

# Sharing and reusing OSS compliance information: The Open Source Curation Database

## How to use the database?

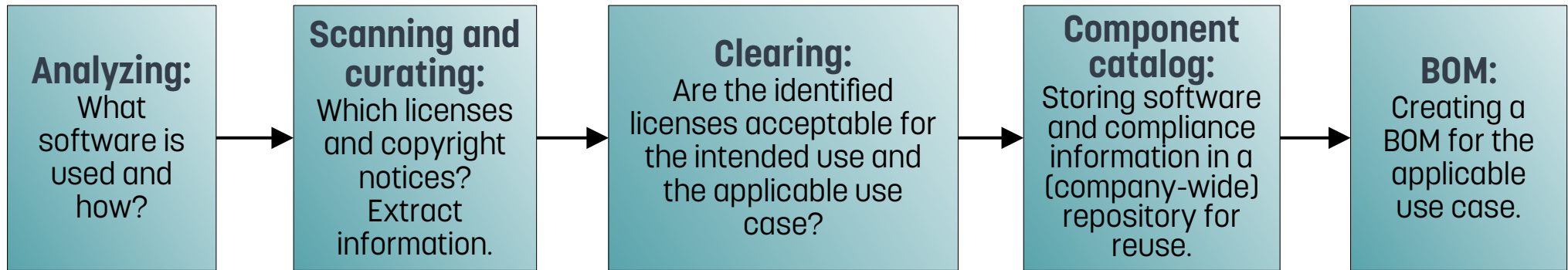
Caren Kresse

Open Source Automation Development Lab (OSADL) eG

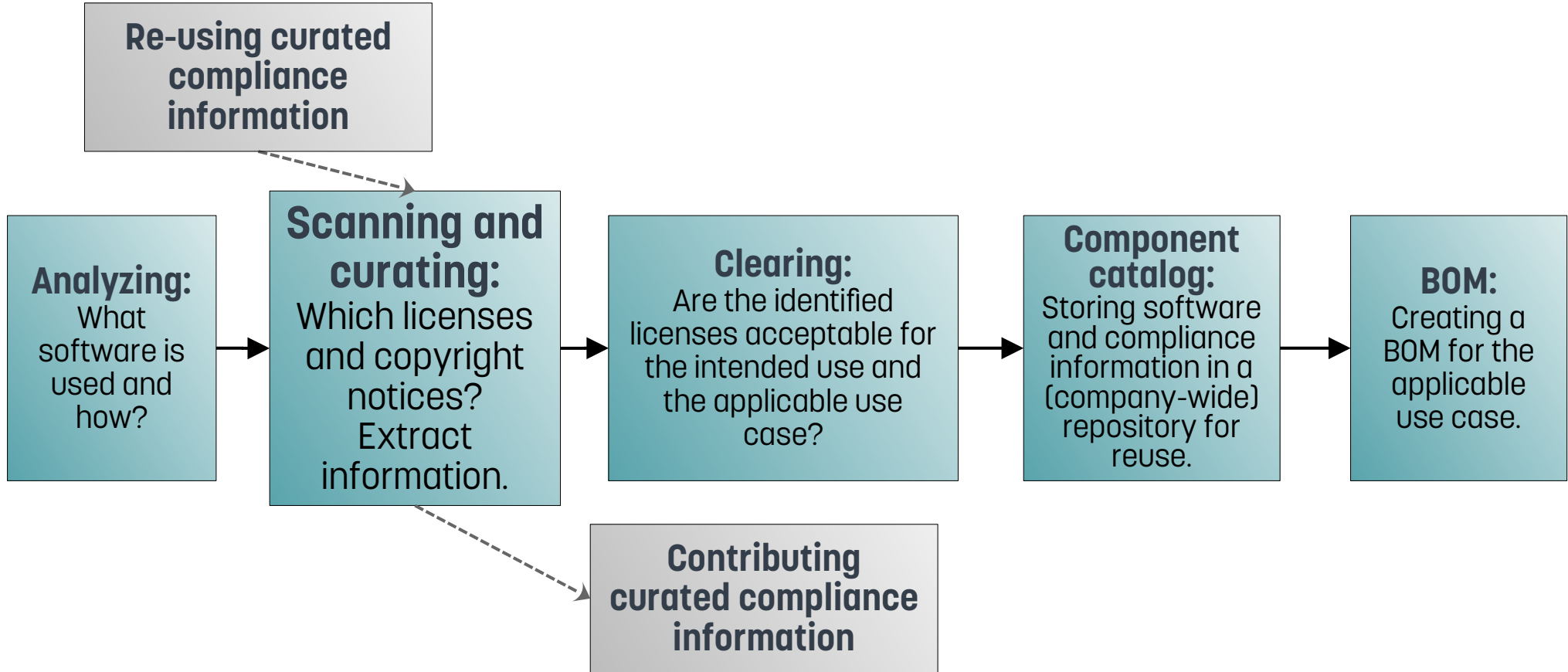
# Why re-using compliance data?

