ScanCode Workbench Documentation

Release 0.0.1

nexB Inc.

Sep 07, 2022

TABLE OF CONTENTS

1 Overview **3**
- 1.1 What Is ScanCode Workbench? . . . 3
- 1.2 Organization of the Documentation . . . 3
- 1.3 Important Links . . . 3

2 Getting Started **5**
- 2.1 Download and Install . . . 5
- 2.2 ScanCode Workbench-ScanCode Toolkit Compatibility . . . 5
- 2.3 Open ScanCode Workbench and Load a ScanCode Toolkit Scan . . . 5
- 2.4 Try a Sample Scan . . . 6

3 How-To Guides **7**
- 3.1 Load Your Data . . . 7
  - 3.1.1 Import a JSON File . . . 7
  - 3.1.2 Open and Save a SQLite File . . . 8
- 3.2 Look Up Your Scan Version . . . 10
- 3.3 Explore Your Data . . . 10
  - 3.3.1 How-To: Navigate the Chart Summary View . . . 10
- 3.4 Export JSON Records . . . 11
  - 3.4.1 Export a JSON file . . . 11
- 3.5 Troubleshooting . . . 11
  - 3.5.1 How-To: Check for Errors in the Developer Tools . . . 11
- 3.6 Special Topics . . . 12
  - 3.6.1 How-To: Use License Policies with ScanCode Workbench . . . 12

4 UI Reference **15**
- 4.1 Directory Tree . . . 15
- 4.2 Table View . . . 15
- 4.3 File Info Dashboard . . . 15
- 4.4 License Info Dashboard . . . 16
- 4.5 Package Info Dashboard . . . 16
- 4.6 Chart Summary View . . . 16

5 Technical Reference **17**
- 5.1 Underlying Technology . . . 17
- 5.2 Platform Support . . . 17
  - 5.2.1 ScanCode Workbench Supported Platforms . . . 17
  - 5.2.2 Electron Supported Platforms . . . 18

6 Contribute **21**
- 6.1 Building . . . 21

6.1.1 Clone, Install, Build and Run . . . 21
6.1.2 Building Requirements . . . 21
6.1.3 Release Instructions . . . 22
6.2 Testing . . . 22
6.3 Reporting Issues . . . 23
6.4 Contributing Code . . . 23
6.5 Community Channels . . . 23
6.6 Google Summer of Code 2022 . . . 23
6.6.1 Overview . . . 23
6.6.2 Goals achieved . . . 23
6.6.3 Quick look . . . 24
6.6.4 Links . . . 24
6.6.5 Post GSoC - Future Plans . . . 24
6.6.6 Closing Thoughts . . . 24

7 License **25**

Index **27**

Welcome to the documentation for ScanCode Workbench!

CHAPTER

ONE

OVERVIEW

1.1 What Is ScanCode Workbench?

ScanCode Workbench is a desktop application designed to view and work with ScanCode Toolkit scans. With ScanCode Workbench, you can:

- Load a ScanCode Toolkit `.json` scan of your codebase.
- Use an advanced visual UI to analyze license and other notices identified by ScanCode Toolkit.

1.2 Organization of the Documentation

This documentation is organized in six sections:

- The *Getting Started* section – the suggested entry point for all new users – will walk you through the process of downloading, installing and opening ScanCode Workbench and loading a ScanCode Toolkit scan.
- The *How-To Guides* section contains feature-specific guides and can be read in any order as the need arises.
- The *UI Reference* section provides an overview of each of ScanCode Workbench's data views.
- The *Technical Reference* section summarizes ScanCode Workbench's underlying technology and platform support.
- The *Contribute* section is intended for advanced users and contributors to ScanCode Workbench development.
- The *License* section provides summary licensing information for ScanCode Workbench.

1.3 Important Links

- Repository: https://github.com/nexB/scancode-workbench
- Issues: https://github.com/nexB/scancode-workbench/issues

CHAPTER

TWO

GETTING STARTED

2.1 Download and Install

- You can download the latest ScanCode Workbench release for your Windows, OS X or Linux operating system from the ScanCode Workbench releases page. Once downloaded, you'll find the ScanCode Workbench executable in the `ScanCode-Workbench-<os>-x64-<version>` folder. On Windows 10, for example, the executable will be named *ScanCode-Workbench.exe*.
- If you're interested in digging into the code, you can also use ScanCode Workbench by cloning the GitHub repository and building it yourself – see the *Contribute/Building* section for details.

