

database finalize

In this article

- Synopsis
- Description
- Options


[Plumbing] Final steps in database creation.

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 finalize [--dbscheme=<file>] [--threads=<num>] [--ram=<MB>] [--mode=<mode>] <options>... -- <database>
```


Description

[Plumbing] Final steps in database creation.

Finalize a database that was created with [codeql database init](#) and subsequently seeded with analysis data using [codeql database trace-command](#). This needs to happen before the new database can be queried.

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.

`--additional-dbs=<database>[:<database>...]`

[Advanced] Path to additional CodeQL databases under construction. These will not themselves be finalized, but the data from them will be included in the finalized database being created. This is an advanced option that may not have the desired effect in all cases. For more information, please refer to the documentation of [codeql database import](#).

If the `--db-cluster` option is given, it is expected that these will be database clusters rather than individual CodeQL databases.

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

`--no-cleanup`

[Advanced] Suppress all database cleanup after finalization. Useful for debugging purposes.

`--no-pre-finalize`

[Advanced] Skip any pre-finalize script specified by the active CodeQL extractor.

`--[no-]skip-empty`

[Advanced] Output a warning instead of failing if a database is empty because no source code was seen during the build. The empty database will be left unfinalized.

Options for controlling the TRAP import operation

`-S, --dbscheme=<file>`

[Advanced] Override the auto-detected dbscheme definition that the TRAP files are assumed to adhere to. Normally, this is taken from the database's extractor.

`-j, --threads=<num>`

Use this many threads for the import operation.

Defaults to 1. You can pass 0 to use one thread per core on the machine, or *-N* to leave *N* cores unused (except still use at least one thread).

`-M, --ram=<MB>`

Use this much memory for the import operation.

Low-level dataset cleanup options

--max-disk-cache=<MB> [🔗](#)

Set the maximum amount of space that the disk cache for intermediate query results can use.

If this size is not configured explicitly, the evaluator will try to use a "reasonable" amount of cache space, based on the size of the dataset and the complexity of the queries. Explicitly setting a higher limit than this default usage will enable additional caching which can speed up later queries.

--min-disk-free=<MB> [🔗](#)

[Advanced] Set target amount of free space on file system.

If `--max-disk-cache` is not given, the evaluator will try hard to curtail disk cache usage if the free space on the file system drops below this value.

--min-disk-free-pct=<pct> [🔗](#)

[Advanced] Set target fraction of free space on file system.

If `--max-disk-cache` is not given, the evaluator will try hard to curtail disk cache usage if the free space on the file system drops below this percentage.

-m, --mode=<mode> [🔗](#)

Select how aggressively to trim the cache. Choices include:

`clear` : Remove the entire cache, trimming down to the state of a freshly extracted dataset

`trim` (*default*): Trim everything except explicitly "cached" predicates.

`fit` : Simply make sure the defined size limits for the disk cache are observed, deleting as many intermediates as necessary.

--cleanup-upgrade-backups [🔗](#)

Delete any backup directories resulting from database upgrades.

Options for checking imported TRAP [🔗](#)

--[no-]check-undefined-labels [🔗](#)

[Advanced] Report errors for undefined labels.

--[no-]check-unused-labels [🔗](#)

[Advanced] Report errors for unused labels.

--[no-]check-repeated-labels [🔗](#)

[Advanced] Report errors for repeated labels.

--[no-]check-redefined-labels [🔗](#)

[Advanced] Report errors for redefined labels.

--[no-]check-use-before-definition [↗](#)

[Advanced] Report errors for labels used before they're defined.

--[no-]fail-on-trap-errors [↗](#)

[Advanced] Exit non-zero if an error occurs during trap import.

--[no-]include-location-in-star [↗](#)

[Advanced] Construct entity IDs that encode the location in the TRAP file they came from. Can be useful for debugging of TRAP generators, but takes up a lot of space in the dataset.

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.)

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)