How to reduce on your OSS compliance work

Jul 18th, 2019
Lei Maohui, Fujitsu
leimaohui@cn.fujitsu.com
whoami

- Working for Fujitsu from 2011
- 7 years experience in Yocto related development

- In-House Embedded Linux Distributor of Fujitsu
- Our Distribution is used for
  - IVI
  - Server System Controller
  - Storage System
  - Network Equipment
  - Printer
  - etc.
Fujitsu’s contributions to Yocto community

- Data comes from yocto (2018-07-01 ~ 2019-07-01)

**Contributions in yocto**

- **Changesets**

<table>
<thead>
<tr>
<th>Layers</th>
<th>Changesets</th>
</tr>
</thead>
<tbody>
<tr>
<td>poky</td>
<td>46</td>
</tr>
<tr>
<td>oe-core</td>
<td>48</td>
</tr>
<tr>
<td>meta-oe</td>
<td>109</td>
</tr>
<tr>
<td>meta-cloud-services</td>
<td>50</td>
</tr>
<tr>
<td>bitbake</td>
<td>0</td>
</tr>
</tbody>
</table>

- Maintain meta-spdxscanner
Fujitsu’s contributions to Yocto community

Data comes from yocto (2018-07-01 ~ 2019-07-01)

Developers with the most changesets

<table>
<thead>
<tr>
<th>No.</th>
<th>Our Developer</th>
<th>Changesets</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Zang Ruochen</td>
<td>25 (0.6%)</td>
</tr>
<tr>
<td>40</td>
<td>Hong Liu</td>
<td>11 (0.3%)</td>
</tr>
<tr>
<td>71</td>
<td>Lei Maohui</td>
<td>7 (0.2%)</td>
</tr>
<tr>
<td>30</td>
<td>Zang Ruochen</td>
<td>25 (0.4%)</td>
</tr>
<tr>
<td>50</td>
<td>Hong Liu</td>
<td>12 (0.2%)</td>
</tr>
<tr>
<td>63</td>
<td>Lei Maohui</td>
<td>7 (0.1%)</td>
</tr>
<tr>
<td>8</td>
<td>Zang Ruochen</td>
<td>72 (3.7%)</td>
</tr>
<tr>
<td>17</td>
<td>Hong Liu</td>
<td>23 (1.2%)</td>
</tr>
<tr>
<td>29</td>
<td>Lei Maohui</td>
<td>11 (0.6%)</td>
</tr>
<tr>
<td>1</td>
<td>Zang Ruochen</td>
<td>26 (22.6%)</td>
</tr>
<tr>
<td>2</td>
<td>Hong Liu</td>
<td>20 (17.4%)</td>
</tr>
</tbody>
</table>
Agenda

**Yocto+SPDX**
- What is SPDX
- What is Yocto
- Overview of Yocto+SPDX

**Meta-spdxscanner**
- What is Meta-spdxscanner
- How to use
- What we have done for Meta-spdxscanner
- Features of fossdriver-host.bbclass

**Manage SPDX**
- Work with ClearlyDefined
- Work with OpenChain
- Manage SPDX files by dnf-plugin-tui
- Future work
Yocto+SPDX

- What is SPDX
- What is Yocto
- Overview of Yocto+SPDX
  - Spdx.bbclass
  - Meta-spdxscanner
**What is SPDX**

- The full name of SPDX is **Software Package Data Exchange**, which is a standard format for communicating the components, licenses and copyrights associated with a software package.

**Vision of SPDX**

- achieve license compliance with minimal cost across the supply chain.

---

**SPDX will be a good solution, if a SPDX implementation can generate SPDX file including license information automatically.**

*Obtain details from*
- [https://spdx.org/tools](https://spdx.org/tools)
What is SPDX (2/2)

Kernel v4.14 added one-liners come from SPDX

Play an important role
What is Yocto

The Yocto Project is an open source collaboration project that helps developers create custom Linux-based systems for embedded products.

https://www.yoctoproject.org/
Overview of Yocto+SPDX (1/2)

Poky (Core Layer)
- meta
- meta-poky
- meta-yocto
- meta-yocto-bsp
- ...

Only support fossology2
Not support SPDX2.0 spec
Complex process to set up environment

spdx.bbclass

Fossology2+SPDX

do_fetch → do_unpack → ... → do_spdx → ...
Overview of Yocto+SPDX (2/2)

Meta-spdxscanner

Poky(Core Layer)
- meta
- meta-poky
- meta-yocto
- meta-yocto-bsp
- ...