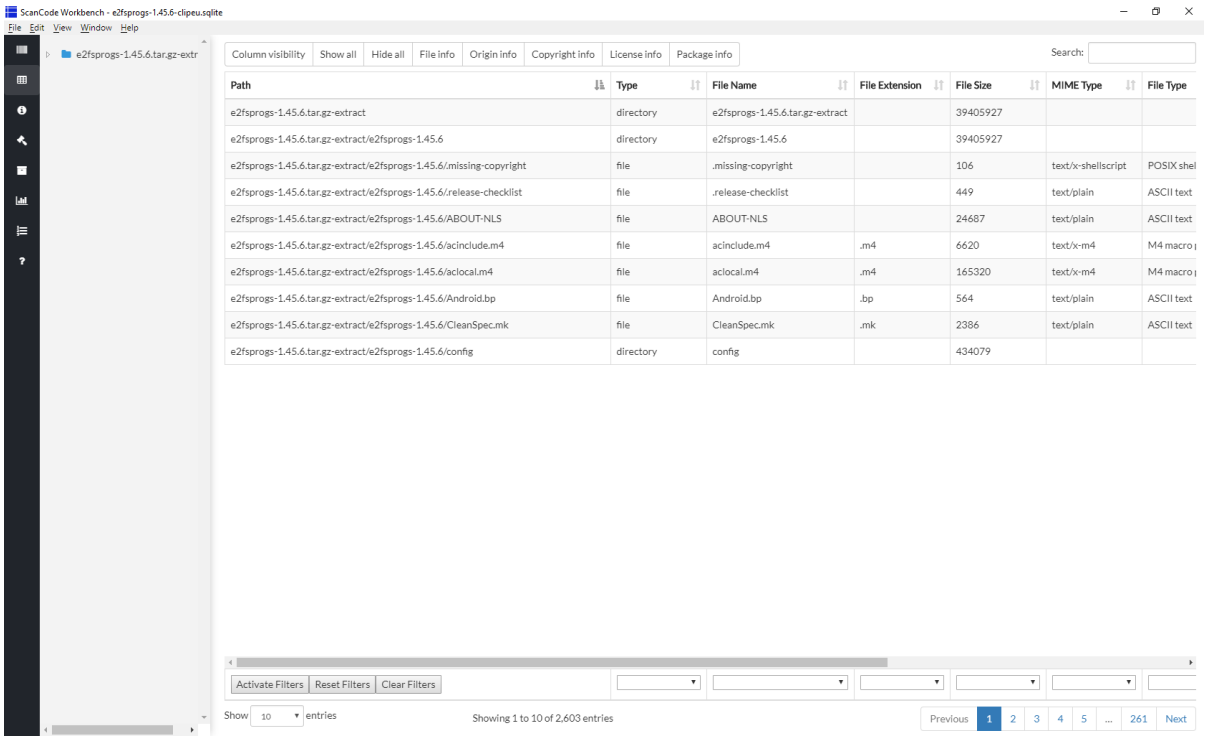
2.2 ScanCode Workbench-ScanCode Toolkit Compatibility

- ScanCode Workbench >= v3.1.1 is only compatible with scans from ScanCode v3.1.1 and above that have been run with the ScanCode Toolkit `-i` option.
  - A list of available ScanCode Toolkit options is available in the ScanCode Toolkit documentation: How to set what will be detected in Scan.
- You would typically create your scan with the following command: `./scancode -clipeu <input> <output_file>`

2.3 Open ScanCode Workbench and Load a ScanCode Toolkit Scan

- Double-click the ScanCode Workbench executable you downloaded. You'll probably want to maximize the application once it has opened.
- Import your JSON scan file and save it as a SQLite file (ScanCode Workbench works with the data in a SQLite database).
  - `File` > `Import JSON File` (`Ctrl` + `I`) ==> opens `Open a JSON File` window.
  - Select your JSON scan and click `Open` ==> opens `Save a SQLite Database File` window.
  - Keep or modify the default SQLite filename and click `Save`.
- You're now looking at your scan data displayed in the Table View – the Table View itself is on the right, and the Directory Tree (which is visible in all views) is on the left.

You can find additional details in the *How-To Guides* section below.



Your first imported ScanCode Toolkit Scan.

2.4 Try a Sample Scan

We've also provided a set of sample scans that you can review in ScanCode Workbench in order to get a sense of its functionality and the types of information captured by a scan.

CHAPTER

THREE

HOW-TO GUIDES

The ScanCode Workbench How-To Guides will walk you through loading and analyzing a `-clipeu` scan of `e2fsprogs-1.45.6.tar.gz`.

