

# resolve metadata

## In this article

- Synopsis
- Description
- Options

[Deep plumbing] Resolve and return the key-value metadata pairs from a query source file.

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 metadata <options>... -- <file>
```

## Description

[Deep plumbing] Resolve and return the key-value metadata pairs from a query source file.

## Options

### Primary Options

`<file>` 


[Mandatory] Query source file from which to extract metadata.

`--format=<fmt>` 

This option is ignored for future compatibility. Pass `json` here to be sure to get the

current JSON output format even in a future version.

## 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](#)