Setting up the CodeQL CLI

In this article

To get started with the CodeQL CLI, you need to download and set up the CLI so that it can access the tools and libraries required to create and analyze databases.

> GitHub CodeQL is licensed on a per-user basis upon installation. You can use CodeQL only for certain tasks under the license restrictions. For more information, see "About the CodeQL CLI." If you have a GitHub Advanced Security license, you can use CodeQL for automated analysis, continuous integration, and continuous delivery. For more information, see "About GitHub Advanced Security."

Setting up the CodeQL CLI

To run CodeQL commands, you need to set up the CLI so that it can access the tools, queries, and libraries required to create and analyze databases.

The CodeQL CLI can be set up to support many different use cases and directory structures. To get started quickly, we recommend adopting a relatively simple setup, as outlined in the steps below.

If you plan to use the CodeQL CLI for security research or to test or contribute queries, you may want a more advanced setup of CodeQL CLI. For more information, see "Advanced setup of the CodeQL CLI."

If you are setting up the CodeQL CLI in your CI system, you need to make the full contents of the CodeQL CLI bundle available to every CI server that you want to run CodeQL code scanning analysis on. For example, you might configure each server to copy the bundle from a central, internal location and extract it. Alternatively, you could use the REST API to get the bundle directly from GitHub, ensuring that you benefit from the latest improvements to queries. For more information, see "Releases" in the REST API documentation.

If you are using macOS on Apple Silicon (for example, Apple M1), ensure that the Xcode command-line developer tools and Rosetta 2 are installed.

> **Note:** The CodeQL CLI is currently not compatible with non-glibc Linux distributions such as (muslc-based) Alpine Linux.

1. Download the CodeQL CLI zip package

The CodeQL CLI download package is a zip archive containing tools, scripts, and various CodeQL-specific files. If you don’t have a GitHub Enterprise license then, by downloading this archive, you are agreeing to the GitHub CodeQL Terms and Conditions.

You should download the CodeQL bundle from https://github.com/github/codeql-action/releases. The bundle contains:

- CodeQL CLI product
- A compatible version of the queries and libraries from https://github.com/github/codeql
- Precompiled versions of all the queries included in the bundle

You should always use the CodeQL bundle as this ensures compatibility and also gives much better performance than a separate download of the CodeQL CLI and checkout of the CodeQL queries. If you will only be running the CLI on one specific platform, download the appropriate `codeql-bundle-PLATFORM.tar.gz` file. Alternatively, you can download `codeql-bundle.tar.gz`, which contains the CLI for all supported platforms.

> **Note:** The CodeQL package management functionality, including CodeQL packs, is currently in beta and subject to change.

Download information for macOS "Catalina" (or newer) users

From macOS version 10.15 ("Catalina") onwards you need to ensure that your web browser does not automatically extract zip files. If you use Safari, complete the following steps before downloading the CodeQL CLI zip archive:

1. Open Safari.

2. From the Safari menu, select **Preferences...** or **Settings...** (version 13 "Ventura" onwards).

3. Click the **General** Tab.

4. Ensure the check-box labeled **Open "safe" files after downloading** is unchecked.

2. Extract the zip archive

For Linux, Windows, and macOS users (version 10.14 "Mojave", and earlier) simply extract the zip archive.

Extraction information for macOS "Catalina" (or newer) users

macOS "Catalina", "Big Sur", "Monterey", or "Ventura" users should run the following commands in the Terminal, where `${extraction-root}` is the path to the directory where you will extract the CodeQL CLI zip archive:

1. `mv ~/Downloads/codeql\*.zip ${extraction-root}`

2. `cd ${extraction-root}`

3. `/usr/bin/xattr -c codeql\*.zip`

4. `unzip codeql\*.zip`

3. Launch `codeql`

Once extracted, you can run CodeQL processes by running the `codeql` executable in a couple of ways:

- By executing `<extraction-root>/codeql/codeql`, where `<extraction-root>` is the folder where you extracted the CodeQL CLI package.
- By adding `<extraction-root>/codeql` to your `PATH`, so that you can run the executable as just `codeql`.

At this point, you can execute CodeQL commands. For a full list of the CodeQL CLI commands, see "CodeQL CLI commands manual."

> **Note:** If you add `codeql` to your `PATH`, it can be accessed by CodeQL for Visual Studio Code to compile and run queries. For more information about configuring VS Code to access the CodeQL CLI, see "Setting up CodeQL in Visual Studio Code."

Testing the CodeQL CLI configuration

After you extract the CodeQL CLI bundle, you can run the following command to verify that the CLI is correctly configured to create and analyze databases:

- `codeql resolve qlpacks` if `/<extraction-root>/codeql` is on the `PATH`.
- `/<extraction-root>/codeql/codeql resolve qlpacks` otherwise.

You should see output similar to the extract below, if successful:

```
codeql/cpp-all (/<extraction-root>/qlpacks/codeql/cpp-all/<version>)
codeql/cpp-examples (/<extraction-root>/qlpacks/codeql/cpp-examples/<version>)
codeql/cpp-queries (/<extraction-root>/qlpacks/codeql/cpp-queries/<version>)
codeql/csharp-all (/<extraction-root>/qlpacks/codeql/charp-all/<version>)
codeql/csharp-examples (/<extraction-root>/qlpacks/codeql/charp-examples/<version>)
codeql/csharp-queries (/<extraction-root>/qlpacks/codeql/charp-queries/<version>)
codeql/java-all (/<extraction-root>/qlpacks/codeql/java-all/<version>)
codeql/java-examples (/<extraction-root>/qlpacks/codeql/java-examples/<version>)
codeql/java-queries (/<extraction-root>/qlpacks/codeql/java-queries/<version>)
codeql/javascript-all (/<extraction-root>/qlpacks/codeql/javascript-all/<version>)
codeql/javascript-examples (/<extraction-root>/qlpacks/codeql/javascript-examples/<version>)
codeql/javascript-queries (/<extraction-root>/qlpacks/codeql/javascript-queries/<version>)
codeql/python-all (/<extraction-root>/qlpacks/codeql/python-all/<version>)
codeql/python-examples (/<extraction-root>/qlpacks/codeql/python-examples/<version>)
codeql/python-queries (/<extraction-root>/qlpacks/codeql/python-queries/<version>)
codeql/ruby-all (/<extraction-root>/qlpacks/codeql/ruby-all/<version>)
codeql/ruby-examples (/<extraction-root>/qlpacks/codeql/ruby-examples/<version>)
codeql/ruby-queries (/<extraction-root>/qlpacks/codeql/ruby-queries/<version>)
...
```

You should check that the output contains the expected languages and also that the directory location for the qlpack files is correct. The location should be within the extracted CodeQL CLI bundle, shown in the earlier example as `<extraction root>`. If the CodeQL CLI is unable to locate the qlpacks for the expected languages, check that you downloaded the CodeQL bundle and not a standalone copy of the CodeQL CLI.

You can also run `codeql resolve languages` to show which languages are available for database creation. This will list the languages supported by default in your CodeQL CLI package.

Optionally, you can download some CodeQL packs containing pre-compiled queries you would like to run. For more information, see "Customizing analysis with CodeQL packs."

Next steps

To learn how to prepare your code to be analyzed by the CodeQL CLI, see "Preparing your code for CodeQL analysis."