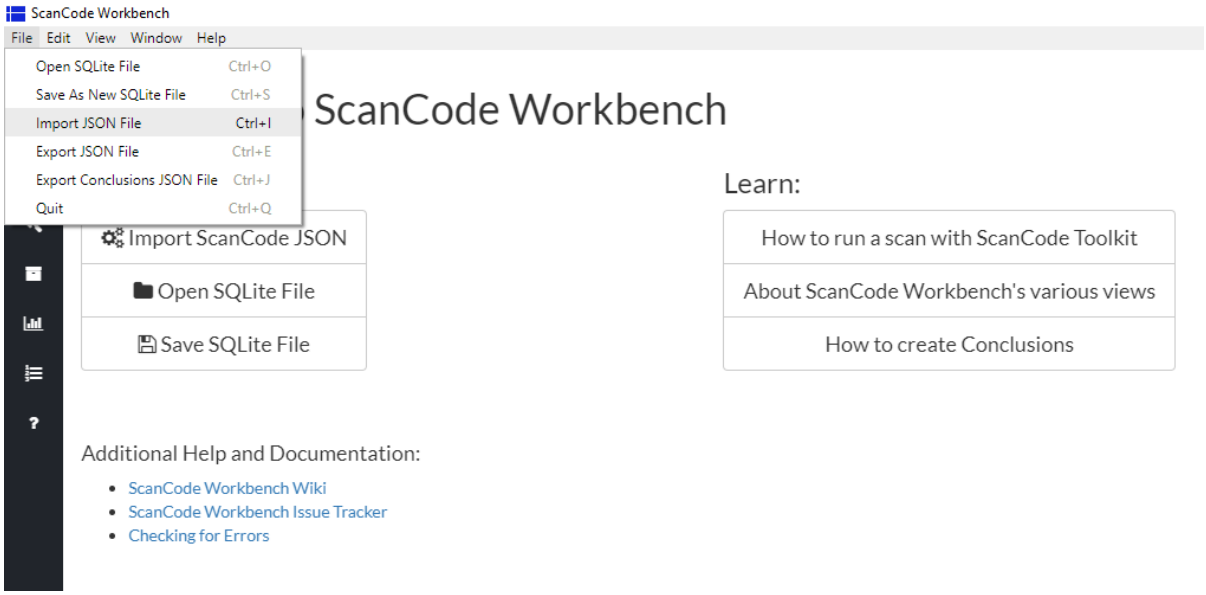
Note: If you'd like to follow along, you can download a copy of the `e2fsprogs-1.45.6` package from SourceForge, run a `-clipeu` scan with ScanCode Toolkit, load the `.json` output into ScanCode Workbench and work your way through the How-To Guides.

3.1 Load Your Data

3.1.1 Import a JSON File

To import a ScanCode JSON file:

- Open the `File` menu and select `Import JSON File` (keyboard shortcut: `Ctrl+I` or `+I`).



Import a JSON scan file.

- In the dialog window that opens, navigate to the JSON file you want to import, select the file and click `Open`.

- You will then be prompted to choose a filename and location to save the JSON file as a SQLite database file. Add a filename, select the folder in which you want to save the SQLite database file, and click `Save`.
- ScanCode Workbench will then create a SQLite database file from your JSON file, indicated by the status message `Creating Database ...`

The JSON scan file is being converted to a SQLite file.

- Once the process has finished, the status message will be replaced by an expandable code tree (the **Directory Tree**) and, to the right of the tree, a table (the **Table View**) displaying provenance information generated by ScanCode.

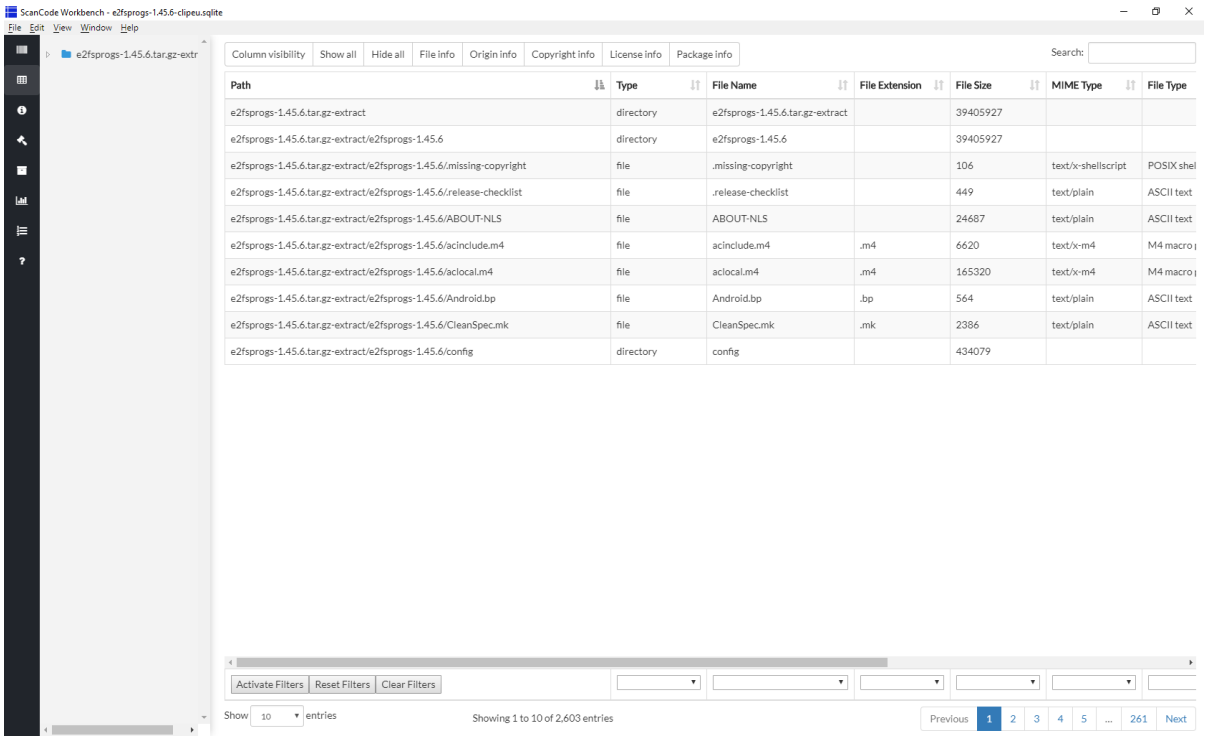
We've provided a set of sample scans that you can quickly review in ScanCode Workbench in order to get a sense of its functionality and the types of information captured by a scan. The samples are located at https://github.com/nexB/scancode-workbench/tree/develop/samples.

