

Exit codes

In this article

About exit codes

0
1
2
3
32
33
98
99
100
Other

Exit codes signify the status of a command after the CodeQL CLI runs it.

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 exit codes

The CodeQL CLI reports the status of each command it runs as an exit code. This exit code provides information for subsequent commands or for other tools that rely on the CodeQL CLI.

0

Success, normal termination.

1

The command successfully determined that the answer to your question is "no".

This exit code is only used by a few commands, such as "[test run](#)", "[dataset check](#)", "[query format](#)", and "[resolve extractor](#)." For more details, see the documentation for those commands.

2

Something went wrong.

The CLI writes a human-readable error message to stderr. This includes cases where an extractor fails with an internal error, because the `codeql` driver can't distinguish between internal and user-facing errors in extractor behavior.

3

The launcher was unable to find the CodeQL installation directory.

In this case, the launcher can't start the Java code for the CodeQL CLI at all. This should only happen when something is severely wrong with the CodeQL installation.

32

The extractor didn't find any code to analyze when running "[database create](#)" or "[database finalize](#)."

33

One or more query evaluations timed out.

It's possible that some queries that were evaluated in parallel didn't time out. The results for those queries are produced as usual.

98

Evaluation was explicitly canceled.

99

The CodeQL CLI ran out of memory.

This doesn't necessarily mean that all the machine's physical RAM has been used. If you don't use the `--ram` option to set a limit explicitly, the JVM decides on a default limit at startup.

100

A fatal internal error occurred.

This should be considered a bug. The CLI usually writes an abbreviated error description to stderr. If you can reproduce the bug, it's helpful to use `--logdir` and send the log files to GitHub in a bug report.

Other

In the case of really severe problems within the JVM that runs `codeql`, it might return a nonzero exit code of its own choosing. This should only happen if something is severely wrong with the CodeQL installation, or if there is a memory issue with the host system running the CodeQL process. For example, Unix systems may return Exit Code 137 to indicate that the kernel has killed a process that CodeQL has started. One way to troubleshoot this is to modify your `-ram=` flag for the `codeql database analyze` step and re-run your workflow.

Legal