Meta-openembedded
- meta-oe
- meta-python

Meta-spdxscanner
- fossdriver-host.bbclass
- dosocs.bbclass
- dosocs-host.bbclass

do_fetch → do_unpack → ...... → do_spdx → ......
Meta-spdxscanner

- What is Meta-spdxscanner
- How to use
- What we have done for Meta-spdxscanner
- Features of fossdriver-host.bbclass
  - Reduce this layer's work time
  - Easier to use
  - More credible
What is Meta-spdxscanner

- **FOSS**
- Patches come from 3rd party

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Description</th>
<th>Type</th>
<th>Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta-spdxscanner</td>
<td>spdx support</td>
<td>Distribution</td>
<td><a href="https://github.com/di9pl/meta-spdxscanner">https://github.com/di9pl/meta-spdxscanner</a></td>
</tr>
</tbody>
</table>

SPDX files
How to use

1. **Start FOSSology Service**
   - `# docker pull fossology/fossology`
   - `# docker run -p 8081:80 fossology/fossology`

2. **install fossdriver**
   - `# pip install -e /WHEREVER/fossdriver`
   - `# Create and edit a config file. E.g. .fossdriverrc`

3. **Setup Yocto Build Environment**
   - `# cd [yocto_dir]
   - # source oe-init-build-env [build_dir]`

4. **Edit local.conf**
   - `# cd [build_dir]
   - # tail –n 2 conf/local.conf
     INHERIT += "fossdriver-host"
     SPDX_DEPLOY_DIR = "/yocto/spdx-outdir-warrior"

5. **Start building**
   - `# cd [build_dir]
   - # bitbake recipe/image`

Deploy

- `# ls /yocto/spdx-outdir-warrior`
- `acl-2.2.52.spdx libusb-compat-0.1.5.spdx python-manilaclient-1.4.0+gitAUTOINC+0bbd2144f7.spdx
  acpid-2.0.31.spdx libuser-0.62.spdx python-markupsafe-1.0.spdx adcli-0.8.2.spdx
  libutempter-1.1.6-alt2+gitAUTOINC+3ef74fff31.spdx python-mccabe-0.4.0.spdx ……`

SPDX Files
### History of SPDX create tools

<table>
<thead>
<tr>
<th>Item</th>
<th>FOSSology+ SPDX</th>
<th>dosocsv2</th>
<th>fossdriver</th>
<th>Fossology REST API</th>
</tr>
</thead>
<tbody>
<tr>
<td>maintainer</td>
<td>University of Nebraska</td>
<td>University of Nebraska</td>
<td>Swinslow of linuxfoundation</td>
<td>fossology</td>
</tr>
<tr>
<td>License</td>
<td>Apache-2.0</td>
<td>GPLv2</td>
<td>BSD-3-Clause OR MIT</td>
<td>GPL-2.0</td>
</tr>
<tr>
<td>Support SPDX version</td>
<td>1.2</td>
<td>2.0</td>
<td>2.1</td>
<td>2.1</td>
</tr>
<tr>
<td>Support fossology</td>
<td>Fossology 2</td>
<td>Fossology3</td>
<td>Fossology3 &lt;=3.5</td>
<td>Fossology3 &gt;=3.4.0</td>
</tr>
<tr>
<td>Support multitask</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Reuse prior results</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

**Now**
What we have done for Meta-spdxscanner(2/2)

- Use dosocsv2
- Support SPDX 2.0
- Work with fossology2

- Use fossdriver instead of dosocsv2

- Added DoSOCSv2-native.
- Easier to build environment

- Make do_spdx work without setup fossology2 server.

- Work with fossology3 (<= 3.5)

- Support REST API
- No depend 3rd party tools

- Work with latest fossology3 (>= 3.4.0)

Meta-spdxscanner (Published)
Features of fossdriver-host.bbclass (1/3)

Reduce this layer's work time

- Support multitask do_spdx
- Can reuse prior results

![Graph showing spend time (seconds) for different OSS components: ntp, busybox, openssl, openssh. The graph compares 'first' and 'reuse' scenarios.](image)
Features of fossdriver-host.bbclass (2/3)

Easier to use

Legal team

Build system 1

Fossology container

Data base

fossdriver

$ Bitbake core-core-image-minimal
0: glibc do SPDX - 79s  30%
1: openssl do SPDX - 32  40%

Build system 2

Fossology container

Data base

fossdriver

$ Bitbake core-core-image-minimal
glibc do SPDX ..........100%  12s
openssl do SPDX ..........100%  12s

Copy container
Features of fossdriver-host.bbclass (3/3)

