Bitbaking SPDX SBoM

Michael Opdenacker, Bootlin

Yocto Project Summit, 2022.11

About

- **Michael Opdenacker**
  - Founder, embedded Linux engineer and trainer at Bootlin
- **Bootlin is a contributor to the Yocto Project**
  - Yocto Project documentation maintenance
  - Yocto Project SWAT Team, keeping track of all the issues encountered by the autobuilders
  - Yocto Project and OpenEmbedded development course with free (as in free speech and free beer) materials.

Introduction and definitions

SBoM and SPDX

SBoM = Software Bill of Materials

A description of all the components in a software release, including:

- Sources
- Licenses
- Dependencies
- Applied changes
- Fixes for known vulnerabilities

https://en.wikipedia.org/wiki/Software_supply_chain

SPDX = Software Package Data Exchange

An open standard for describing an SBoM

- A Linux Foundation project
- Originally created for license compliance, first version in 2011
- An ISO standard since August 2021 (version 2.2.1)
- Can be described in various human readable formats, currently YAML 1.2, JSON, RDF/XML, *tag:value* flat text file, .xls spreadsheet.

https://en.wikipedia.org/wiki/Software_Package_Data_Exchange

Why does it matter?

- **SBoM information is essential for vulnerability and license compliance assessment.**
- **The US government is pushing for having such information in all software it procures and will probably make it mandatory soon.**
- **The Yocto Project is a pioneer in this area**
  - Cannot generate SPDX SBoM with Buildroot yet 😉

How to generate SPDX SBoM

for your images

SPDX SBoM support in Yocto Project

- Yocto Project can generate JSON SPDX for your images since version 3.4 ("Honister", October 2021), from the metadata in the recipes.
- Implemented by Joshua Watt
  in meta/classes/create-spdx.bbclass
- But not documented in the YP manuals until Nov. 2022
- This presentation shares the findings from this documentation work.

create-spdx class: how to use

- Add this to a configuration file (conf/local.conf)

```
INHERIT += "create-spdx"
```

- And generate your image as usual

```
bitbake core-image-minimal
```

create-spdx class: optional variables

- SPDX_PRETTY
  Make generated files more human readable (newlines, indentation)
- SPDX_ARCHIVE_PACKAGED
  Add compressed archives of the files in the generated target packages.
- SPDX_INCLUDE_SOURCES
  Add descriptions of the source files for host tools and target packages.
- SPDX_ARCHIVE_SOURCES
  Add archives of these source files themselves.
  Only works when SPDX_INCLUDE_SOURCES is set.

create-spdx class: how to set optional variables

- **Add them to a configuration file** (conf/local.conf)

```
SPDX_PRETTY = "1"
SPDX_ARCHIVE_PACKAGED = "1"
SPDX_INCLUDE_SOURCES = "1"
SPDX_ARCHIVE_SOURCES = "1"
```

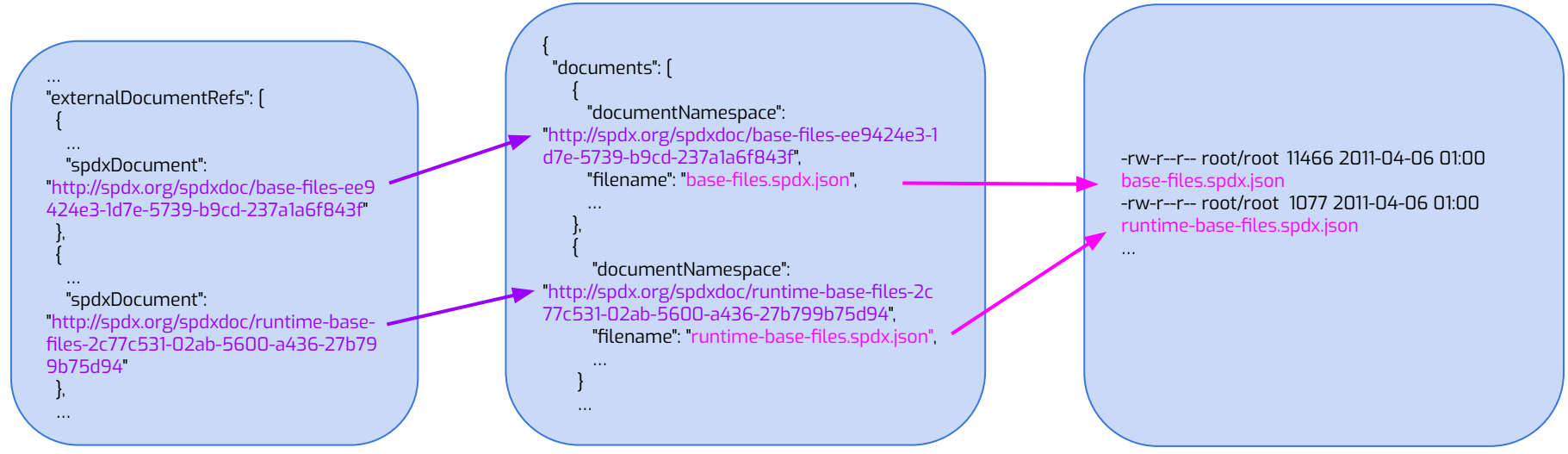
create-spdx class output (1)