3.1.2 Open and Save a SQLite File

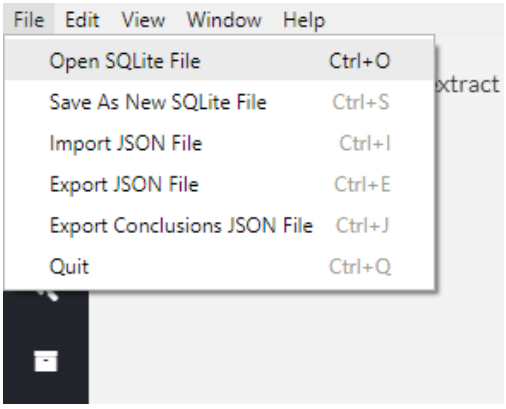
Open a SQLite File

Once you've imported your JSON scan and saved it as SQLite, you'll continue your review and analysis work solely in the SQLite file.

- To open a SQLite File:
  - Select the `File` menu and then select `Open SQLite File` (keyboard shortcut: `Ctrl+O` or `+O`).
  - In the dialog window that opens, navigate to the SQLite file you want to open, select the file and click `Open`.



The SQLite version of your JSON scan is ready for your analysis.

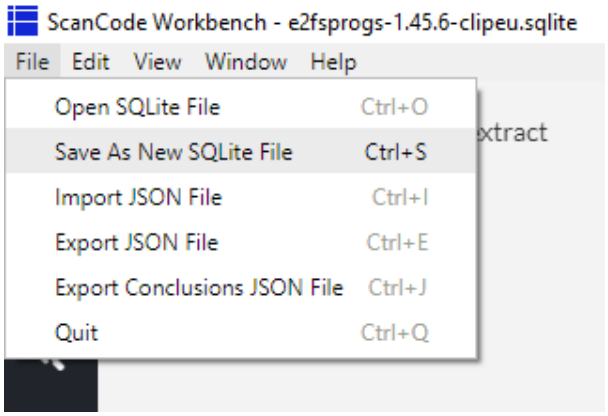


Open a SQLite file.

Save as a New SQLite File

There may be times when you need to preserve the current state of your work and continue working in a separate file. No problem – just save your work as a new SQLite file.

- To save as a new SQLite file:



Save as new SQLite file.

  - Select the `File` menu and then select `Save As New SQLite File` (keyboard shortcut: `Ctrl+S` or `+S`).
  - In the dialog window that opens, add a name for the file, navigate to the directory in which you want to save the file and click `Save`.

3.2 Look Up Your Scan Version

[to come]

3.3 Explore Your Data

3.3.1 How-To: Navigate the Chart Summary View

Display the view

Once you have a SQLite file loaded into ScanCode Workbench, displaying the Chart Summary View is easy:

1. Select a file or directory in the Tree View on the left.
2. Click the chart icon in the sidebar or open the `View` menu and select `Chart Summary View` (keyboard shortcut: `Ctrl+Shift+D` or `+Shift+D`).

Select an attribute

Use the dropdown at the top of the view to select the attribute you want to examine (e.g., `Copyright Statements`, `License Key`). These attribute values are detected from ScanCode, and can also be viewed in the Table View.

When you select an attribute, the Chart Summary View will automatically refresh to display a horizontal bar chart showing – in descending order of frequency – each value identified in the scanned codebase for the selected attribute and the number of times it occurs in the codebase. You can also see the value for a particular entry in the bar chart in a tooltip that appears when you move your cursor over the text on the left or the bar on the right.

