ScanCode Workbench Documentation

nexB Inc. and others

Jan 17, 2025

TABLE OF CONTENTS

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 *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 Underlying Technology

- ScanCode Workbench is a cross-platform application built using the Electron framework that works on Windows, macOS and Linux operating systems.
- **It uses**
  - TypeScript as the primary language.
  - React for user interface.
  - Sequelize ORM for database access.
  - Sqlite3 for managing sqlite database.

1.4 Platform Support

- Linux - `x64`
- Windows 10/11 - `x64`
- MacOS - `x64`, `arm64`

1.5 Important Links

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

CHAPTER

TWO

GETTING STARTED

2.1 Download and Install

- **ScanCode Workbench is available for Windows, macOS and Linux -**
  ScanCode Workbench releases.
- Once downloaded, you'll find the ScanCode Workbench executable inside the `ScanCode Workbench-<os>-<arch>` folder.

  On Windows 10, for example, the executable will be named *ScanCode-Workbench-4.0.1.exe*.

Note

Browsers on MacOS add a `com.apple.quarantine` extended attribute to downloaded apps. which prevents you from running the app & shows error dialog: Application Is Damaged and Can't Be Opened. You Should Move It To The Trash

Delete the attribute using `xattr` utility before running the app for first time:

```
xattr -d com.apple.quarantine ~/Downloads/ScanCode-Workbench-4.0.3-darwin-arm64/
↪ScanCode-Workbench-4.0.3.app
```

Note

For CentOS (or linux distros without the new libstdc++), follow these steps:

- Install the new libstdc++ library:

  ```
  yum provides libstdc++
  ```

- Update LD_LIBRARY_PATH:

  ```
  export LD_LIBRARY_PATH="/usr/local/lib64/:$LD_LIBRARY_PATH"
  ```

- Run the application with `no-sandbox` option:

  ```
  ./ScanCode-Workbench-4.0.3-linux-x64/ScanCode-Workbench-4.0.3 --no-sandbox
  ```

- 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 v4.x is only compatible with scans from ScanCode v32.x and above that have been run with ScanCode Toolkit.
- You would typically create your scan with the following command:

```
scancode -clipeu <input> --json-pp <output_file>
```

Note

A list of available ScanCode Toolkit options is available in the ScanCode Toolkit documentation: How to set what will be detected in Scan. Also see How to Run a Scan for more details on running a ScanCode Toolkit scan.

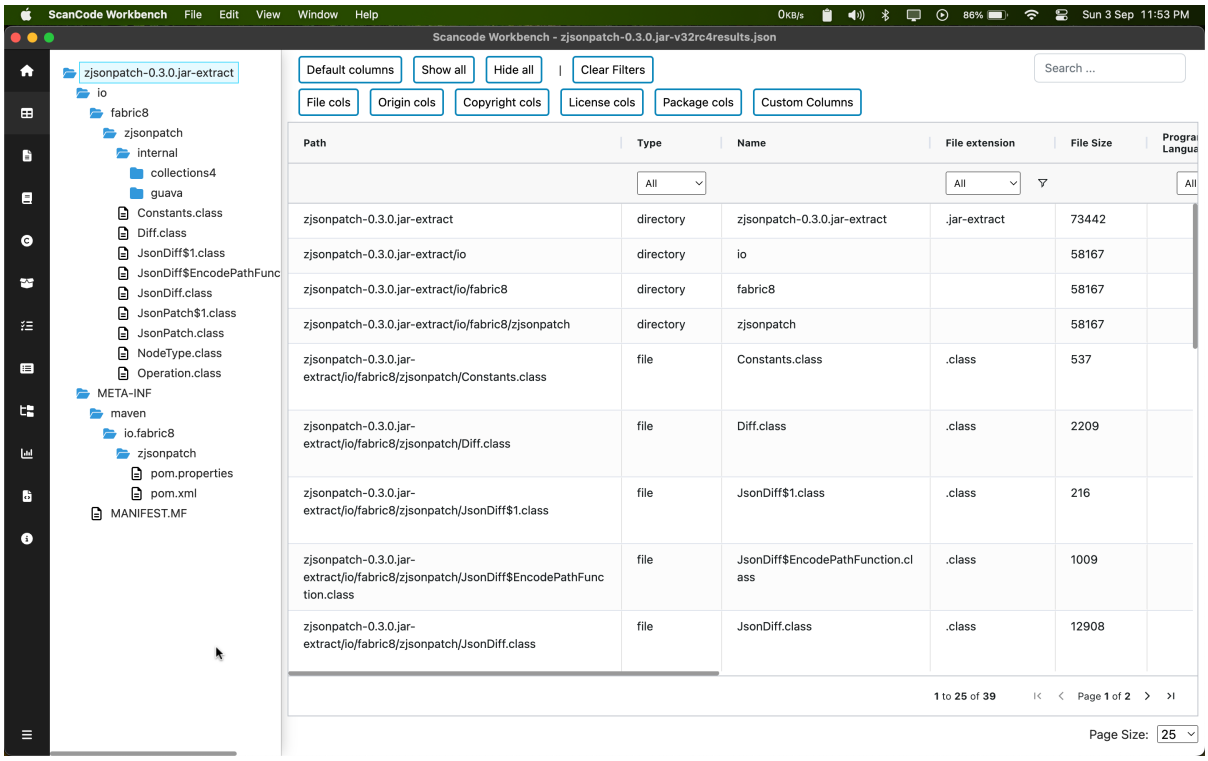
2.3 Open ScanCode Workbench and Load a ScanCode Toolkit Scan

- 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 (shown in resource level views) is on the left.

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

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.



Your first imported ScanCode Toolkit Scan.

CHAPTER

THREE

HOW-TO GUIDES

The ScanCode Workbench How-To Guides will walk you through loading and analyzing a *-clipeu* scan. The guides are not meant to be exhaustive, but rather to give you a taste of what you can do with the Workbench.

For full set of features, please refer to the *UI Reference*.

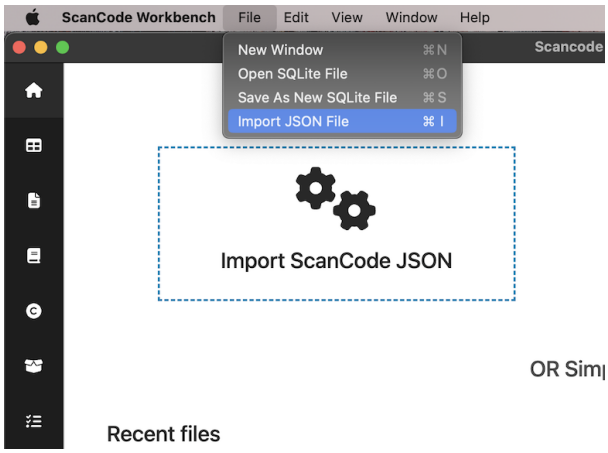
3.1 Load Your Data

3.1.1 Import a JSON File

To import a ScanCode JSON file:

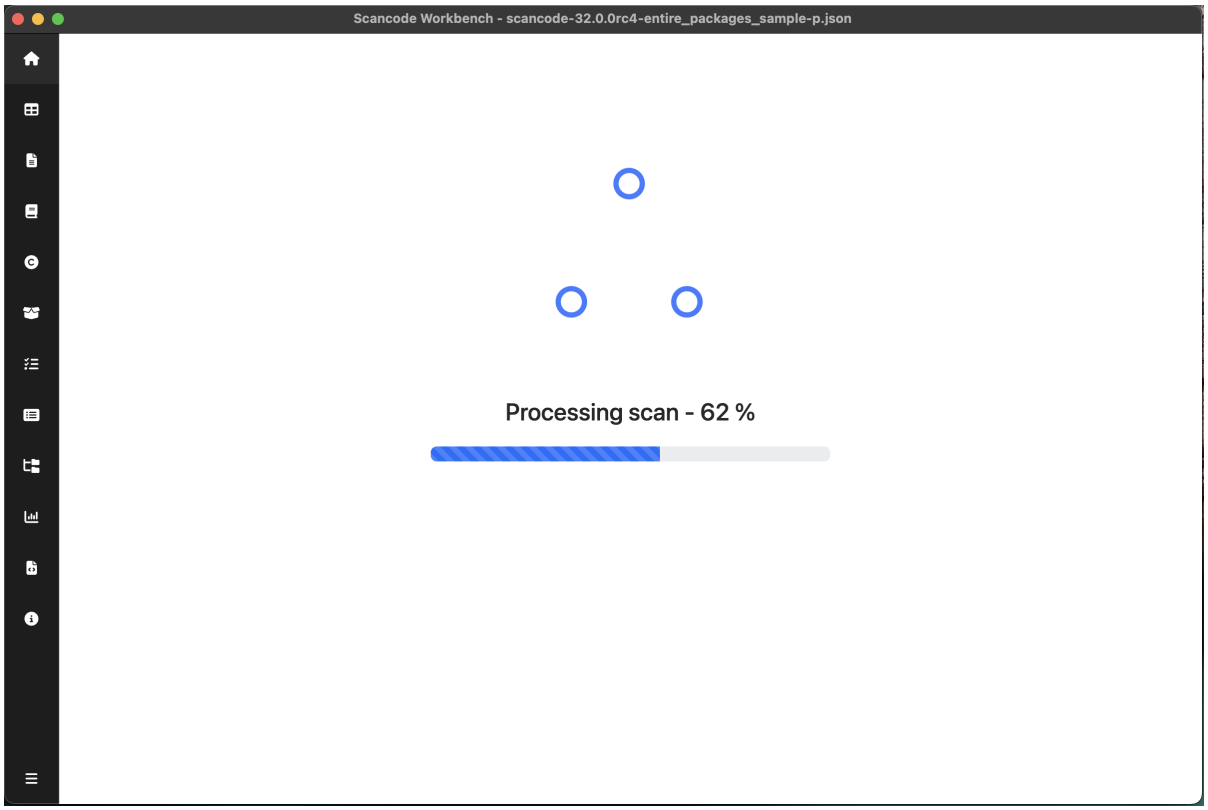
We've provided a set of sample scans that you can quickly review in ScanCode Workbench in order to get a sense of its functionality and the types of information captured by a scan. The samples are located at https://github.com/aboutcode-org/scancode-workbench/tree/develop/samples.

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



Click Import a JSON scan file.
Or Simply click on the Import ScanCode JSON button

- 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 `Processing scan - <Progress percentage>`



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.