- **Output SPDX in** tmp/deploy/images/MACHINE/:
  - IMAGE-MACHINE.spdx.json:
    Toplevel output
  - IMAGE-MACHINE.spdx.index.json:
    Index of JSON SPDX files for individual host and target recipes
  - IMAGE-MACHINE.spdx.tar.zst:
    Compressed archive containing all such files

IMAGE-MACHINE.spdx.json example


```
{
  "SPDXID": "SPDXRef-DOCUMENT",
  "creationInfo": {
        "comment": "This document was created by analyzing the source of the Yocto recipe during the build.",
        "created": "2022-10-25T12:32:13Z",
        "creators": [
        "Tool: OpenEmbedded Core create-spdx.bbclass",
        "Organization: OpenEmbedded ()",
        "Person: N/A ()"
        ],
        "licenseListVersion": "3.14"
  },
  "dataLicense": "CC0-1.0",
  "documentNamespace": "http://spdx.org/spdxdoc/core-image-minimal-qemux86-64-20221025122556-f686f4f3-b1af-5a74-ac94-7b96ecc4d75a",
  "externalDocumentRefs": [
        {
        "checksum": {
        "algorithm": "SHA1",
        "checksumValue": "f6de08ea7fa026f480fd80cf7862a5c99c4d7a2b"
        },
        "externalDocumentId": "DocumentRef-base-files",
        "spdxDocument": "http://spdx.org/spdxdoc/base-files-ee9424e3-1d7e-5739-b9cd-237a1a6f843f"
        },
...
```

create-spdx class output (2)



create-spdx class output (3)

- **Ancillary generated files in** tmp/deploy/spdx/MACHINE:
  - **Individual JSON files included in** IMAGE-MACHINE.spdx.tar.zst
  - **Compressed archives of the files in the generated target packages, in** packages/packagename.tar.zst **(when** SPDX_ARCHIVE_PACKAGED **is set).**
  - **Compressed archives of the source files used to build host tools and target packages in** recipes/recipe-packagename.tar.zst **(when** SPDX_ARCHIVE_SOURCES **is set).**

Going further

- Tools to validate and consume SPDX output?
  Would love to be able to browse our output files.
  See https://spdx.dev/resources/tools/
- Validation tools fail with warnings on Yocto Project's output
  https://lists.openembedded.org/g/openembedded-core/message/173723
- Upcoming 3.0 version of the standard
  (Joshua Watt contributing to this effort)

References

- New section in the Yocto Project Manual:
  https://docs.yoctoproject.org/dev-manual/common-tasks.html#creating-a-software-bill-of-materials
- Joshua Watt's upcoming presentation at this Yocto Project Summit:
  SBoMs and Supply Chain with the Yocto Project
  https://summit.yoctoproject.org/yocto-project-summit-2022-11/talk/QFTUWN/
- Kate Stewart's presentation at ELCE 2022:
  SBOMs: Essential for Embedded too!
  https://elinux.org/images/3/3e/Stewart-sboms-elce2022.pdf
- SPDX standard specifications, useful reference:
  https://spdx.dev/specifications/

¿Questions?

yocto PROJECT | THE LINUX FOUNDATION