Filter Chart Summary

You can further filter the summary results by choosing a specific directory or file in the Tree View. The chart will then only show results for that selected directory or file.

[to come]

3.4 Export JSON Records

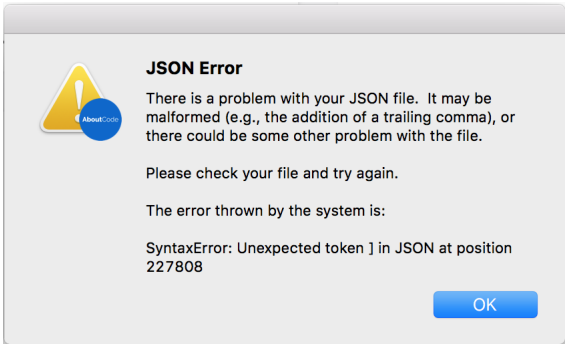
3.4.1 Export a JSON file

- To export a JSON file:
  - Select the `File` menu and then select `Export JSON File` (keyboard shortcut: `Ctrl+E` or `+E`).
  - In the dialog window that opens, add a name for the file, navigate to the directory in which you want to save the file and click `Save`.

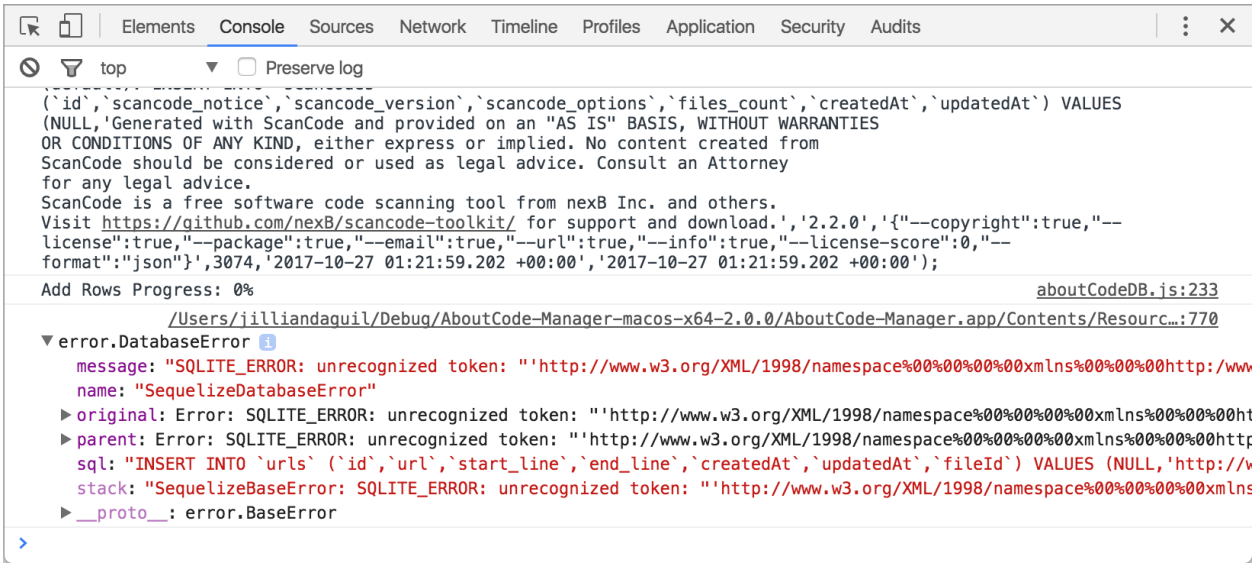
3.5 Troubleshooting

3.5.1 How-To: Check for Errors in the Developer Tools

When an unexpected error occurs in ScanCode Workbench, you will normally see a dialog message which provides details about the error and allows you to create an issue.



If you can reproduce the error, use this approach to get the stack trace and report the issue. Open the Developer Tools with `Ctrl+Shift+I` or `Alt+Cmd+I`. From there, click the Console tab. Include the error that is logged in the issue in a code block or a file attachment.



[to come]

3.6 Special Topics

3.6.1 How-To: Use License Policies with ScanCode Workbench

ScanCode Workbench now has basic support for tracking and viewing license policies that have been applied to a ScanCode Toolkit scan. In order for things to work, your initial scan must be run with the `--license-policy` option. For more information, see the License Policy Plugin section of the ScanCode Toolkit ReadTheDocs.

The basics

While the License Policy Plugin can be customized with any number of custom fields and values, ScanCode Workbench currently only supports a pre-defined set of policy labels.

license_key	label
scancode_license_key	Approved License
scancode_license_key	Prohibited License
scancode_license_key	Recommended License
scancode_license_key	Restricted License

This means in order to take advantage of ScanCode Workbench's policy features, your `license-policy.yml` needs to have `license_key` and `label` fields at the very least.