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

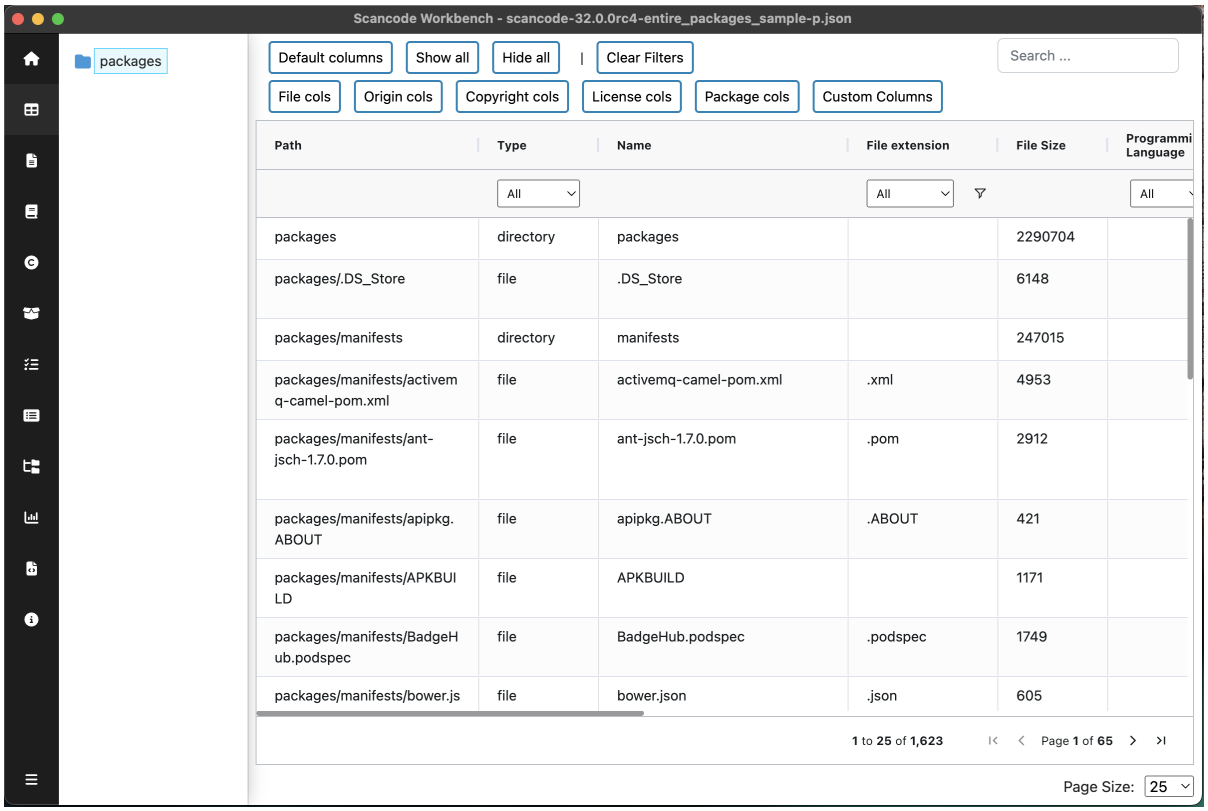
- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

3.1.2 Open or Save a SQLite File

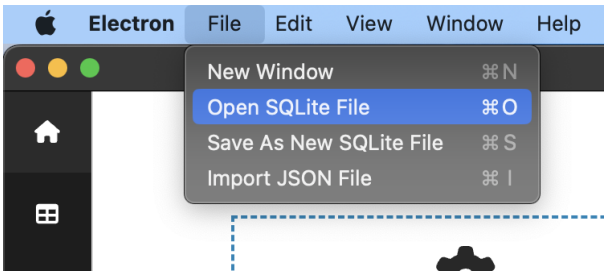
Open a SQLite File

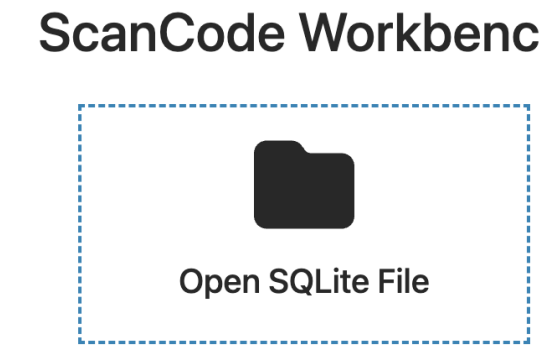
Once you've imported your JSON scan it is parsed & saved as a SQLite file, which can be later imported quickly compared to opening json scan again

- To open a SQLite File:
  - Select the `File` menu and then select `Open SQLite File` (keyboard shortcut: `Ctrl`+O or +O)



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





Open a SQLite file.

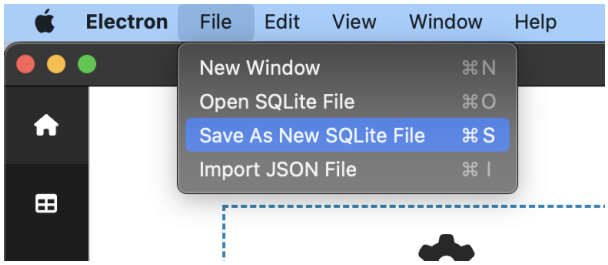
Or Simply Click on `Open SQLite file` quick action button.

– In the dialog window that opens, navigate to the SQLite file you want to open, select the file and click `Open`.

Save as a New SQLite File

There may be times when you need to save your work as a new SQLite file.

- To save as a new SQLite file:



OR

Save as new SQLite file.

– Select the `File` menu and then select `Save As New SQLite File` (keyboard shortcut: `Ctrl`+`S` or +`S`)

Or Simply click `Save SQLite File` quick action button.

– 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 information

To view scan information:

Hover to expand the sidebar with icons on the left. Click on Scan Information or, open the `View` menu and select `Scan Info`

You can view various details about the scan like scan tool name & version, options, platform etc. More detailed explanation about the view can be found at *Scan Info*

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

3.3 Explore Your Data

3.3.1 How-To: Navigate the Table View

Display the Table view

Once you have a SQLite file loaded into ScanCode Workbench, you're automatically redirected to the Tableview. To navigate to Tableview from other views you can click on *Table View* in the sidebar or open the `View` menu and select `Table View`

Click on *Package Explorer* in the sidebar or open the `View` menu and select `Table View`

Select a path

You can collapse/expand the FileTree on the left and select a file/directory. Only the files under this path will be visible in the tableview

Select Preset columns

You can select one of the available column groups, grouped based on their nature such as `Copyright Cols`, `Origin Cols`, `License Cols`

For entire UI reference, Read *Table View*

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

3.3.2 How-To: Navigate the License Explorer View

Display the License Explorer view

Once you have a SQLite file loaded into ScanCode Workbench,

Click on *License Explorer* in the sidebar or open the `View` menu and select `License Explorer`

Select and search for a License

On the left pane, Licenses are shown in two categories: *License Detections* & *License clues*. You can select any of these licenses & more detailed information about that license will be reflected on the right

Review licenses

When reviewing large number of licenses over multiple sessions, You can tick the checkbox beside any of theselicense detections or clues to mark them as reviewed or unreviewed so that you can continue from where you left off.

Based on this review status, you can filter the licenses using filter options on the top

Matches & File regions table

For the selected license, you can view the details of matches resulting into the detection and the file regions where the license was detected.

In the file regions table, You can click on the data file path to view that path in Tableview. Similarly, detection origin is a clickable icon. It determines the source of detection.

- Package icon indicates that the detection belongs to a package manifest, you can click on it to view the source package manifest in *Package Explorer*.
- File icon indicates that the detection belongs to a plain file, you can click on it to view the file in *Table View*.

For entire UI reference, Read *License Explorer*

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

3.3.3 How-To: Navigate the Package Explorer View

Display the Package Explorer view

Packages are uniquely identified using PURLs. A package URL is used to identify and locate a software package in a mostly universal and uniform way across programing languages, package managers, packaging conventions, tools, APIs and databases.

Once you have a SQLite file loaded into ScanCode Workbench,

Click on *Package Explorer* in the sidebar or open the `View` menu and select `Package Explorer`

Select a Package or Dependency

On the left pane, Packages are shown in a hierarchical manner as *Package Type* > *Package* > *Dependencies* Upon selecting a Package / Dependency, details of the entire entity is shown on the right.

For a Package, information like *PURL, namespace, Declared license expression, Dependencies table, etc* are shown.

For a Dependency, information like *Scope, Data file, Data source ID, etc* are shown.

Filters

You can filter the packages / dependencies by their data sources and/or dependency flags by selecting filter values in the top-left

Dependencies

When a package is selected, the dependencies for that package are shown in a table in the details pane showing Dependency information like *PURL, Scope, Data source ID, etc*

You can sort the table as per convenience.

You can click on the `Data file path` to see that file in the TableView

For entire UI reference, See *Package Explorer*

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

3.3.4 How-To: Navigate the Chart Summary View

Display the Chart Summary view

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

- Select a file or directory in the Tree View on the left.
- Click on *Chart Summary View* 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`, `Detected License expression`). 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.

For entire UI reference, Read *Chart Summary View*

Note

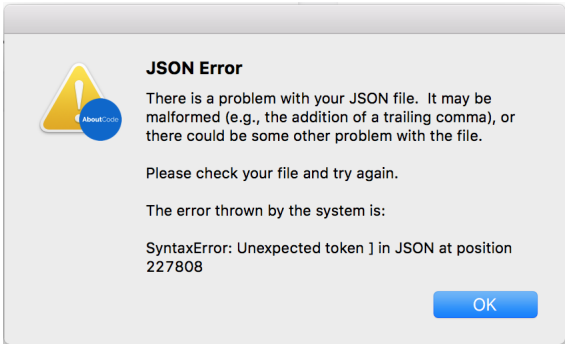
Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

3.4 Troubleshooting

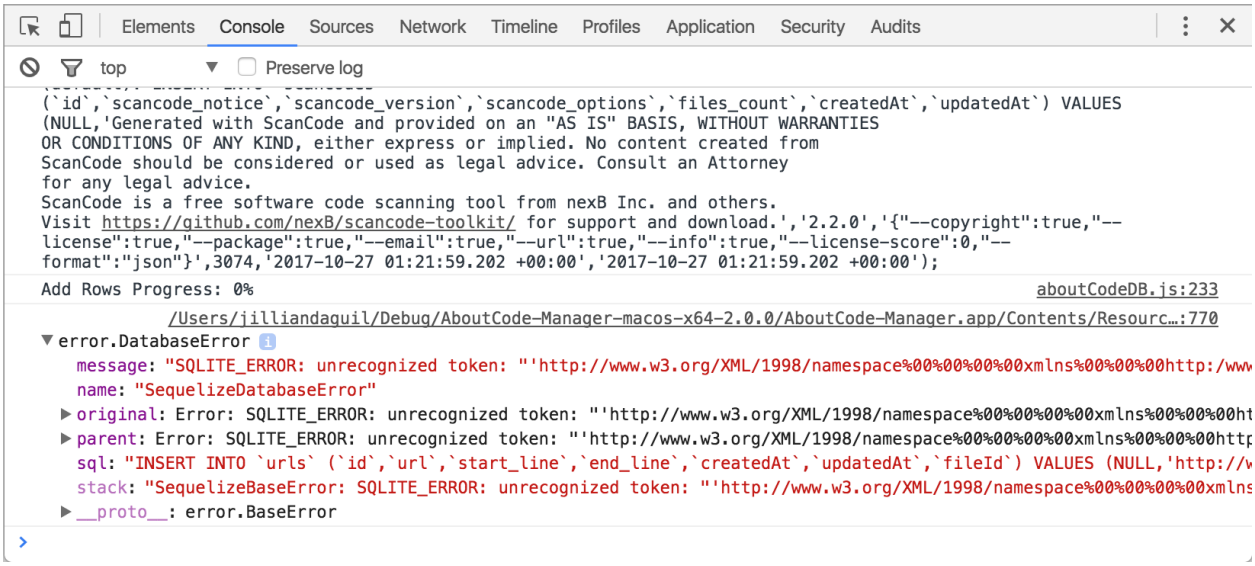
3.4.1 How-To: Check for Errors in the Developer Tools

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 by clicking `View -> Toggle Developer Tools`. From there, click the `Console` tab. Include the error that is logged in the issue in a code block or a file attachment.