- The success of FOSS is partly due to the fact that development resources and efforts are reduced by **re-using existing software components**.
- Some of the reduction is negatively compensated by the effort required to scan and clear FOSS for license compliant use.
- There are many FOSS components that are deployed unmodified by a large number of users.
- Why not **share compliance tasks** in the same way as the software development?

# Open Source compliance toolchain



# Re-using compliance information



# The OSADL project: A Curation Database

<https://github.com/Open-Source-Compliance/package-analysis>

- Contains **license and copyright analysis results** for various packages:
  - Readme with metadata of the package, *e.g.* download location, reviews, comments
  - SPDX tag:value file with concluded licenses, copyright notices and comments on decisions
  - Disclosure document with aggregated license texts, copyright notices and acknowledgments

# The OSADL project: A Curation Database

<https://github.com/Open-Source-Compliance/package-analysis>

- Contains **license and copyright analysis results** for various packages:
  - Readme with metadata of the package, *e.g.* download location, reviews, comments
  - **SPDX tag:value file with concluded licenses, copyright notices and comments on decisions**
  - Disclosure document with aggregated license texts, copyright notices and acknowledgments

# SPDX tag:value report – Creation information

```
##-----
```

```
## Creation Information
```

```
##-----
```

```
Creator: Tool: spdx2
```

```
Creator: Person: Oliver Fendt
```

```
CreatorComment: <text>
```

```
This document was created using license information and a generator  
from Fossology.
```

```
It contains the license and copyright analysis of OpenSSL 3.0.5
```

```
Please check "LicenseComments" for explanations of concluded licenses
```

```
</text>
```

```
Created: 2022-07-06T14:58:22Z
```

```
LicenseListVersion: 2.6
```

# SPDX tag:value report – Package information

```
##-----  
## Package Information  
##-----
```

**PackageName:** openssl-openssl-3.0.5.tar.gz

[...]

**PackageChecksum:** SHA1: edc3465a8a43ce580268e726b6f7b827f4a6261e

**PackageChecksum:** SHA256: b6363cf1bca88f0a46a768883a225e644135432d6a51ab1c4660ab58af541078

**PackageChecksum:** MD5: 22733b9187548b735201fd9f7aa12e71

**PackageLicenseConcluded:** NOASSERTION

**PackageLicenseDeclared:** LicenseRef-Apache-2.0

**PackageLicenseComments:** <text> licenseInfoInFile determined by Scanners:

- nomos ("4.1.0.28".bb8a6d)
- monk ("4.1.0.28".bb8a6d)
- ojo ("4.1.0.28".bb8a6d)
- scancode ("4.1.0.28".bb8a6d)</text>

[...]

Only if there is a main  
license for the package  
(LICENSE/COPYING file  
in root directory)



# SPDX tag:value report – Package information

```
##-----  
## Package Information  
##-----
```

PackageName: openssl-openssl-3.0.5.tar.gz

[...]