Additionally, in order to take advantage of policy visualizations, `label` values must be one of the 4 above values: Approved License, Prohibited License, Recommended License or Restricted License. Later versions of ScanCode Workbench will eventually evolve to support more dynamic policy values.

Here is a simple example of a valid `license-policy.yml` file:

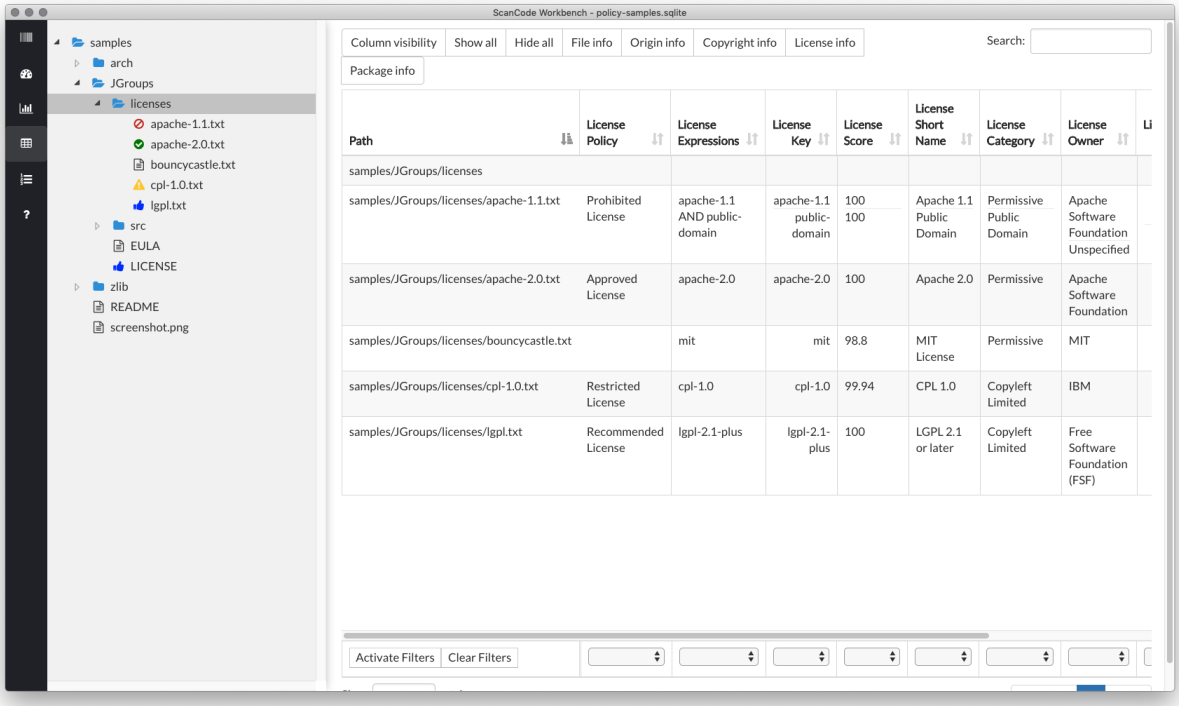
```
license_policies:
-   license_key: apache-2.0
    label: Approved License
```

(continues on next page)

(continued from previous page)

```
-   license_key: apache-1.1
    label: Prohibited License
-   license_key: lgpl-2.1-plus
    label: Recommended License
-   license_key: cpl-1.0
    label: Restricted License
```

After running a scan with that particular `license-policy.yml` file, viewing the scan in ScanCode Workbench will look like the following:



As you can see, files which have detected licenses that fall under a particular policy will be shown in the JSTree view with specific icons. This way, you are able to quickly see what files fall under a specific policy.

Additionally, policy details can be found in the scan data view in their own column: License Policy. This column has been added to both the "Origin" column set and "License info" column set.

[to come]

CHAPTER

FOUR

UI REFERENCE

[Intro to come.]

4.1 Directory Tree

[Intro to come.]

An interactive directory tree is always present on the left side of the application. The tree is expandable and collapsible. This allows the user to navigate the codebase structure. If a directory is selected, only that directory and its sub-files and folders will be shown in the view. Similarly, if a single file is selected, only information for that selected file will be shown.

4.2 Table View

[Intro to come.]

In the table view, the available clues detected by ScanCode Toolkit are shown in a tabular format. A user can see provenance clues such as license and copyright information detected by ScanCode. A user can also see the file information (e.g., file type, file size, etc) and package information (package type, primary language of package) that was detected.

The columns can be sorted as well as shown or hidden based on what the user's preferences. Searching for specific clues (license names, copyrights, etc.) is also available in this view.

4.3 File Info Dashboard

[Intro to come.]

[Screenshots and narrative to come.]

4.4 License Info Dashboard

[Intro to come.]

[Screenshots and narrative to come.]

4.5 Package Info Dashboard

[Intro to come.]