You can Create an issue here.



CHAPTER

FOUR

UI REFERENCE

4.1 Directory Tree

An interactive directory tree is present on the left side of the application for resource-focused views. This allows the user to navigate the codebase structure.

If a directory is selected, only the information for 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.

4.1.1 Demo of filetree on different views:

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

4.2 Table View

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 preference is. Searching for specific clues (license names, copyrights, etc.) is also available in this view.

Note

The data in tableview is only available if corresponding options were enabled when performing the scan. Refer - All Available Options.

4.2.1 Column groups

Copyright

- copyright_statements
- copyright_holders
- copyright_authors
- copyright_start_line
- copyright_end_line

File

- type
- name
- extension
- size
- programming_language
- mime_type
- file_type
- is_binary
- is_text
- is_archive
- is_media
- is_source
- is_script

License

- detected_license_expression
- detected_license_expression_spdx
- percentage_of_license_text
- license_detections
- license_policy
- license_clues

Origin

- copyright_statements
- license_policy
- email
- url
- mime_type
- file_type
- programming_language

Package

- package_data_type
- package_data_name
- package_data_version
- package_data_extracted_license_statement
- package_data_declared_license_expression
- package_data_declared_license_expression_spdx
- package_data_primary_language
- for_packages

Misc

- scan_error

Preset Column Groups

The columns are grouped based on their nature such as Copyright Cols, Origin Cols, License Cols, etc. User can select one of these groups using the corresponding buttons available in the quick actions row.

View Custom set of columns

In addition to the preset groups, Users can select a custom set of columns by using the Custom Columns button found in the quick actions row.

Pin columns

Users can pin any column(s) to the left by dragging them through the column-header towards the left

Column wise filter & search

User can easily filter rows based on specific column values or use advanced search using and, or in combination with different search patterns like Contains, Starts with, etc.

Search in tableview

Use the Search bar in the top-right corner to perform searches within the specified directory scope and selected column fields (hidden columns are excluded from the search)

Go to License Explorer

User can go to specific licenses in the licenses explorer by the clickable links in License Detections column (Grouped under *License cols*). This will open the license explorer with the target license selected. Similarly, Path column in the File regions table in license details pane also provide links which navigate back to the Table view with that File path selected

Go to Package Explorer

User can go to specific package in the package explorer by the clickable links in `For Packages` column (Grouped under *Package cols*). This will open the package explorer with the target package selected. Similarly, Datafile paths in package/dependency details pane as well as dependency table also provide links which navigate back to the Table view with that File path selected

Pagination

Users can easily navigate through rows split into pages, with the flexibility to select a custom number of rows per page, tailoring the display to their convenience and preferences

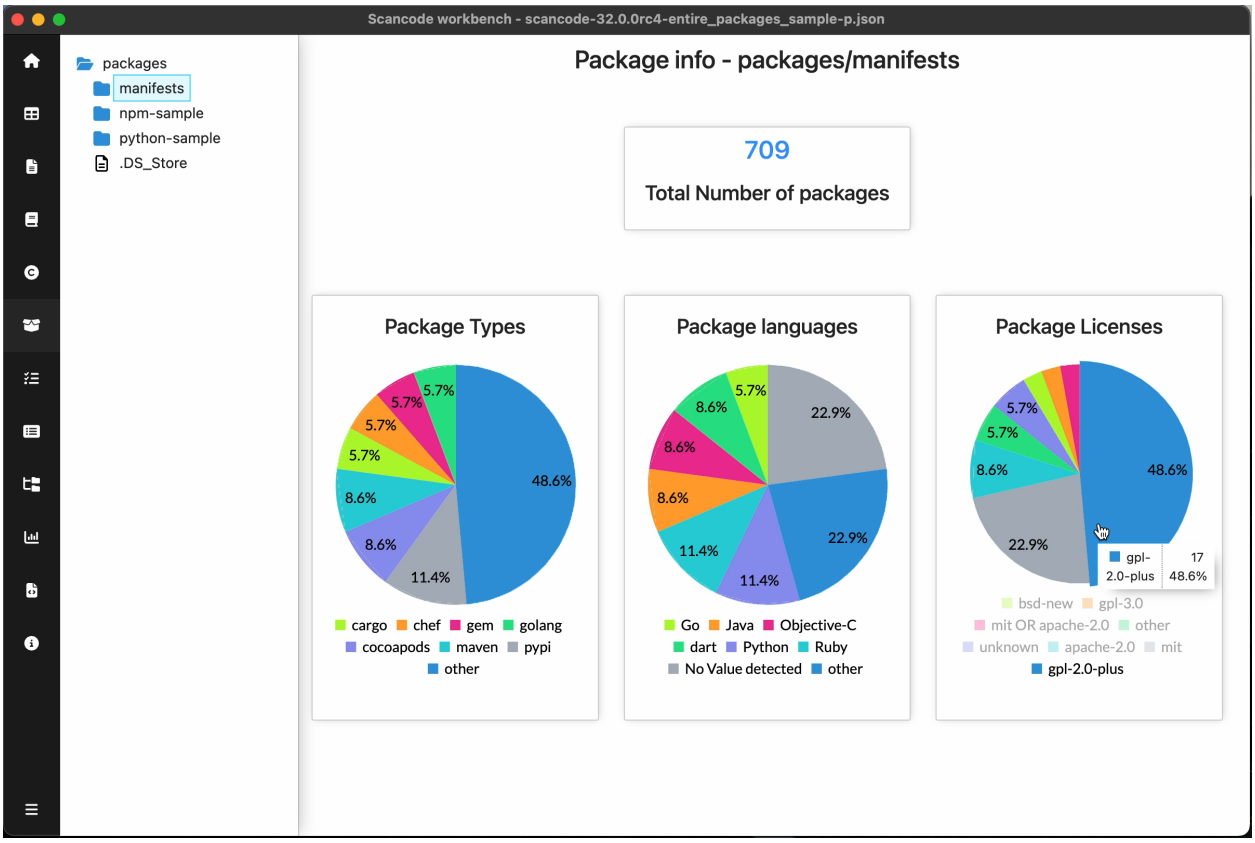
4.3 Pie charts

Pie charts give a quick overview of the distribution of values for a given attribute. ScanCode Workbench utilises c3js charts on the following views:

- *File Info Dashboard*
- *License Info Dashboard*
- *Copyright Info Dashboard*
- *Package Info Dashboard*
- *Dependency Info Dashboard*

4.3.1 Tooltip

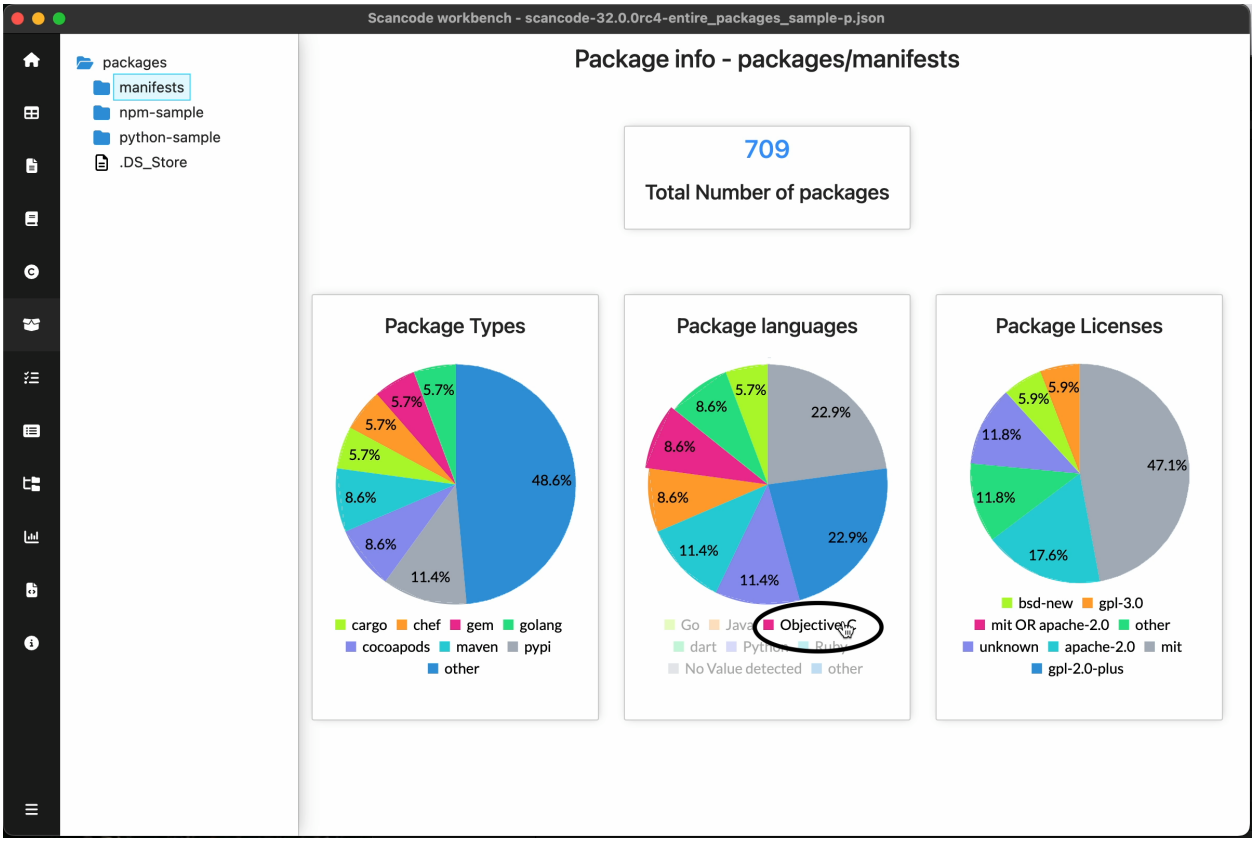
You can hover over the pie chart to see the exact value and percentage of the pie chart.



4.3.2 Legend items

> **Note**
>
> Legend items are not available on the *Copyright Info Dashboard* due to long copyright statements.

You can hover over the legend items to highlight the corresponding pie chart segment.



You can click on the legend items to toggle the visibility of corresponding pie chart segments.