<table>
<thead>
<tr>
<th>Files</th>
<th>Scanner Results (N: nomos, M: monk, Nk: ninka, I: reportImport)</th>
<th>Edited Results</th>
<th>Clearing Status</th>
<th>Files Cleared</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoconf</td>
<td>Autoconf-exception, GPL-2.0+, No_license_found, Public-domain, Public-domain-ref</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>doc</td>
<td>GPL-2.0, No_license_found, Public-domain, Sun-possibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>examples</td>
<td>BSD-2-Clause, BSD-2-Clause-FreeBSD, GPL-2.0, GPL-2.0+, No_license_found</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>include</td>
<td>GPL-2.0, GPL-2.0+, No_license_found, Zlib</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>platforms</td>
<td>CDDL, GPL, GPL-2.0, GPL-2.0+, No_license_found, See-file, UnclassifiedLicense</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>src</td>
<td>FreeBSD, AML, BSD, BSD-2-Clause, BSD-2-Clause-FreeBSD, GPL-2.0, GPL-2.0+, ISC, LGPL-2.0+, MIT, No_license_found, See-file, UnclassifiedLicense, WebM, Zlib</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPYING</td>
<td>GPL-2.0 [N][M: 88%]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChangeLog</td>
<td>No_license_found [N]</td>
<td></td>
<td></td>
<td>1/1</td>
<td>[View][INFO][Download] [Tag][Edit]</td>
</tr>
<tr>
<td>DISCLAIMER</td>
<td>No_license_found [N]</td>
<td></td>
<td></td>
<td>0/0</td>
<td>[View][INFO][Download] [Tag][Edit]</td>
</tr>
<tr>
<td>Developers</td>
<td>No_license_found [N]</td>
<td></td>
<td></td>
<td>0/0</td>
<td>[View][INFO][Download] [Tag][Edit]</td>
</tr>
<tr>
<td>INSTALL</td>
<td>No_license_found [N]</td>
<td></td>
<td></td>
<td>0/0</td>
<td>[View][INFO][Download] [Tag][Edit]</td>
</tr>
<tr>
<td>Makefile</td>
<td>No_license_found [N]</td>
<td></td>
<td></td>
<td>0/0</td>
<td>[View][INFO][Download] [Tag][Edit]</td>
</tr>
<tr>
<td>ReleaseNotes</td>
<td>No_license_found [N]</td>
<td></td>
<td></td>
<td>0/0</td>
<td>[View][INFO][Download] [Tag][Edit]</td>
</tr>
<tr>
<td>VERIFYING</td>
<td>No_license_found [N]</td>
<td></td>
<td></td>
<td>0/0</td>
<td>[View][INFO][Download] [Tag][Edit]</td>
</tr>
</tbody>
</table>

File Name: apcupsd-3.14.10/ChangeLog
SPDXID: SPDXRef-item98865
FileChecksum: SHA1: b828a5bc0b5da803da10dfetaf8ed3f6662154f1
FileChecksum: MD5: ed35c39d420dd53275d2c1439e9247b8
LicenseConcluded: AGPL-1.0 AND APSL-1.0
LicenseInfoInFile: NOASSERTION
FileCopyrightText: NOASSERTION

More credible
Manage SPDX

- Work with ClearlyDefined
- Work with OpenChain
- Manage SPDX files by dnf-plugin-tui
- Future work
Work with ClearlyDefined (1/2)

What is ClearlyDefined

About ClearlyDefined

ClearlyDefined and our parent organization, the Open Source Initiative, are on a mission to help FOSS projects thrive by being clearly defined. Lack of clarity around licenses and security vulnerabilities reduces engagement — that means fewer users, fewer contributors and a smaller community.

ClearlyDescribed

Knowing simple things like the source location for the open source component you are using enables contribution of docs, bug fixes, or new features. It also inspires confidence when doing IP and security code scans, and source code archiving and disclosure. Round that out with issue tracking site info, and you have a sound basis for engagement. Learn more...

ClearlyLicensed

Defining and knowing the license for an open source component is essential to a successful partnership. Communities choose a license with terms they like. ClearlyDefined helps clarify that choice and enables consumers to follow the terms by identifying key data such as license set, attribution parties, and code location. Learn more...

ClearlySecure

Teams working hard to create quality, secure free and open source components need a simple way of recording security issues they find and fix. Bug report and pull requests are great, CVEs and global notifications are even better. It can still be hard to relate that data to the components you use. ClearlyDefined gives communities a security forum that builds confidence and makes for even more collaboration. Learn more...