PackageChecksum: SHA1: edc3465a8a43ce580268e726b6f7b827

PackageChecksum: SHA256: b6363cf1bca88f0a46a768883a225e

PackageChecksum: MD5: 22733b9187548b735201fd9f7aa12e71

PackageLicenseConcluded: NOASSERTION

PackageLicenseDeclared: LicenseRef-Apache-2.0

PackageLicenseComments: <text> licenseInfoInFile determined by Scanners:

- nomos ("4.1.0.28".bb8a6d)
- monk ("4.1.0.28".bb8a6d)
- ojo ("4.1.0.28".bb8a6d)
- scancode ("4.1.0.28".bb8a6d)</text>

[...]

Only if there is a main license for the package (LICENSE/COPYING file in root directory)

**! This license is not automatically valid for all files and there is usually more than this one license in a package.**

# SPDX tag:value report – File information

##File

Final license decision

FileName: openssl-3.0.5.tar.gz/openssl-3.0.5.tar/openssl-openssl-3.0.5/crypto/LPdir\_wince.c

SPDXID: SPDXRef-item158856105

FileChecksum: SHA1: dc3e4bb9f2cf76426da9ad5dbc8ad4a2356c3359

FileChecksum: SHA256: fd878a5b569cd41d63ba673420a4d95adf9ad3048ea0fb4854504ba55572d8

FileChecksum: MD5: 62fba2db5fb486d537d869af119135b

LicenseConcluded: LicenseRef-Apache-2.0 OR LicenseRef-BSD-2-Clause-3185f2587757a9c63eaa83143f7c0386

LicenseComments: <text>The information in the file is:

Besides the Apache-2.0 header the following information is in the file:

This file is dual-licensed and is also available under the following terms:

Followed by the BSD-2-clause license text. Thus dual licensing was concluded. </text>

LicenseInfoInFile: LicenseRef-Apache-2.0

LicenseInfoInFile: LicenseRef-OpenSSL

LicenseInfoInFile: LicenseRef-Dual-license

LicenseInfoInFile: LicenseRef-BSD-2-Clause\_REGENTS-AND-CONTRIBUTORS

FileCopyrightText: <text> Copyright 2004-2016 The OpenSSL Project Authors.

Copyright (c) 2004, Richard Levitte <richard@levitte.org></text>

If not obvious,  
explanation of license  
decision

Scanner findings

# SPDX tag:value report – License information

```
##-----  
## License Information  
##-----
```

```
LicenseID: LicenseRef-Apache-2.0  
LicenseName: Apache License 2.0  
ExtractedText: <text> Apache License  
Version 2.0, January 2004  
http://www.apache.org/licenses/
```

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

[...]

# SPDX tag:value report – License information

LicenseID: LicenseRef-BSD-2-Clause-3185f2587757a9c63eaa83143f7c0386

LicenseName: BSD-2-Clause-3185f2587757a9c63eaa83143f7c0386

ExtractedText: <text> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. </text>

[...]

# Example: Re-using curated data to clear OpenSSL

1. Download the curated data of OpenSSL v3.0.5 from the project page and upload the source files into FOSSology.

OliverFendt Update openssl-3.0.5-SPDX2TV.spdx

ef73c4e on Sep 5 History

..

README.md	adding OpenSSL 3.0.5	4 months ago
openssl-3.0.5-OSS-disclosure.txt	adding OpenSSL 3.0.5	4 months ago
openssl-3.0.5-SPDX2TV.spdx	Update openssl-3.0.5-SPDX2TV.spdx	2 months ago

README.md

## Download Location

<https://github.com/openssl/openssl/archive/refs/tags/openssl-3.0.5.tar.gz>

## Reviewers

The information was reviewed by:

- add reviewer here

## Upload from URL

Version: [4.1.0.95], Branch: [master], Commit: [#82b3b2] 2022/09/27 09:21 +02:00 built @ 2022/09/28 16:48 +02:00

To manage your own group permissions go into **Admin > Groups > Manage Group Users**. To manage permissions for this one upload, go to **Admin > Upload Permissions**.