Fig. 1: Click legend item to toggle segment visibility in Pie chart

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

4.4 File Info Dashboard

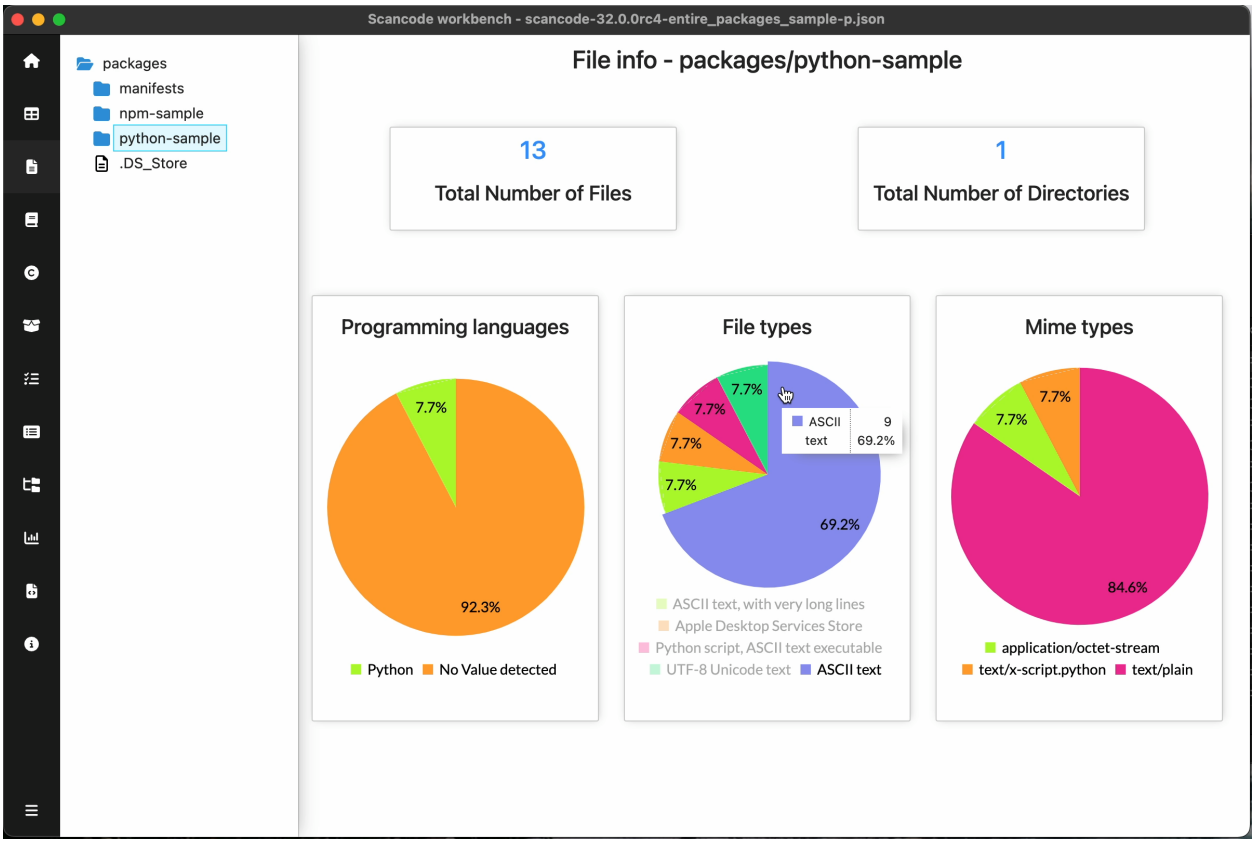
File Info Dashboard summarizes the file information in scan for the selected path. It is a good starting point to understand the file types and programming languages used in the project. Scan must have `--info` option enabled for the data required in File Info Dashboard.

Summary of files:

- Total number of Files (under the selected path)
- Total number of Directories (under the selected path)

- **Pie charts**
  - Programming languages
  - File Types
  - Mime Types

> **Note**
>
> Refer *Pie charts* for UI features in charts



> **Note**
>
> Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:
>
> - anglesharp.css.0.16.4-scan-results-without-text-referendces.json
> - aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
> - hazelcast-3293_v32.0.0rc3.json
> - scancode-32.0.0rc4_python_sample-wref.json
> - zjsonpatch-0.3.0.jar-v32rc4results.json

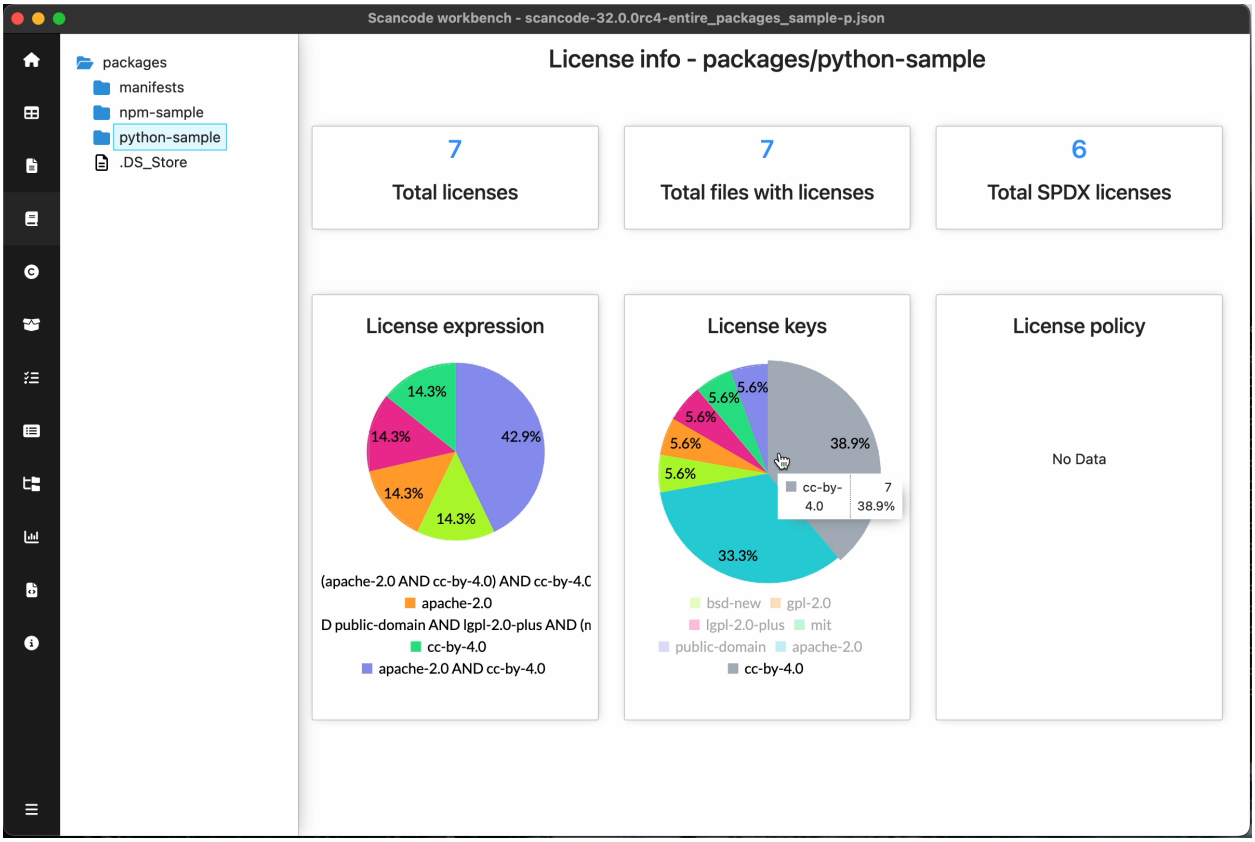
4.5 License Info Dashboard

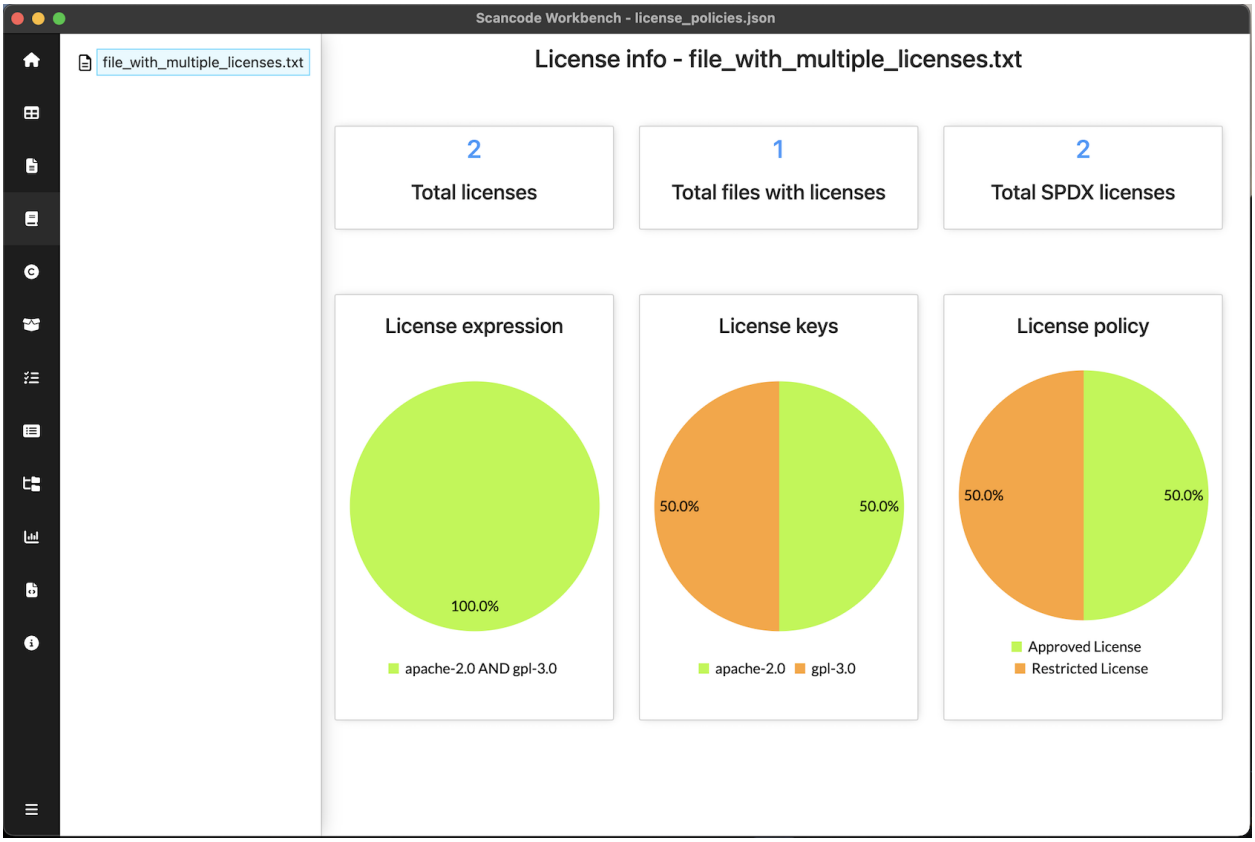
License info dashboard summarizes the license information detected for files in the scan under the selected path. It is a good starting point to understand the declaration of licenses in the project. Scan must have `--license` option enabled for the data required in License Info Dashboard.

Summary of Licenses:

- **Licenses**
  No. of unique licenses detected.
- **Files**
  No. of files (under the selected path) having at least one license detected.
- **SPDX Licenses**
  No. of unique SPDX detected.
- **Pie charts**
  - **License expressions**
    Distribution of unique license expressions indicating the number of files in which each license expression is detected.
  - **License keys**
    Distribution of unique license keys indicating the number of files in which each license key is detected.
  - **License policies**
    Distribution of unique license policies indicating the number of files in which each license policy is detected.