Can be get from spdx files

$ Less openssh-7.9p1.spdx
PackageName: openssh
PackageVersion: 7.9p1
PackageFileName: openssh-7.9p1-r0...
SPDXID: SPDXRef-upload265
PackageDownloadLocation: http://ftp.op...
PackageSummary: ......
......
FileName: spdx_temp.openssh-7.9p1/LICE...
SPDXID: SPDXRef-item671105
FileChecksum: SHA1: 6f569d09a2bd52b...
FileChecksum: MD5: 429658c6612f3a9b...
LicenseConcluded: NOASSERTION
LicenseInfoInFile: X11
LicenseInfoInFile: BSD-2-Clause
LicenseInfoInFile: MIT
LicenseInfoInFile: BSD-3-Clause
......
Work with ClearlyDefined (2/2)

openssh-portable

Described
Source: https://github.com/openssh/openssh-portable
Release: 2001-12-01
Facets: data, dev, doc, examples, tests

Tools
- clearlydefined/1.1.0
- scancode/2.9.2
- fossology/3.4.0

Curations
No curations found for this component

Licensed
Declared: SPDX license
Discovered: BSD-2-Clause, BSD-2-Clause AND BSD-4-Clause
Attributes: copyright 2000, Copyright (c) 2001, copyright...
Files: Total: 313, Licensed: 203 (65%), Attributed: 19

Files
- INSTALL
- LICENCE

Made in
SPDX license

Copyright 2019 FUJITSU COMPUTER TECHNOLOGIES LIMITED
The OpenChain Project makes open source licensing simple and consistent in the supply chain

The OpenChain Specification identifies the key requirements of a quality open source compliance program. OpenChain Conformance allows organizations to show they meet these requirements. The OpenChain Curriculum supports this process by providing extensive reference material for effective open source training and management. The result is that open source license compliance becomes more predictable, understandable and efficient for all participants in the software supply chain.

The OpenChain Project builds trust in open source by making open source license compliance simpler and more consistent.

Quality Open Source Compliance Defined
Work with OpenChain (2/3)

The Companies behind OpenChain

- Adobe
- ARM
- Bosch
- Cisco
- Comcast
- Facebook
- Fujitsu
- Google
- Hitachi
- Microsoft
- Panasonic
- Qualcomm
- Siemens
- Sony
- Toshiba
- Toyota
- Uber
- Western Digital
Work with OpenChain (3/3)

A diagram illustrates an example of an enterprise process. Key steps include:

- **Identify FOSS components for review**
- **Scan or audit source code**
- **Resolve any audit issues in line with company FOSS policies**
- **Record approved software version in inventory per product and per release**
- **Verify source packages for distribution and appropriate notices are provided**
- **Review SPDX files and Approvals FOSS software Components.**

The diagram shows the process flow from **Incoming Software** (Proprietary Software, 3rd Party Software, FOSS) to **Outgoing Software** (Notices & Attributions, Written Offer, SPDX files) and **Verification** stages.

An example of an enterprise process comes from OpenChain.

**Generate SPDX files by Source Code Scanner.**

**Discover and Resolve issues by using SPDX files.**

**Review SPDX files and Approvals FOSS software Components.**

**Archive Source Packages and SPDX files.**

Example of Compliance Management End-to-End Process

https://wiki.linuxfoundation.org/_media/openchain/openchain-curriculum-for-1-1.pdf
Manage SPDX files by dnf-plugin-tui (1/2)

What is dnf-plugin-tui

• A plugin of dnf that can manage packages created by YP on host.

Why we use dnf-plugin-tui

• Because rpm5 will be replaced by rpm4 from Yocto 2.3 due to the version change of python, Upstream (Yocto) pretends to use DNF from Yocto 2.3 as DNF is more suitable for rpm4.

Features of dnf-plugin-tui

• Text-based user interface for dnf
• Manage SPDX files
• Manage SRPM file

https://github.com/ubinux/dnf-plugin-tui
Manage SPDX files by dnf-plugin-tui(2/2)

Build

Packages

SRPM packages

SPDX files

Package Installer

lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Select your operation tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq

x

x Install --->

x Remove --->

x Upgrade --->

x Create binary package archive --->

x Create source archive --->

x Create SPDX archive --->

x Create archive(rpm, src.rpm and spdx files) --->

x Make filesystem image --->

x

Copyright 2019 FUJITSU COMPUTER TECHNOLOGIES LIMITED
Future work

- Maintain meta-spdxscanner
  - Solve issues
  - Meet more user case

- Added fossology REST API support
  - There will be no necessary to install tools come from 3rd party.
  - Maintained by fosslogy, can get long-term stable technical support.
Any Questions?
FUJITSU

shaping tomorrow with you