database print-baseline

In this article

Synopsis

Description

Options

[Plumbing] Print a summary of the baseline lines of code seen.

> 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 database print-baseline <options>... -- <database>
```

Description 🔗

[Plumbing] Print a summary of the baseline lines of code seen.

This command will print to standard out the baseline lines of code seen within the source root specified at codeql database init time for each language present in the database.

The baseline is an estimate of the non-empty, non-comment lines of code in a database. This count is different from the lines of code counted by CodeQL metrics queries, which only counts code that is passed to the CodeQL evaluator. In some cases, the baseline count may be lower than the count in metrics queries since metrics queries may include external files that are passed to the evaluator, but are not included in the source root.

Options 🔗

Primary Options 🔗

`<database>` 🔗

[Mandatory] Path to the CodeQL database under construction. This must have been prepared for extraction with [codeql database init](#).

If the `--db-cluster` option is given, this is not a database itself, but a directory that *contains* databases, and all of those databases will be processed together.

`--[no-]db-cluster` 🔗

Indicates that the directory given on the command line is not a database itself, but a directory that *contains* one or more databases under construction. Those databases will be processed together.

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.)