> **Note**
>
> Refer *Pie charts* for UI features in charts


Scancode workbench - scancode-32.0.0rc4-entire_packages_sample-p.json
packages
manifests
npm-sample
python-sample
.DS_Store
License info - packages/python-sample
7
Total licenses
7
Total files with licenses
6
Total SPDX licenses
License expression
14.3%
42.9%
14.3%
14.3%
14.3%
(apache-2.0 AND cc-by-4.0) AND cc-by-4.0
apache-2.0
D public-domain AND lgpl-2.0-plus AND (n
cc-by-4.0
apache-2.0 AND cc-by-4.0
License keys
5.6%
5.6%
5.6%
5.6%
5.6%
38.9%
33.3%
cc-by-4.0
7
38.9%
bsd-new
gpl-2.0
lgpl-2.0-plus
mit
public-domain
apache-2.0
cc-by-4.0
License policy
No Data


Scancode Workbench - license_policies.json
file_with_multiple_licenses.txt
License info - file_with_multiple_licenses.txt
2
Total licenses
1
Total files with licenses
2
Total SPDX licenses
License expression
100.0%
apache-2.0 AND gpl-3.0
License keys
50.0%
50.0%
apache-2.0
gpl-3.0
License policy
50.0%
50.0%
Approved License
Restricted License

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

4.6 Copyright Info Dashboard

Copyright info dashboard summarizes the copyright information detected in scan for files under the selected path. Scan must have `--copyright` option for Copyright Info Dashboard.

Summary of Copyrights:

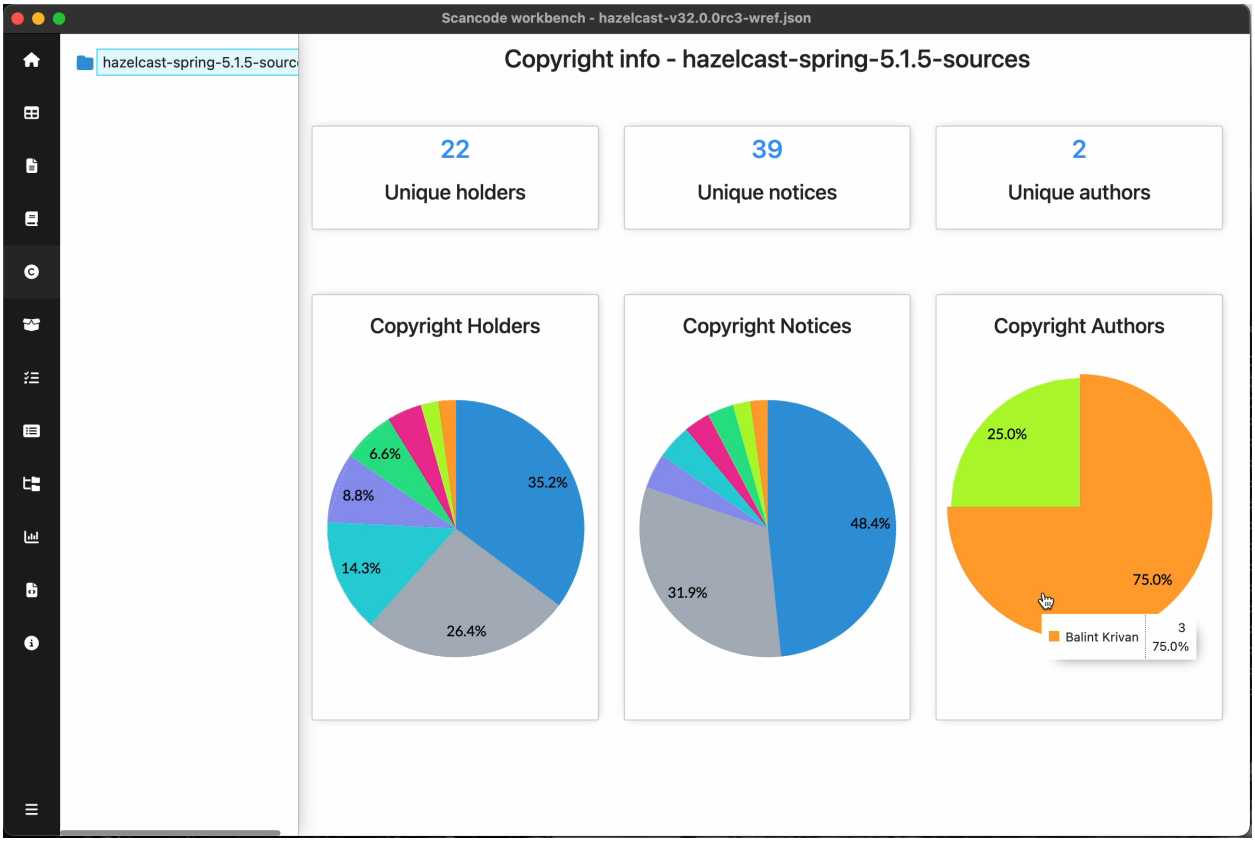
- **Unique holders**
  No. of unique copyright holders detected.
- **Unique notices**
  No. of unique copyright notices detected.
- **Unique authors**
  No. of unique copyright authors detected.
- **Pie charts**
  - **Copyright Holders**
    Distribution of unique copyright holders indicating the number of files in which each copyright holder was detected.
  - **Copyright Notices**
    Distribution of unique copyright notices indicating the number of files in which each copyright notice was detected.
  - **Copyright Authors**
    Distribution of unique copyright authors indicating the number of files in which copyrights were authored by each author.

Note

Refer *Pie charts* for UI features in charts

Note

Legend items are not available on the *Copyright Info Dashboard* due to long copyright statements.



Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

4.7 Package Info Dashboard

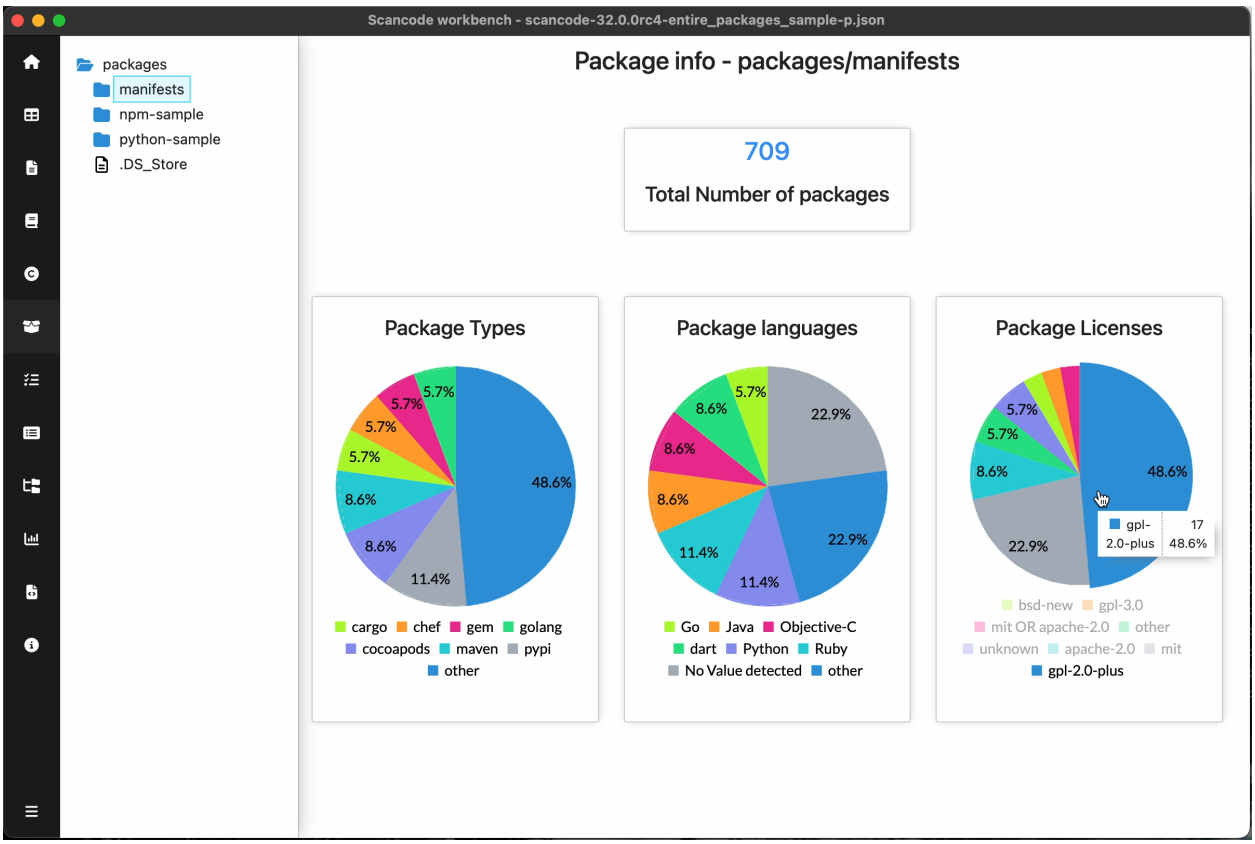
Package info dashboard summarizes the packages detected for files in the scan under the selected path. Scan must have `--package` option for License Info Dashboard

Summary of Packages:

- Total number of packages detected
- **Pie charts**
  - **Package Types**
    Distribution of unique package types detected. eg. `npm`, `cargo`, `pypi`, etc.
  - **Package Languages**
    Distribution of unique package languages detected. eg. `Javascript`, `Python`, `Ruby`, etc.

– **Package Licenses**
  Distribution of unique package licenses detected.

> **Note**
>
> Refer *Pie charts* for UI features in charts



> **Note**
>
> Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:
>
> - anglesharp.css.0.16.4-scan-results-without-text-referendces.json
> - aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
> - hazelcast-3293_v32.0.0rc3.json
> - scancode-32.0.0rc4_python_sample-wref.json
> - zjsonpatch-0.3.0.jar-v32rc4results.json

