
ScanCode-Toolkit

Sep 04, 2019

Contents

1	Guide	1
1.1	License	1
1.2	Help	1
1.3	Document Maintenance	2
2	Tutorial Documents	5
3	How-To Documents	7
4	Reference Documents	9
5	Discussion Documents	11
6	Indices and Tables	13

1.1 License

ScanCode Toolkit includes documents that are dedicated to the Public Domain using the Creative Commons CC0 1.0 Universal (CC0 1.0) Public Domain Dedication: <http://creativecommons.org/publicdomain/zero/1.0/>

1.2 Help

ScanCode Toolkit is part of the AboutCode family of open source projects.

1.2.1 AboutCode Projects

- **[ScanCode Toolkit](<https://github.com/nexB/scancode-toolkit>)**: This is a set of code scanning tools to detect the origin and license of code and dependencies. ScanCode Toolkit uses a plug-in architecture to run a series of scan-related tools in one process flow. This is the most popular project and is used by hundreds of software teams. <https://github.com/nexB/scancode-toolkit> . The lead maintainer is @pombredanne
- **[Scancode Workbench](<https://github.com/nexB/scancode-workbench>)** (formerly AboutCode Manager) This is a desktop application (based on Electron) to review the results of a scan and document your conclusions about the origin and license of software components and packages. <https://github.com/nexB/aboutcode-manager> . The lead maintainer is @majurg
- **[AboutCode Toolkit](<https://github.com/nexB/aboutcode-toolkit>)**: This is a set of command line tools to document the provenance of your code and generate attribution notices. AboutCode Toolkit uses small yaml files to document code provenance inside a codebase. <https://github.com/nexB/aboutcode-toolkit> . The lead maintainer is @chinyeungli
- **[TraceCode Toolkit](<https://github.com/nexB/tracecode-toolkit>)**: This is a set of tools to trace files from your deployment or distribution packages back to their origin in a development codebase or repository. The primary tool uses strace <https://github.com/strace/strace/> to trace system calls on Linux and construct a build

graph from syscalls to show which files are used to build a binary. We are contributors to strace. Maintained by @pombredanne

- **[Conan]**(<https://github.com/nexB/conan>): “conan” stands for “CONtainer ANalysis” and is a tool to analyze the structure and provenance of software components in Docker images using static analysis. <https://github.com/nexB/conan> Maintained by @pombredanne
- **[license-expression]**(<https://github.com/nexB/license-expression/>): This is a library to parse, analyze, compare and normalize SPDX-like license expressions using a boolean logic expression engine. See <https://spdx.org/spdx-specification-21-web-version#h.jxpfx0ykyb60> to understand what a license expression is. See <https://github.com/nexB/license-expression> for the code. The underlying boolean engine is at <https://github.com/bastikr/boolean.py> . Both are co-maintained by @pombredanne
- **ABCD aka AboutCode Data** is a simple set of conventions to define data structures that all the AboutCode tools can understand and use to exchange data. The specification lives in this repository. .ABOUT files and ScanCode toolkit data are examples of this approach. Other projects such as <https://libraries.io> and [OSS Review Toolkit](<https://github.com/heremaps/oss-review-toolkit>) also use these conventions.
- **[DeltaCode]**(<https://github.com/nexB/deltacode>) is a command line tool to compare scans and determine if and where there are material differences that affect licensing. The lead maintainer is @majurg
- **[VulnerableCode]**(<https://github.com/nexB/vulnerablecode>): an emerging server-side application to collect and track known package vulnerabilities.

1.3 Document Maintenance

1.3.1 Document Software Setup

ScanCode Toolkit documentation is built using Sphinx. See <http://www.sphinx-doc.org/en/master/index.html>

ScanCode Toolkit documentation is distributed using “Read the Docs”. See <https://readthedocs.org/>

Individual document files are in reStructuredText format. See <http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html>

You create, build, and preview ScanCode Toolkit documentation on your local machine.

You commit your updates to the ScanCode Toolkit repository on GitHub, which triggers an automatic rebuild of <https://scancode-toolkit.readthedocs.io/en/latest/index.html>

1.3.2 Clone ScanCode Toolkit

To get started, create or identify a working directory on your local machine.

Open that directory and execute the following command in a terminal session:

```
git clone https://github.com/nexB/scancode-toolkit.git
```

That will create an /scancode-toolkit directory in your working directory. Now you can install the dependencies in a virtualenv:

```
cd scancode-toolkit
python3.6 -m venv .
cd docs
source bin/activate
```

Now you can install Sphinx and the format theme used by readthedocs:

```
pip install Sphinx sphinx_rtd_theme
```

Now you can build the HTML documents locally:

```
cd docs
make html
```

Assuming that your Sphinx installation was successful, Sphinx should build a local instance of the documentation .html files:

```
open build/html/index.html
```

You now have a local build of the ScanCode Toolkit documents.

1.3.3 Improve ScanCode Toolkit Documents

Before you begin creating and modifying ScanCode Toolkit documents, be sure that you understand the basics of reStructuredText as explained at <http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html>

Ensure that you have the latest ScanCode Toolkit files:

```
git pull
git status
```

Use your favorite text editor to create and modify .rst files to make your documentation improvements.

Review your work:

```
cd docs
make html
open build/html/index.html
```

1.3.4 Share ScanCode Toolkit Document Improvements

Follow standard git procedures to upload your new and modified files. The following commands are examples:

```
git status
git add source/index.rst
git add source/how-to-scan.rst
git status
git commit -m "New how-to document that explains how to scan"
git status
git push
git status
```

The ScanCode Toolkit webhook with ReadTheDocs should rebuild the documentation. You can review your results online.

CHAPTER 2

Tutorial Documents

Tutorial documents provide specific instructions to help you get started.

CHAPTER 3

How-To Documents

How-To documents explain how to accomplish specific tasks.

CHAPTER 4

Reference Documents

Reference documents describe application concepts in depth.

CHAPTER 5

Discussion Documents

Discussion documents provide insights into integration of the application into your own software development life cycle.

CHAPTER 6

Indices and Tables

- `genindex`
- `modindex`
- `search`