95179595AF4D89D6F63C3782C3BD046651</td>
</tr>
<tr>
<td>Release date</td>
<td>NA</td>
</tr>
<tr>
<td>Main license(s)</td>
<td>Apache-1.1.</td>
</tr>
<tr>
<td>Other license(s)</td>
<td>License(s) Not Identified.</td>
</tr>
<tr>
<td>SW360 Portal Link</td>
<td>NA</td>
</tr>
<tr>
<td>Result of License Scan</td>
<td>Apache-1.1, IBM-possibility, No_license_found.</td>
</tr>
</tbody>
</table>

## 1. Assessment Summary

The following table only contains significant obligations, restrictions & risks for a quick overview – all obligations, restrictions & risks according to Section 3 n considered.
Feature: Obligations / Policies Management

SPDX Import allows for applying SPDX license analysis information to uploaded source code packages

<table>
<thead>
<tr>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>· List of licenses is good, but ...</td>
</tr>
<tr>
<td>· … who understands what to do in your organisation?</td>
</tr>
<tr>
<td>· Obligations / Policies explain what to do, but how to get them in?</td>
</tr>
<tr>
<td>· How can I have the involved obligations with the licenses?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>· Similar to licenses: obligations or policies</td>
</tr>
<tr>
<td>· Managed in the admin section</td>
</tr>
<tr>
<td>· But it is not so simple:</td>
</tr>
<tr>
<td>· How to deal with candidate licenses?</td>
</tr>
<tr>
<td>· How to deal with redundant obligations?</td>
</tr>
<tr>
<td>· How to update the database?</td>
</tr>
</tbody>
</table>
License Scanning or Source Code Scanning?

Why Or?
License Scanning and Source Code Scanning

*SPDX Import allows for applying SPDX license analysis information to uploaded source code packages*

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Solution</th>
</tr>
</thead>
</table>
| · Searching for licensing information by license statements does not find the source code origin.  
  · Was source code copied?  
  · How can I find the origin of Source code? | · Well, there was no solution  
  · FOSSology can search for licensing statements, but it does not have a database for source code.  
  · A source code database would be needed to match source code and determine its origin (and more)  
  · But FOSSology has a plug in architecture ... |
Fossology REST API
Feature: REST API – Basic functionality

After the release of version 3.4.0, the project has added a REST API to FOSSology

Current Endpoints:
- Uploads
- Folders
- Search
- Users
- Jobs
- Reports
- Tokens

https://www.fossology.org/get-started/basic-rest-api-calls/
Atarashi

A Step towards non-rule based standalone command line scanner…

(https://github.com/fossology/atarashi)
Conclusion
Summary

1. FOSSology for precise license analysis
2. FOSSology is a mature framework and Web application for license analysis
3. SPDX Import
4. Finally: review of SPDX documents and … reuse of licensing info at new versions
5. New Document Report
6. Beyond exchange of license information: Complete documentation of analysis
7. Obligations / Policies handling
8. Organise obligations with the found licenses.
9. Rest API
10. To get container running in the continuous build
Thank you very much - … some links:

- https://github.com/fossology/fossology
- https://www.fossology.org/

Try it Yourself:

$ docker run -p 8081:80 fossology/fossology