[Screenshots and narrative to come.]

4.6 Chart Summary View

[Intro to come.]

With the chart summary view, a user can select a node in the directory tree (i.e., a directory, folder or file) and display a horizontal bar chart listing the values identified in the scanned codebase – that is, the clues detected by ScanCode Toolkit – for a number of different attributes. The attributes are a subset of the columns displayed in the table view, and can be selected by clicking the dropdown at the top of the view. The chart displays the full range of values for the selected directory tree node and attribute and the number of times each value occurs in the scanned codebase.

CHAPTER

FIVE

TECHNICAL REFERENCE

5.1 Underlying Technology

- ScanCode Workbench is based on Electron and works on Windows, OS X and Linux operating systems.

5.2 Platform Support

5.2.1 ScanCode Workbench Supported Platforms

Our approach for platform support is to focus on one primary release for each of Linux, MacOS and Windows. The Priority definitions are:

1. Primary - These are the primary platforms for build/test/release on an ongoing basis.
2. Secondary - These are platforms where the primary ScanCode Workbench release for the corresponding OS Group should be forward-compatible, e.g., Windows 7 build should work on Windows 10. Issues reported and traced to a Secondary platform may not be fixed.
3. Tertiary - These are any other platforms not listed as Primary or Secondary. In these cases, we will help users help themselves, but we are likely not to fix Issues that only surface on a Tertiary platform.

OS Group	Desktop OS Version	Arch	Priority	Notes
Windows	Windows 7 SP1	x64	1	
Windows	Windows 10 SP?	x64	2	
MacOS	10.9 Mavericks	x64	1	
MacOS	10.10 Yosemite	x64	2	
MacOS	10.11 El Capitan	x64	2	
MacOS	10.12 Sierra	x64	2	
Linux Deb	Ubuntu 12.04	x64	1	From Electron Docs: The prebuilt ia32 (i686) and x64 (amd64) binaries of Electron are built on Ubuntu 12.04.
Linux Deb	Ubuntu 14.xx	x64	2	Verified to be able to run the prebuilt binaries of Electron.
Linux Deb	Ubuntu 16.xx	x64	2	Verified to be able to run the prebuilt binaries of Electron.
Linux	Fedora 21	x64	2	Verified to be able to run the prebuilt binaries of Electron.
Linux	Debian 8	x64	2	Verified to be able to run the prebuilt binaries of Electron.
Linux RH	CentOS 7.xx	x64	?	
Linux RH	RHEL 7.xx	x64	?	

5.2.2 Electron Supported Platforms

https://electronjs.org/docs/tutorial/support#supported-platforms

The following platforms are supported by Electron:

MacOS

Only 64-bit binaries are provided for MacOS, and the minimum MacOS version supported is MacOS 10.9.

Windows

Windows 7 and later are supported, while older operating systems are not supported (and do not work). Both ia32 (x86) and x64 (amd64) binaries are provided for Windows. Please note: the ARM version of Windows is not supported for now.

Linux

The prebuilt ia32 (i686) and x64 (amd64) binaries of Electron are built on Ubuntu 12.04, and the ARM binary is built against ARM v7 with hard-float ABI and NEON for Debian Wheezy.

Whether the prebuilt binary can run on a distribution depends on whether the distribution includes the libraries that Electron is linked to on the building platform, so only Ubuntu 12.04 is guaranteed to work, but the following platforms are also verified to be able to run the prebuilt binaries of Electron:

- Ubuntu 12.04 and later
- Fedora 21
- Debian 8

CHAPTER

SIX

CONTRIBUTE

[Intro to come.]

6.1 Building

6.1.1 Clone, Install, Build and Run

You'll need Node.js (which comes with npm) installed on your computer in order to build this app. (See below for a list of platform-specific requirements.) Then, from your command line:

```
# Clone this repository
$ git clone https://github.com/nexB/scancode-workbench.git

# Go into the repository
$ cd scancode-workbench

# Install dependencies and run the app
$ npm install

# Rebuild native Node.js modules against the app version of Node.js
# MacOS, Linux and Git Bash on Windows
$ $(npm bin)/electron-rebuild
# Windows except for Git Bash
> .\node_modules\.bin\electron-rebuild.cmd

# Run the app
$ npm start
```

6.1.2 Building Requirements

Linux

- Python 3.7
- Node.js version 6.x or later
- npm 3.10.x or later but <= 5.2.0 (run `npm install npm@5.2.0 -g`)

MacOS

- Python 3.7
- Node.js >=6.x or later but <=8.9.4
- npm 3.10.x or later but <= 5.2.0 (run `npm install npm@5.2.0 -g`)
- Command Line Tools for Xcode (run `xcode-select --install` to install)