This option permits uploading a single file (which may be iso, tar, rpm, jar, zip, bz2, msi, cab, etc.) or a directory from a remote web or FTP server to FOSSology. The file or directory to upload must be accessible via a URL and must not require human interaction such as login credentials.

1. Select the folder for storing the uploaded files:

Use cases ▾

2. Enter the URL to the file or directory:

3. (Optional) Enter a viewable name for this file or directory:

Note: If no name is provided, then the uploaded file (directory) name will be used.

4. (Optional) Enter comma-separated lists of file name suffixes or patterns to accept:

5. (Optional) Enter comma-separated lists of file name suffixes or patterns to reject:

6. (Optional) maximum recursion depth (inf or 0 for infinite):

7. (Optional) Enter a description of this file:

8.  Apply global decisions for current upload ⓘ

9.  Ignore SCM files (Git, SVN, TFS) and files with particular Mimetype ⓘ

10.  Visible only for active group ⓘ

Visible for all groups ⓘ

Make Public ⓘ

# cont'd:

## 11. Select optional analysis:

- Bucket Analysis
- Copyright/Email/URL/Author Analysis
- ECC Analysis, scanning for text fragments potentially relevant for export control
- Keyword Analysis
- MIME-type Analysis (Determine mimetype of every file. Not needed for licenses or buckets)
- Monk License Analysis, scanning for licenses performing a text comparison
- Nomos License Analysis, scanning for licenses using regular expressions
- Ojo License Analysis, scanning for licenses using SPDX-License-Identifier
- Package Analysis (Parse package headers)
- REUSE.Software Analysis (forces \*Ojo License Analysis\*)
- Software Heritage Analysis

## 12. Automatic Concluded License Decider ⓘ, based on

- Scanners matches if all Nomos findings are within the Monk findings
- Scanners matches if Ojo or REUSE.Software findings are no contradiction with other findings
- Bulk phrases from reused packages
- New scanner results, i.e., decisions were marked as work in progress if new scanner finds additional licenses

## 13. (Optional) Reuse ⓘ

- Select an already uploaded package for reuse in specific folder Software Repository (cemde:3) ▾
- Enhanced reuse (slower) ⓘ
- Reuse main license/s ⓘ
- Reuse report configuration settings ⓘ
- Reuse deactivated copyrights ⓘ

Upload to reuse:

## 14. ScanCode Toolkit ⓘ, scan for

- License ⓘ
- Copyright ⓘ
- Email ⓘ
- URL ⓘ

Upload



# Example: Re-using curated data to clear OpenSSL

1. Download the curated data of OpenSSL v3.0.5 from the project page and upload the source files into FOSSology.
2. Convert SPDX tag:value file into SPDX RDF format (using SPDX tools: <https://github.com/spdx/tools-java>) and import the SPDX RDF file into FOSSology to clear the package (**3303 of 3318** files are automatically cleared!).

## Report Import

Version: [4.1.0.95], Branch: [master], Commit: [#82b3b2] 2022/09/27 09:21 +02:00 built @ 2022/09/28 16:48 +02:00

1. Select the folder that contains the upload:

2. Select the upload you wish to edit:

3. Select report to upload:  openssl-3.0.5-SPDX2RDF.spdx.rdf

4. Select how the information should be imported:

- Create new licenses as
  - license candidate
  - 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

# Change concluded License

Version: [4.1.0.95], Branch: [master], Commit: [#82b3b2] 2022/09/27 09:21 +02:00 built @ 2022/09/28 16:48 +02:00

Folder: [Software Repository/ ckresse/ Use cases/ openssl-3.0.5.tar.gz/openssl-3.0.5/openssl-openssl-3.0.5/apps/asn1parse.c](#)

[Copyright/Email/Url/Author](#) | [ECC](#) | [keyword](#) | [Bucket](#) | [Spasht](#) • [Hex](#) | [Text](#) | [Formatted](#) • [Refresh](#)

