

database import

In this article

Synopsis

Description

Options

[Advanced] [Plumbing] Import unfinalized database(s) into another unfinalized database.

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

Description

[Advanced] [Plumbing] Import unfinalized database(s) into another unfinalized database.

The result of this command is that the target database (the one in the *first* argument) will be augmented with the data from all the other databases passed. In particular, TRAP files from the other databases will be imported and sources in them will be copied.

Note that this command will probably not have the desired effect in most cases. In particular, the resulting database may not correctly track dataflow between the partial databases that were combined. It is only intended to be used in certain advanced scenarios involving distributed build systems where special care has been taken in how the build was separated in order to ensure that the resulting final database is meaningful.

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.

<additionalDbs>... [↗](#)

[Mandatory] Paths to the unfinished database(s) that should imported into the first database.

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

--[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.

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.

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](#)