4.8 Dependency Info Dashboard

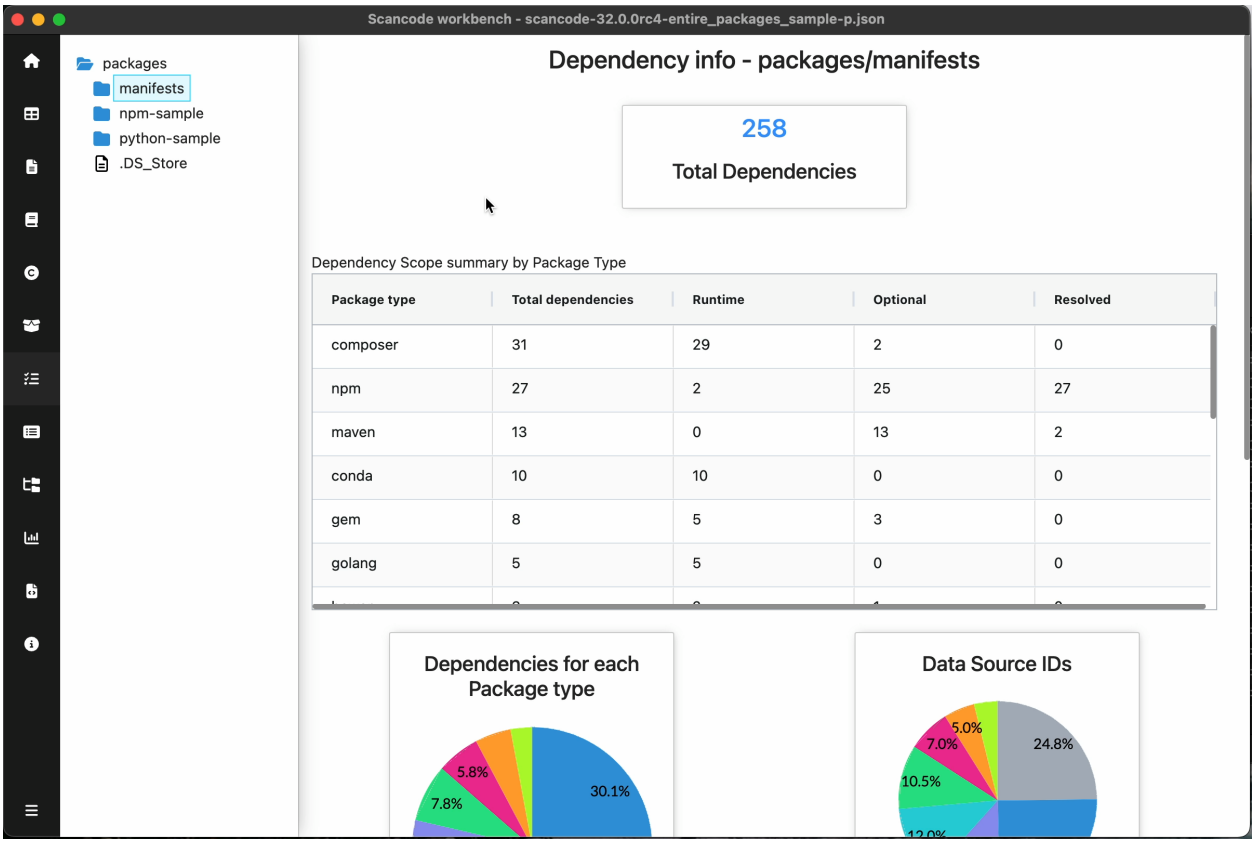
Dependency info dashboard summarizes the dependencies information detected in scan for files under the selected path. Scan must have `--package` option for Dependency Info Dashboard

Summary of Dependencys:

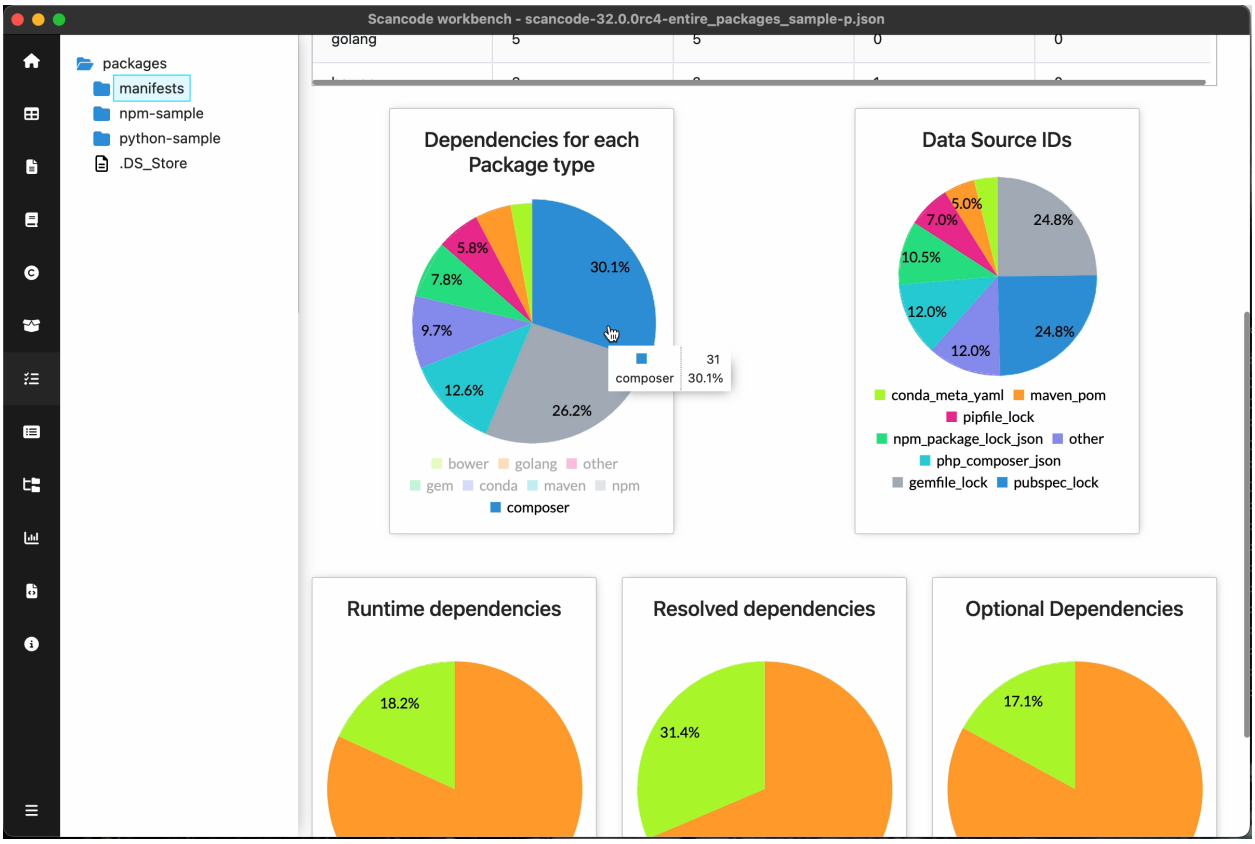
- Total number of Dependencies detected
- **Summary table**
  - Dependencies per Scope (Runtime, Optional, Pinned) for each Package Type
- **Pie charts**
  - Dependencies for each Package type
  - **Data source IDs**
    Distribution of Unique data source IDs. eg. `maven_pom`, `pipfile_lock`, etc
  - Runtime dependencies
  - Resolved dependencies
  - Optional dependencies

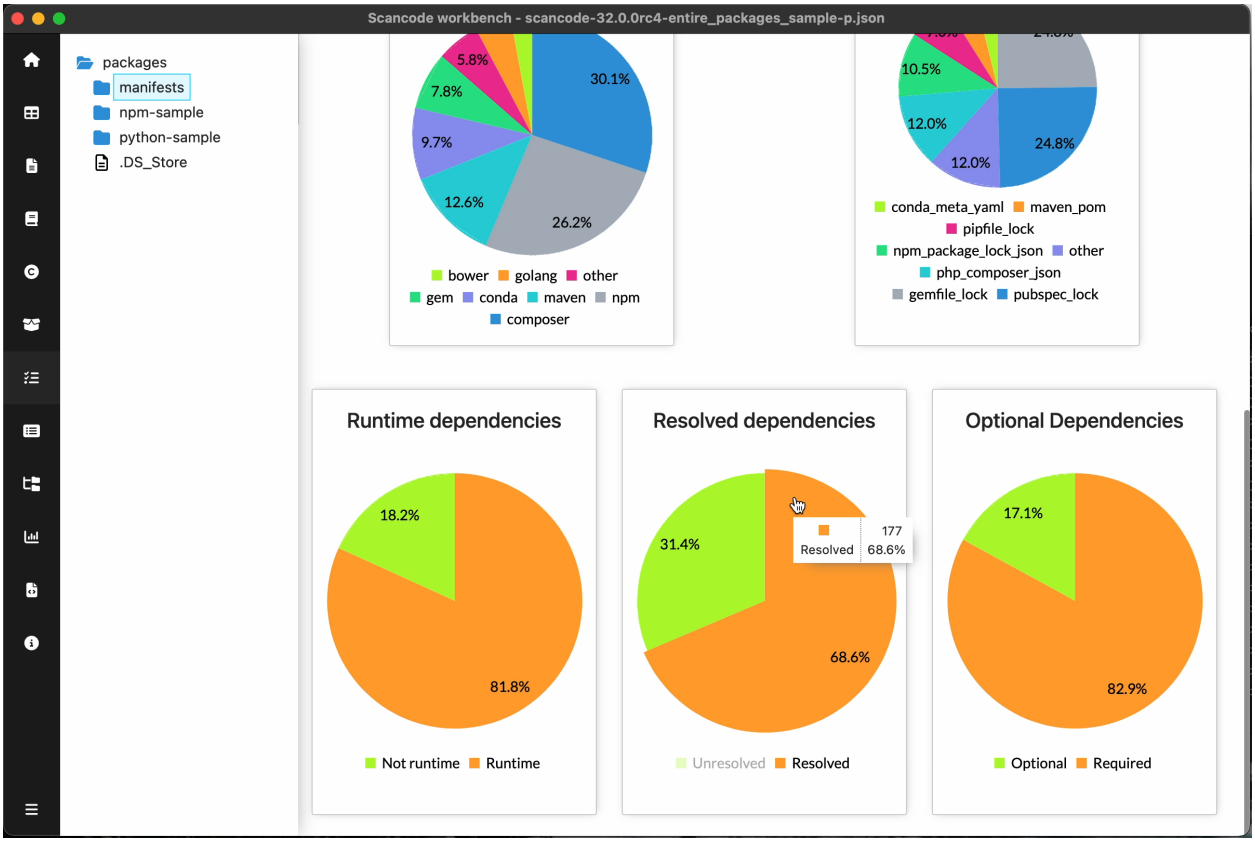
> **Note**
>
> Refer *Pie charts* for UI features in charts

4.8.1 Package type wise Scope summary



4.8.2 Charts


Scancode workbench - scancode-32.0.0rc4-entire_packages_sample-p.json
packages
manifests
npm-sample
python-sample
.DS_Store
golang 5 5 0 0
Dependencies for each Package type
30.1%
26.2%
12.6%
9.7%
7.8%
5.8%
composer 31 30.1%
bower golang other
gem conda maven npm
composer
Data Source IDs
24.8%
24.8%
12.0%
12.0%
10.5%
7.0%
5.0%
conda_meta_yaml maven_pom
pipfile_lock
npm_package_lock_json other
php_composer_json
gemfile_lock pubspec_lock
Runtime dependencies
18.2%
Resolved dependencies
31.4%
Optional Dependencies
17.1%



Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

4.9 License Explorer

Details of all the License detections and clues including Matches as well as files in which they were detected can be viewed using the License Explorer.

Refer - License Detection Updates

4.9.1 License Navigation Pane

User can scroll through the licenses and select particular license to see detailed information in the License details pane. Licenses are divided into two sections:

- License Detections
- License Clues

These sections' height & navigation pane's width can be adjusted as per convenience.

Search Licenses

You can search for any license detecttion / clue by typing in the search box on the top-left

Filter & Mark Licenses as Reviewed / Unreviewed

When reviewing large number of licenses over multiple sessions, You can tick the checkbox beside any of theselicense detections or clues to mark them as reviewed or unreviewed so that you can continue from where you left off.