Windows

- Node.js 6.x or later
- npm 3.10.x or later but <= 5.2.0 (`run npm install npm@5.2.0 -g`)
- Python v3.7.x
  - Make sure your Python path is set. To verify, open a command prompt and type `python --version`. Then, the version of python will be displayed.
- Visual C++ Build Environment:
  - Either:
  - Option 1: Install Visual C++ Build Tools 2015 (or modify an existing installation) and select Common Tools for Visual C++ during setup. This also works with the free Community and Express for Desktop editions.
  - Option 2: Visual Studio 2015 (Community Edition or better)
  - Note: Windows 7 requires .NET Framework 4.5.1
  - Launch cmd, `npm config set msvs_version 2015`

6.1.3 Release Instructions

You can build a `dist` directory containing executables for any one of three target platforms by running:

```
$ python build.py
```

After building is done, you can find ScanCode-Workbench under `dist/ScanCode-Workbench-<os>-x64-<version>`. Archives (`tar.gz` and `.zip`) are also built.

Note: A build for any of the three target platforms must be executed on the targeted platform.

6.2 Testing

Test ABCM functionality using:

```
$ npm test
```

[Maybe add details about location of tests, examples of how to create?]

6.3 Reporting Issues

If you want to report an issue in case you find a bug or want to suggest a new feature, report here.

6.4 Contributing Code

[to come]

6.5 Community Channels

If you have a question, a suggestion or find a bug, enter an issue.

For questions and chats, you can join the Gitter channel at https://gitter.im/aboutcode-org/discuss

[more to come]

6.6 Google Summer of Code 2022

By, Omkar Phansopkar
Project: Scancode-Workbench
Organization - AboutCode

6.6.1 Overview

ScanCode Workbench provides an advanced visual UI to help you quickly evaluate license and other notices identified by ScanCode. ScanCode detects licenses, copyrights, packages and other interesting information in your code that can be visualised efficiently using workbench.

The GSoC project's goal was to refactor ScanCode Workbench to a React + Typescript implementation and improve various sections of the application including Table view, file uploads, data sync across components, etc. Plus, the workbench wasn't updated for a long time, to support the latest scans from scancode-toolkit, hence the schema and UI had to be changed accordingly.

6.6.2 Goals achieved

- Refactor Workbench from vanilla javascript to React + Typescript implementation
- Update Workbench to support the latest scancode-toolkit output format. (currently supports outputs from scancode-toolkit v31.1.1)
- Improve User experience in Tableview, Home page, etc
- Add convenience features like Drop files, History records, etc.
- Add new sections for Scan Info, packages-dependencies info, etc
- Automated releases using github actions (except for apple-silicon builds)

6.6.3 Quick look

6.6.4 Links

- Download ScanCode Workbench beta
- Download latest ScanCode Workbench
- GSoC Project Details
- Proposal
- ScanCode Workbench
- ScanCode Toolkit

6.6.5 Post GSoC - Future Plans

I wish to carry on with the development of Scancode-workbench and implement the ideas suggested by my mentors. Workbench isn't yet utilising all the scan data it imports, I wish to propose best ways to express this data. Moreover, with new release formats of scancode-toolkit, workbench must adapt and update its schema & UI at pace with scancode-toolkit.

6.6.6 Closing Thoughts

It was a great experience contributing to Workbench, Aboutcode team provided me with the freedom to implement UI improvements and suggested a lot of improvements throughout the journey. I learned a lot of things from working on an electron app to creating automated GitHub release scripts.

Heartiest thanks to @steven-esser @AyanSinhaMahapatra @TG1999 @pombredanne and all the About code members for your constant feedback and support over the course of the project.

You can connect with me here
https://github.com/OmkarPh
https://www.linkedin.com/in/omkarphansopkar

To the reader, Have a look at aboutcode's amazing projects. Surely you'll find something to contribute that suits your preference

See ya

CHAPTER

SEVEN

LICENSE

- Apache-2.0
- Multiple licenses (LGPL, MIT, BSD, etc) for third-party components.
- See the NOTICE file for more details.

INDEX

B

Building, 21

C

Chart Summary View, 16
Community Channels, 23
Contribute, 21
Contributing Code, 23

D

Directory Tree, 15

E

Explore Your Data, 10
Export JSON Records, 11

F

File Info Dashboard, 15

G

Getting Started, 5

H

How-To Guides, 7
How-To: Check for Errors in the Developer Tools, 11
How-To: Navigate the Chart Summary View, 10
How-To: Use License Policies with ScanCode Workbench, 12

I

Import a JSON File, 7

L

License, 25
License Info Dashboard, 16
Load Your Data, 7
Look Up Your Scan Version, 10

O

Open and Save a SQLite File, 8
Overview, 3

P

Package Info Dashboard, 16

R

Reporting Issues, 23

S

Special Topics, 12

T

Table View, 15
Technical Reference, 17
Testing, 22
Troubleshooting, 11

U

UI Reference, 15