diagnostic export

In this article

Synopsis
Description
Options

[Experimental] Export diagnostic information for a failed analysis.

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 diagnostic export --format=<format> [--output=<output>] <options>...
```

Description

[Experimental] Export diagnostic information for a failed analysis.

Available since `v2.12.6`.

Options

Primary Options

`--format=<format>`

[Mandatory] The format in which to write the results. One of:

`raw`: A list of raw, uninterpreted diagnostic messages as JSON objects.

`sarif-latest`: Static Analysis Results Interchange Format (SARIF), a JSON-based format

for describing static analysis results. This format option uses the most recent supported version (v2.1.0). This option is not suitable for use in automation as it will produce different versions of SARIF between different CodeQL versions.

`sarifv2.1.0`: SARIF v2.1.0.

`text`: A bullet point list of diagnostic messages.

`-o, --output=<output>`

The output path to write diagnostic information to.

`--sarif-exit-code=<sarifExitCode>`

[SARIF formats only] Exit code of the failing process.

`--sarif-exit-code-description=<sarifExitCodeDescription>`

[SARIF formats only] Reason that the failing process exited.

`--sarif-category=<category>`

[SARIF formats only] Specify a category for this analysis to include in the SARIF output. A category can be used to distinguish multiple analyses performed on the same commit and repository, but on different languages or different parts of the code.

If you analyze the same version of a code base in several different ways (e.g., for different languages) and upload the results to GitHub for presentation in Code Scanning, this value should differ between each of the analyses, which tells Code Scanning that the analyses *supplement* rather than *supersede* each other. (The values should be consistent between runs of the same analysis for *different* versions of the code base.)

This value will appear (with a trailing slash appended if not already present) as the `<run>.automationId` property in SARIF v1, the `<run>.automationLogicalId` property in SARIF v2, and the `<run>.automationDetails.id` property in SARIF v2.1.0.

`--diagnostic-dir=<diagnosticDirs>`

Directory containing CodeQL diagnostic messages. You can pass this multiple times to include multiple directories.

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.)