

Specifying command options in a CodeQL configuration file

In this article

About specifying command options in a CodeQL configuration file

Using a CodeQL configuration file

You can save default command options in a CodeQL configuration 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](#)."

About specifying command options in a CodeQL configuration file [↗](#)

You can save default or frequently used options for your commands in a per-user configuration file.

You can specify CodeQL CLI command options in two ways:

- Directly in the command line, using the appropriate flag.
- In a configuration (or `config`) file that CodeQL scans for relevant options each time a command is executed.

For options that are likely to change each time you execute a command, specifying the value on the command line is the most convenient way of passing the information to CodeQL. Saving options in a `config` file is a good way to specify options you use frequently. It's also a good way to add custom CodeQL packs that you use regularly to your search path.

Using a CodeQL configuration file [↗](#)

You need to save the `config` file under your home (Linux and macOS) or user profile (Windows) directory in the `.config/codeql/` subdirectory. For example, `$HOME/.config/codeql/config`.

The syntax for specifying options is as follows:

```
<command> <subcommand> <option> <value>
```

To apply the same options to more than one command you can:

- Omit the `<subcommand>`, which will specify the option for every `<subcommand>` to

which it's relevant.

- Omit both `<command>` and `<subcommand>`, which will globally specify the option for every `<command>` and `<subcommand>` to which it's relevant.

Notes:

- `config` files only accept spaces between option flags and values—CodeQL will throw an error if you use `=` to specify an option value.
- If you specify an option in the command line, this overrides the `config` value defined for that option.
- If you want to specify more than one option for a `<command>`, `<subcommand>` or globally, use one line per option.

Examples

- To output all analysis results generated by `codeql database analyze` as CSV format, you would specify:

```
database analyze --format csv
```

Here, you have to specify the command and subcommand to prevent any of the low-level commands that are executed during `database analyze` being passed the same `--format` option.

- To define the RAM (4096 MB) and number of threads (4) to use when running CodeQL commands, specify the following, on separate lines:

```
--ram 4096  
--threads 4
```

- To globally specify a directory for CodeQL to scan for CodeQL packs (which is not a sibling of the installation directory), use:

```
--search-path <path-to-directory>
```

Legal