Cleared: 3303/3318

Hide Legend

```
/*
 * Copyright 1995-2021 The OpenSSL Project Authors. All Rights Reserved.
 *
 * Licensed under the Apache License 2.0 (the "License"). You may not use
 * this file except in compliance with the License. You can obtain a copy
 * in the file LICENSE in the source distribution or at
 * https://www.openssl.org/source/license.html
 */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "apps.h"
#include "progs.h"
#include <openssl/err.h>
#include <openssl/evp.h>
#include <openssl/x509.h>
#include <openssl/pem.h>
#include <openssl/asn1t.h>
```

```
typedef enum OPTION_choice {
OPT_COMMON,
OPT_INFORM, OPT_IN, OPT_OUT, OPT_INDENT, OPT_NOOUT,
OPT_OID, OPT_OFFSET, OPT_LENGTH, OPT_DUMP, OPT_DLIMIT,
OPT_STRPARSE, OPT_GENSTR, OPT_GENCONF, OPT_STRICTPEM,
OPT_ITEM
} OPTION_CHOICE;
```

```
const OPTIONS asn1parse_options[] = {
OPT_SECTION("General"),
{"help", OPT_HELP, ':', "Display this summary"},
{"oid", OPT_OID, '<', "file of extra oid definitions"},
```

**Legend:**  
license relevant text

les ⓘ  
les with licenses ⓘ  
les with licenses and no clearing result ⓘ

## Clearing decision scope

Apply decision to all future occurrences of this file ⓘ

## Clearing decision type

- No license known ⓘ
- To be discussed ⓘ
- Irrelevant ⓘ
- Identified ⓘ
- Do not use ⓘ
- Non functional ⓘ

Action ⓘ ▲	License ⓘ ⬆	Source ⓘ	License Text ⓘ	Acknowledgement ⓘ	Comment ⓘ
✖ ⭐	Apache-2.0	nomos: #1 reportImport Imported decision	Click to add	Click to add	Click to add
+ ⭐	OpenSSL	nomos: #1 scancode: #1 (97 %)	-	-	-

Showing 1 to 2 of 2 entries

# Example: Re-using curated data to clear OpenSSL

1. Download the curated data of OpenSSL v3.0.5 from the project page and upload the source files into FOSSology.
2. Convert SPDX tag:value file into SPDX RDF format (using SPDX tools: <https://github.com/spdx/tools-java>) and import the SPDX RDF file into FOSSology to clear the package (**3303** of **3318** files are automatically cleared!).
3. The information can also be re-used for different versions, *e.g.* v3.0.7. To do so, upload the new source files into FOSSology and choose "Reuse" referencing the already cleared package.

## Upload from URL

Version: [4.1.0.95], Branch: [master], Commit: [#82b3b2] 2022/09/27 09:21 +02:00 built @ 2022/09/28 16:48 +02:00

To manage your own group permissions go into **Admin > Groups > Manage Group Users**. To manage permissions for this one upload, go to **Admin > Upload Permissions**.

This option permits uploading a single file (which may be iso, tar, rpm, jar, zip, bz2, msi, cab, etc.) or a directory from a remote web or FTP server to FOSSology. The file or directory to upload must be accessible via a URL and must not require human interaction such as login credentials.

1. Select the folder for storing the uploaded files:

Use cases ▾

2. Enter the URL to the file or directory:

3. (Optional) Enter a viewable name for this file or directory:

Note: If no name is provided, then the uploaded file (directory) name will be used.

