resolve extractor

In this article

Synopsis

Description

Options

[Deep plumbing] Determine the extractor pack to use for a given language.

> 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."

> This content describes the most recent release of the CodeQL CLI. For more information about this release, see https://github.com/github/codeql-cli-binaries/releases.
>
> To see details of the options available for this command in an earlier release, run the command with the `--help` option in your terminal.

Synopsis 🔗

Shell

```
codeql resolve extractor --language=<lang> <options>...
```

Description 🔗

[Deep plumbing] Determine the extractor pack to use for a given language.

Options 🔗

Primary Options 🔗

`-l, --language=<lang>` 🔗

[Mandatory] The name of the extractor to locate.

`--search-path=<dir>[:<dir>...]` 🔗

A list of directories under which extractor packs may be found. The directories can either be the extractor packs themselves or directories that contain extractors as immediate

subdirectories.

If the path contains multiple directory trees, their order defines precedence between them: if the target language is matched in more than one of the directory trees, the one given first wins.

The extractors bundled with the CodeQL toolchain itself will always be found, but if you need to use separately distributed extractors you need to give this option (or, better yet, set up `--search-path` in a per-user configuration file).

(Note: On Windows the path separator is `;`).

`--just-check` 🔗

Don't print any output, but exit with code 0 if the extractor is found, and code 1 otherwise.

`--format=<fmt>` 🔗

Select output format. Choices include:

`text` *(default)*: Print the path to the found extractor pack to standard output.

`json` : Print the path to the found extractor pack as a JSON string.

`betterjson` : Print details about the found extractor pack as a JSON string.

Common options 🔗

`-h, --help` 🔗

Show this help text.

`-J=<opt>` 🔗

[Advanced] Give option to the JVM running the command.

(Beware that options containing spaces will not be handled correctly.)

`-v, --verbose` 🔗

Incrementally increase the number of progress messages printed.

`-q, --quiet` 🔗

Incrementally decrease the number of progress messages printed.

`--verbosity=<level>` 🔗

[Advanced] Explicitly set the verbosity level to one of errors, warnings, progress, progress+, progress++, progress+++. Overrides `-v` and `-q` .

`--logdir=<dir>` 🔗

[Advanced] Write detailed logs to one or more files in the given directory, with generated names that include timestamps and the name of the running subcommand.

(To write a log file with a name you have full control over, instead give `--log-to-stderr` and redirect stderr as desired.)