Based on this review status, you can filter the licenses using filter options on the top

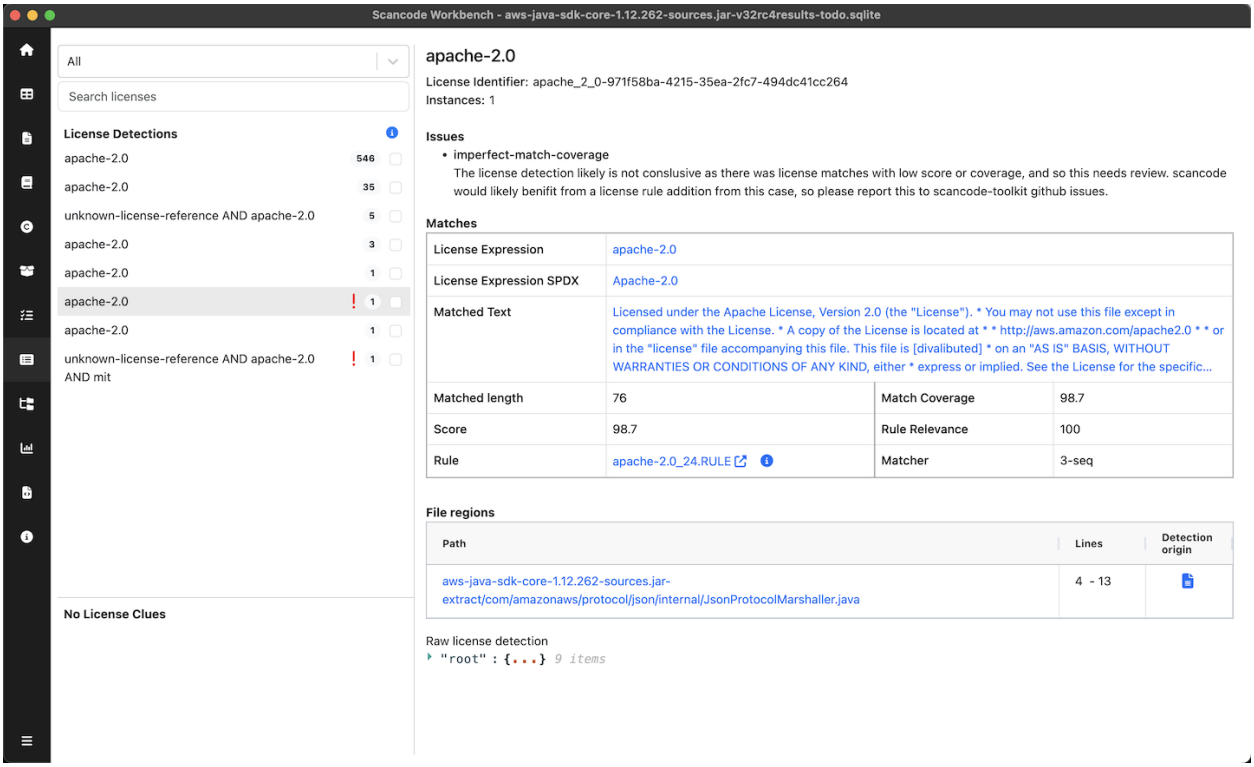
4.9.2 License Details Pane

For the selected license, you can view the details of matches resulting into the detection and the file regions where the license was detected.

- Title of details pane - License Expression
- Instances - No. of times license is detected
- Score - Clue's confidence about the license (Shown only for clues)

To-Do

Issues (if any) associated with license detections are listed with relevant details. License detections with issues are marked with a warning icon in the navigation pane.



Matches Table

User can view the match details resulting the selected license :

- `License expression` & `SPDX License expression`
- `Matched length`
- `Match Coverage`
- `Score`
- `Rule relevance`
- `Matcher`
- **`Matched Text`**

  User can click on the text to view a diff of Matched & Rule text

- `Rule` - User can click on the Rule to open the rule used byScanCode Toolkit in browser

To view more information about match rule, you can click on info button beside Rule name. You can click on the Matched text to see the complete text as well as its difference with Rule text.

> **Note**
>
> Matched text is available in the scan only when `--license-references` flag is enabled in scan.

File Regions Table

File regions table shows the files in which selected license was detected with following details

- `Path` - Path of file, User can click on it to view the specific file in TableView
- `Lines` - Specific lines in the file at which license was detected
- `Detection origin` - Type of origin (File or Package manifest)

You can click on the data file path to view that path in Tableview. Similarly, detection origin is a clickable icon. It determines the source of detection.

- Package icon indicates that the detection belongs to a package manifest, you can click on it to view the source package manifest in *Package Explorer*.
- File icon indicates that the detection belongs to a plain file, you can click on it to view the file in *Table View*.

> **Note**
>
> Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:
>
> - anglesharp.css.0.16.4-scan-results-without-text-referendces.json
> - aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
> - hazelcast-3293_v32.0.0rc3.json
> - scancode-32.0.0rc4_python_sample-wref.json
> - zjsonpatch-0.3.0.jar-v32rc4results.json

4.10 Package Explorer

Details of all the Packages & dependencies as well as files in which they were detected can be viewed using the Package Explorer.

Refer - Supported package manifests and package datafiles.

4.10.1 Package Navigation Pane

User can scroll through the packages & dependencies and select particular package to see detailed information in the Details pane. Packages and dependencies are organised in a hierarchical structure as shown below.

```
> Package Type
    > Package (PURL)
        > Dependencies
```

User can select different filters:

- Data sources - `maven_pom`, `go_mod`, `npm_package_lock_json`, etc
- Dependency flags - `Runtime`, `Pinned`, `Optional`

  Tip - You can click on the tag on the right side of the dependency to toggle that flag.

The navigation pane's width can also be adjusted as per convenience.

4.10.2 Package Details

User can see following information of the selected package:

- `Title of details pane`- PURL

  A package URL is used to identify and locate a software package in a mostly universal and uniform way across programing languages, package managers, packaging conventions, tools, APIs and databases.
- `Type`
- `Namespace`
- `Name`
- `Version`
- `Subpath`
- `Primary Language`
- `Extracted license statement`
- `Declared license expression`
- `Declared license expression SPDX`
- `Other license expression`
- `Other license expression SPDX`
- `Homepage URL`

Datafile paths

List of paths of datafiles in which the selected package/dependency was detected. User can click on the datafile paths to open that path in the Tableview the match details resulting the selected license.

Dependencies table

User can see all the dependencies of the selected package in the table, with following details:

- `Purl`

  A package URL is used to identify and locate a software package in a mostly universal and uniform way across programing languages, package managers, packaging conventions, tools, APIs and databases.

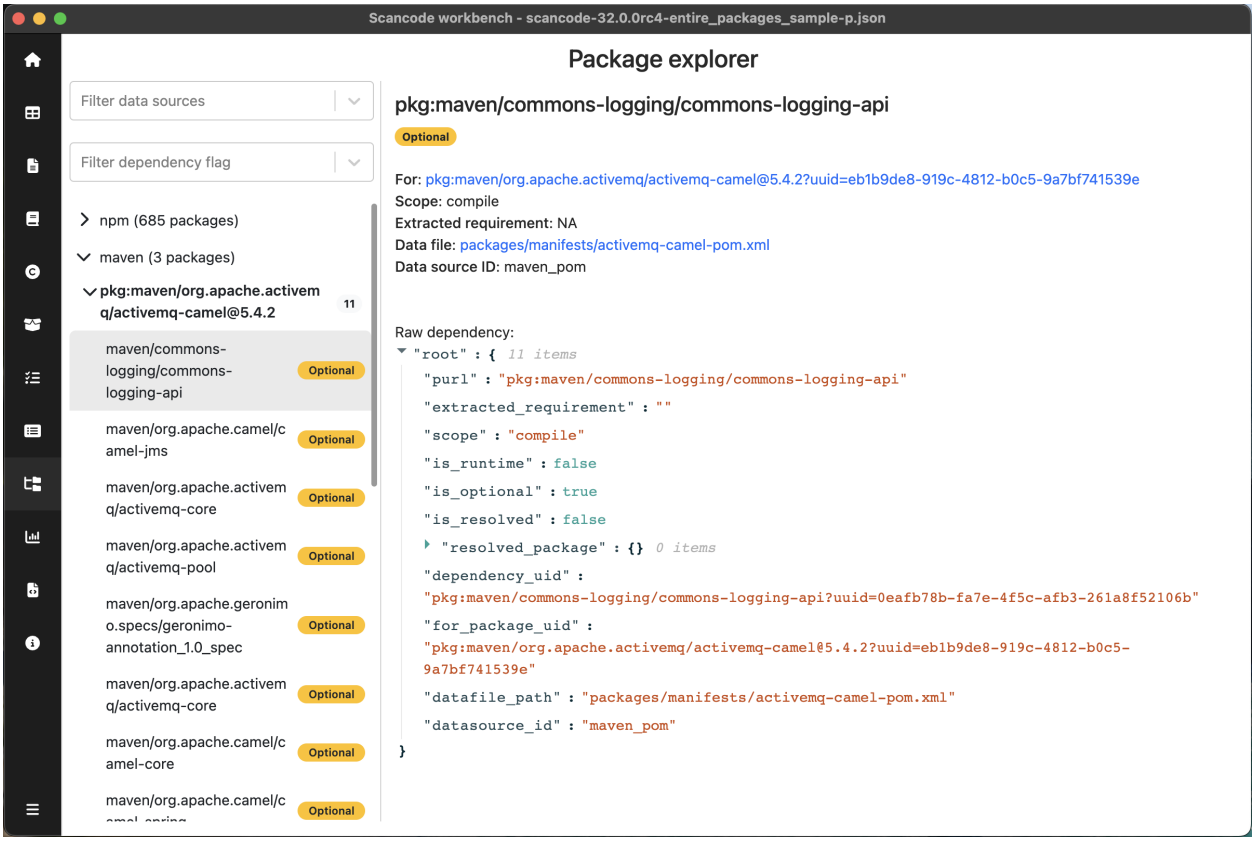
- `Scope`
- `Resolved`
- `Runtime`
- `Optional`
- `Data source ID`
- `Data file`
- `Extracted requirement`

4.10.3 Dependency Details

User can see following information of the selected dependency:

- `For` - PURL of the package for which this dependency was detected
- `Scope`
- `Extracted requirement`
- `Data file` - Clickable datafile path to open the datafile in the Tableview
- `Data source ID`

The Data file property is a clickable link that opens the Tableview with that file selected.



Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

4.11 Chart Summary View

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, i.e. the ScanCode Toolkit detections, 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.

Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

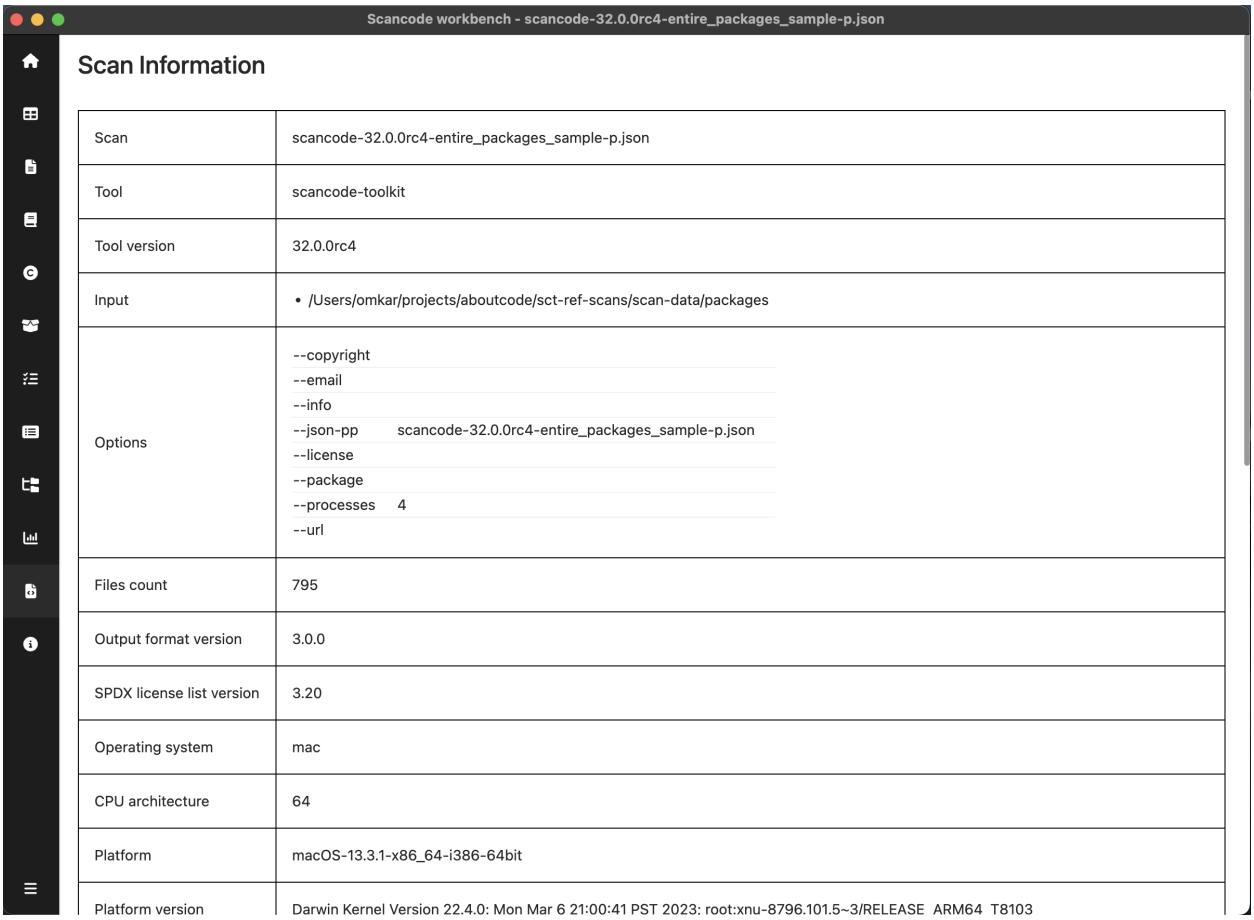
- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

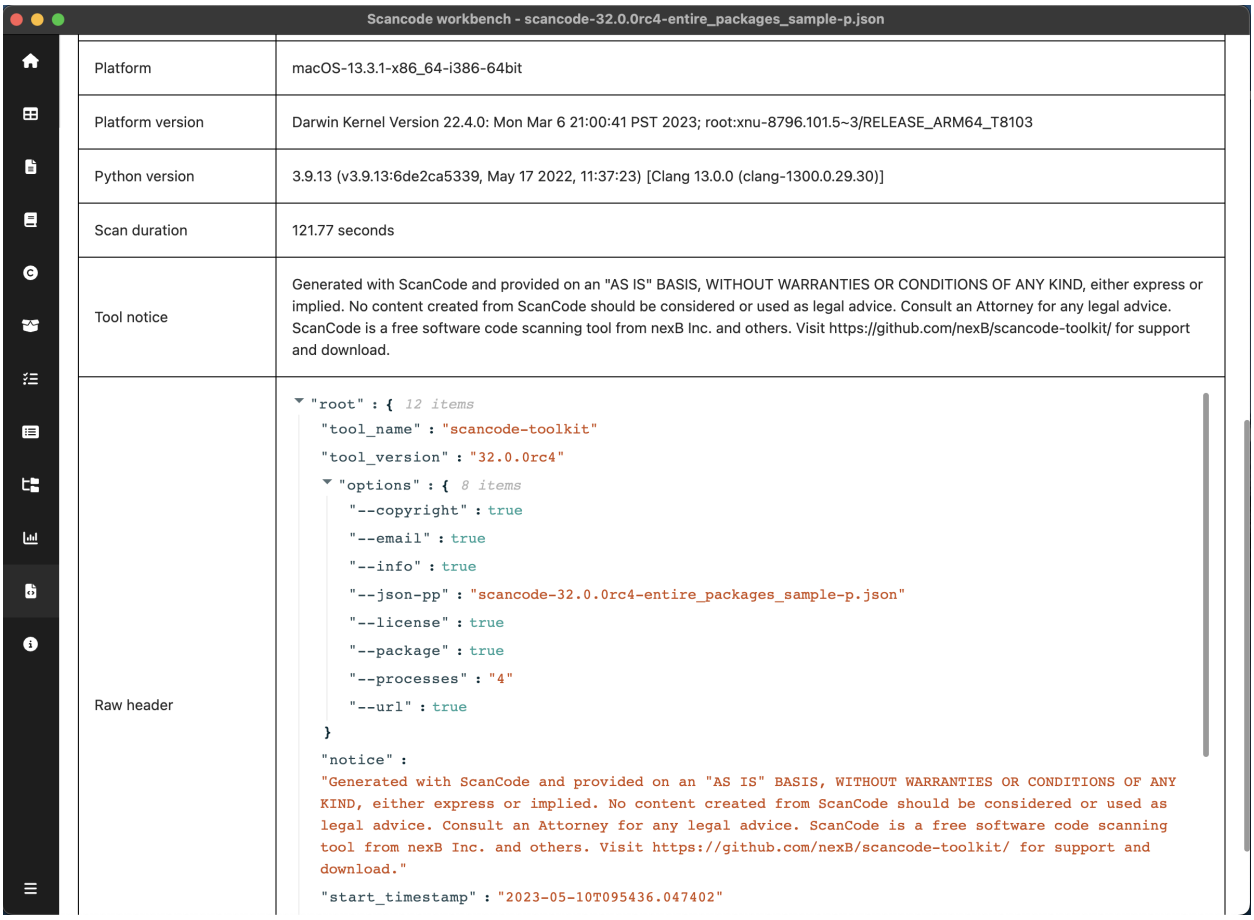
4.12 Scan Info

Scan Info provides information about the scan available in `headers` in the scan. It provides information about the environment & configuration on host machine when performing the scan. You can also see the raw header JSON at the bottom

Fields:

- `Tool` - Name of the tool which generated this scan
- `Tool version` - Version of the tool which generated this scan
- `Input` - Input path on which scan was run
- `Scan Options` - Basic Options.
- `Files count` - Total number of files scanned
- `Output format version` - Used to track compatibility with current workbench application
- `SPDX license list version` - Version of SPDX license list used
- `Operating system` - Operating system on which scan was run
- `CPU architecture` - CPU architecture on which scan was run
- `Platform` - Platform on which scan was run
- `Platform version` - Platform version on which scan was run
- `Python version`
- `Scan duration`
- `Tool notice` - Notice of the tool which generated this scan
- `Errors` (if any)
- `Raw header`


Scancode workbench - scancode-32.0.0rc4-entire_packages_sample-p.json
Scan Information
Scan
scancode-32.0.0rc4-entire_packages_sample-p.json
Tool
scancode-toolkit
Tool version
32.0.0rc4
Input
• /Users/omkar/projects/aboutcode/sct-ref-scans/scan-data/packages
Options
--copyright
--email
--info
--json-pp scancode-32.0.0rc4-entire_packages_sample-p.json
--license
--package
--processes 4
--url
Files count
795
Output format version
3.0.0
SPDX license list version
3.20
Operating system
mac
CPU architecture
64
Platform
macOS-13.3.1-x86_64-i386-64bit
Platform version
Darwin Kernel Version 22.4.0: Mon Mar 6 21:00:41 PST 2023: root:xnu-8796.101.5~3/RELEASE_ARM64_T8103



Note

Refer the titlebar to see the name of the sample scan used in the screenshot. Sample scans:

- anglesharp.css.0.16.4-scan-results-without-text-referendces.json
- aws-java-sdk-core-1.12.262-sources.jar-v32rc4results-todo.json
- hazelcast-3293_v32.0.0rc3.json
- scancode-32.0.0rc4_python_sample-wref.json
- zjsonpatch-0.3.0.jar-v32rc4results.json

CHAPTER

FIVE

CONTRIBUTE

5.1 Building

5.1.1 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/aboutcode-org/scancode-workbench.git

# Go into the repository
cd scancode-workbench

# Install dependencies and run the app (Native dependencies are handled automatically)
npm install

# Run the app
npm start
```

5.1.2 Building Requirements

Linux

- Python v3.9 or later
- Node.js 16.x or later
- npm v8.x or later

Note

For CentOS (or linux distros without the new libstdc++), follow these steps:

- Install the new libstdc++ library:

```
yum provides libstdc++
```

- Update LD_LIBRARY_PATH:

```
export LD_LIBRARY_PATH="/usr/local/lib64/:$LD_LIBRARY_PATH"
```

- Run the application with `no-sandbox` option:

```
./ScanCode-Workbench-4.0.3-linux-x64/ScanCode-Workbench-4.0.3 --no-sandbox
```

MacOS

- Python v3.9 or later
- Node.js 16.x or later
- npm v8.x or later
- **Command Line Tools for Xcode**
  Install using:

```
xcode-select --install
```

Windows

- Python v3.9 or later
  - Make sure your Python path is set. To verify, open a command prompt and see the python version:

```
python --version
```

- Node.js v16.x or later
- npm v8.x or later

5.1.3 Release Instructions

ScanCode Workbench release is built using electron-forge. You can build the application for your platform using following command:

```
npm run publish
```

You can find the executible `ScanCode-Workbench-<version>` inside `out/ScanCode-Workbench-<version>-<os>-<arch>` and a distributable archive in `dist/` directory.

Archives are built as:

- `tar.gz` - Linux / MacOS
- `.zip` - Windows

> **Note**
>
> Due to usage of native modules, a build must be done on target platform only. For example, a linux build must be done on linux machine only.

5.1.4 Building Documentation

Create python environment, make docs

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

# Go into the docs directory
cd docs/
```

(continues on next page)

(continued from previous page)

```
# Setup virtual environment for python dependencies
python -m venv venv
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt

# Build Documentation
make html

# Run Documentation server
make docs
```

5.2 Testing

Test ABCM functionality using:

```
$ npm test
```

Tests are written in folder `tests/` utilising the sample scans in `tests/test-scans`

- `file.test.ts` - Test assertions
- `file.test.data.ts` - Data samples per test file

5.3 Reporting Issues

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

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

5.4 Contributing Code

- Contributions comes as bugs/questions/issues and as pull requests.
- Source code and runtime data are in the `src/` directory.
- Test code and test data are in the `tests/` directory.
- We use DCO signoff in commit messages, like Linux does.

5.5 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://matrix.to/#/#aboutcode-org_discuss:gitter.im

CHAPTER

SIX

LICENSE

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

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- search

INDEX