4. (Optional) Enter comma-separated lists of file name suffixes or patterns to accept:

5. (Optional) Enter comma-separated lists of file name suffixes or patterns to reject:

6. (Optional) maximum recursion depth (inf or 0 for infinite):

7. (Optional) Enter a description of this file:

8.  Apply global decisions for current upload [?](#)

9.  Ignore SCM files (Git, SVN, TFS) and files with particular Mimetype [?](#)

10.  Visible only for active group [?](#)

Visible for all groups [?](#)

Make Public [?](#)

# cont'd:

## 11. Select optional analysis:

- Bucket Analysis
- Copyright/Email/URL/Author Analysis
- ECC Analysis, scanning for text fragments potentially relevant for export control
- Keyword Analysis
- MIME-type Analysis (Determine mimetype of every file. Not needed for licenses or buckets)
- Monk License Analysis, scanning for licenses performing a text comparison
- Nomos License Analysis, scanning for licenses using regular expressions
- Ojo License Analysis, scanning for licenses using SPDX-License-Identifier
- Package Analysis (Parse package headers)
- REUSE.Software Analysis (forces \*Ojo License Analysis\*)
- Software Heritage Analysis

## 12. Automatic Concluded License Decider ⓘ, based on

- Scanners matches if all Nomos findings are within the Monk findings
- Scanners matches if Ojo or REUSE.Software findings are no contradiction with other findings
- Bulk phrases from reused packages
- New scanner results, i.e., decisions were marked as work in progress if new scanner finds additional licenses

## 13. (Optional) Reuse ⓘ

- Select an already uploaded package for reuse in specific folder
- Enhanced reuse (slower) ⓘ
- Reuse main license/s ⓘ
- Reuse report configuration settings ⓘ
- Reuse deactivated copyrights ⓘ

Upload to reuse:

## 14. ScanCode Toolkit ⓘ, scan for

- License ⓘ
- Copyright ⓘ
- Email ⓘ
- URL ⓘ

Upload

# Example: Re-using curated data to clear OpenSSL

1. Download the curated data of OpenSSL v3.0.5 from the project page and upload the source files into FOSSology.
2. Convert SPDX tag:value file into SPDX RDF format (using SPDX tools: <https://github.com/spdx/tools-java>) and import the SPDX RDF file into FOSSology to clear the package (**3303 of 3318** files are automatically cleared).
3. The information can also be re-used for different versions, *e.g.* v3.0.7. To do so, upload the new source files into FOSSology and choose “Reuse” referencing the already cleared package.
4. Only **227 of 3329** files must be cleared individually (< 7 %), 170 of which can be cleared with one bulk scan.

# License Browser

Version: [4.1.0.95], Branch: [master], Commit: [#82b3b2] 2022/09/27 09:21 +02:00 built @ 2022/09/28 16:48 +02:00

Folder: [Software Repository/ ckresse/ Use cases/ openssl-3.0.7.tar.gz](#)

[Software Heritage](#) | [License Browser](#) | [File Browser](#) | [Spasht](#) | [Copyright](#) | [ECC](#) | [Email/URL/Author](#) | [Keyword](#)

Display  files (tree view or flat)

	-- filter for scan results --	-- filter for edited results --	<input type="checkbox"/> Open	Cleared / Open / Total	<input type="checkbox"/> Clear
<b>Files</b>	<b>Scanner Results (N: nomos, M: monk, Nk: ninka, I: reportImport, O: ojo, S: scancode, Sp: spasht, Rs: reso)</b>	<b>Edited Results</b>	<b>Clear Status</b>		<b>Decisions</b>
<a href="#">openssl-3.0.7.tar/openssl-openssl-3.0.7</a>	Apache-2.0, Artistic-1.0, Artistic-1.0-Perl, BSD-2-Clause, BSD-3-Clause, BSD-Source-Code, CC0-1.0, Cryptogams, Dual-license, GPL, GPL-1.0, GPL-1.0+, GPL-2.0, GPL-2.0+, LGPL-2.1+, LicenseRef-scancode-generic-cla, LicenseRef-scancode-public-domain, LicenseRef-scancode-public-domain-disclaimer, LicenseRef-scancode-unknown-license-reference, MIT, MPL-1.1, No_license_found, OpenSSL, Perl-possibility, Public-domain, RSA-possibility, See-doc.OTHER, See-file	GPL-1.0+, Apache-2.0, BSD-2-Clause, BSD-3-Clause, CC0-1.0, Cryptogams, Dual-license, Public-domain, Artistic-1.0-Perl, License-of-GNU-Licenses	<input checked="" type="checkbox"/>	3102 / 3329 / 4565	<input type="checkbox"/> [Bulk] <input type="checkbox"/>

Showing 1 to 1 of 1 files

Page 1 of 1



# Automated use of curation data

- **Integrating** curated data into a **build process**.
- For every source file that is **actually compiled into a binary**, the compliance information is extracted from the SPDX tag:value file **via checksum**.
- This information is combined and now contains only those licenses that must be considered for distribution of the binary.
- In addition, all **files without a matching checksum** are listed.
- These must be **checked individually**.

# Example: BusyBox

1. List all compiled source files:

a) Use compiler flags

```
# gcc -MMD file.list [TARGET]
```

or

b) Build binary with debug information and extract source files from symbol table

```
# gdb -ex "info sources" [BINARY] >file.list
```

# Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble:

```
##File
FileName: busybox-1.35.0.tar.bz2/busybox-1.35.0.tar/busybox-1.35.0/coreutils/timeout.c
SPDXID: SPDXRef-item160248981
FileChecksum: SHA1: ad066d3cdf9f0c525cd9857d4b3cb26cd49dc6c2
[...]
LicenseConcluded: LicenseRef-GPL-2.0
[...]
FileCopyrightText: <text> Copyright (C) 2005-6, Roberto A. Foglietta
<me@roberto.foglietta.name> </text>
```

# Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble:

```
if sed -n "/[FileName]/,/#File/p" [SPDX] | grep -q $checksum
then
    sed -n "/[FileName]/,/#File/p" [SPDX] >>assembled.spdx
fi
```

# Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.

```
if sed -n "/[FileName]/,##File/p" [SPDX] | grep -q $checksum
then
    sed -n "/[FileName]/,##File/p" [SPDX] >>assembled.spdx
```

## 4. List changed files:

```
else
    echo [FileName] >>changed-files.list
fi
```

# Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.
4. List changed files.
5. Get unique `ConcludedLicenses` and add respective license texts to *assembled.spdx*:

```
LicenseID: LicenseRef-GPL-2.0  
LicenseName: GNU General Public License v2.0 only  
ExtractedText: <text> GNU General Public License, version 2  
[...]
```

# Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.
4. List changed files.
5. Get unique ConcludedLicenses and add respective license texts to *assembled.spdx*:

```
for i in ${grep LicenseConcluded assembled.spdx | cut -d' ' -f 2- | sort -u}
do
  echo $i >>licenses.list
  sed -n "\&LicenseID: $i$&,\&LicenseID:&p" [SPDX] >>assembled.spdx
done
```

# Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.
4. List changed files.
5. Get unique ConcludedLicenses and add respective license texts to *assembled.spdx*.

➔ **Should of course be scripted and automated!**



# Example: BusyBox – result

- List of **all source files that were compiled into the binary.**
- List of **all source files that were changed** w.r.t. to the curated data.
- List of **all licenses of compiled files** that are present in the curated data.
- **SPDX tag:value file of compiled files** that are present in the curated data including concluded licenses, copyright notices and license texts.

# Conclusion (1)

- FOSS compliance must be **integrated** into the development workflow.
- A large number of tools is available, but must be chosen for their respective strengths: **There is no one-in-all tool!**
- **Not everything can be automated!** Human expertise and some manual labor are always required.
- **Re-using curated compliance information** reduces the required effort significantly.

# Conclusion (2)

- OSADL is launching a project to provide a publicly available **trusted database** with **curated compliance information** for frequently used FOSS components.
- The curated data can help to automate the **clearing of FOSS to license it for use in industrial products**.
- Some manual input and review is still indispensable, but:

**The reuse of curated licensing and copyright information can tremendously reduce the time required to clear a software package!**



[osselot.org](https://osselot.org)



- <https://www.osselot.org> is the official project page that will contain news, explanations, example use cases, tools and to dos.
- The curated material will still be available via GitHub <https://github.com/Open-Source-Compliance/package-analysis>

**Questions, requests and contributions  
are very welcome!**