

pack publish

In this article

- Synopsis
- Description
- Options

[Experimental] Publishes a QL library pack to a package registry.

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 pack publish [--dry-run] [--threads=<num>] [--ram=<MB>] [--pack=<folder> | <package.tgz>] <options>... -- <dir>
```

Description

[Experimental] Publishes a QL library pack to a package registry.

This command publishes a pack to a package registry. Before publishing, the pack is first compiled (if necessary) and bundled.

Available since `v2.6.0`.

Options

Primary Options

`<dir>`

The root directory of the package or workspace, defaults to the current working directory. If this parameter points to a directory containing a `qlpack.yml`, then this

operation will run on only that CodeQL package. If this parameter points to a directory containing a `codeql-workspace.yml`, then this operation will run on all CodeQL packages in the workspace.

-n, --[no-]dry-run [↗](#)

Print what would have been done, but do not actually publish the package.

--no-keep [↗](#)

Delete the pack bundle after publishing.

-j, --threads=<num> [↗](#)

Use this many threads to compile queries.

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> [↗](#)

Set total amount of RAM that the compiler should be allowed to use.

--no-precompile [↗](#)

[Advanced] Avoid precompiling the compilation cache in the pack output directory. This will reduce the size of the pack and the time it takes to create it, but will require compilation before the pack can be run. Only meaningful for query packs.

--no-validate-extensions [↗](#)

[Advanced] Avoid validating data extensions as part of the compile step.

Available since `v2.13.3`.

--no-overwrite [↗](#)

[Advanced] Avoid recompiling and overwriting any existing compiled output from a previous invocation of this command.

-p, --pack=<folder> | <package.tgz> [↗](#)

[Advanced] The location to place the packaged tarball. If this option ends in `.tgz`, then the tarball will be created with that name. Otherwise, a new directory is created at the specified path and one or more tarballs are created there. Using `.tgz` is not allowed for workspaces.

--file=<package.tgz> [↗](#)

[Advanced] Upload `package.tgz` directly to the package registry without trying to create or bundle anything new.

--[no-]allow-prerelease [↗](#)

Allow packs with pre-release version qualifiers (e.g., `X.Y.Z-qualifier`) to be used. Without this flag, pre-release packs will be ignored.

Available since `v2.11.3`.

Options for configuring which CodeQL packs to apply this command to. [↗](#)

`--format=<fmt>` [↗](#)

Select output format, either `text` (*default*) or `json`.

`--groups=[-]<group>[, [-]<group>...]` [↗](#)

List of CodeQL pack groups to include or exclude from this operation. A qlpack in the given workspace is included if:

- It is in at least one of the groups listed without a minus sign (this condition is automatically satisfied if there are no groups listed without a minus sign), and
- It is not in any group listed with a minus sign

Options to set up compilation environment [↗](#)

`--search-path=<dir>[:<dir>...]` [↗](#)

A list of directories under which QL packs may be found. Each directory can either be a QL pack (or bundle of packs containing a `.codeqlmanifest.json` file at the root) or the immediate parent of one or more such directories.

If the path contains more than one directory, their order defines precedence between them: when a pack name that must be resolved is matched in more than one of the directory trees, the one given first wins.

Pointing this at a checkout of the open-source CodeQL repository ought to work when querying one of the languages that live there.

If you have checked out the CodeQL repository as a sibling of the unpacked CodeQL toolchain, you don't need to give this option; such sibling directories will always be searched for QL packs that cannot be found otherwise. (If this default does not work, it is strongly recommended to set up `--search-path` once and for all in a per-user configuration file).

(Note: On Windows the path separator is `;`).

`--additional-packs=<dir>[:<dir>...]` [↗](#)

If this list of directories is given, they will be searched for packs before the ones in `--search-path`. The order between these doesn't matter; it is an error if a pack name is found in two different places through this list.

This is useful if you're temporarily developing a new version of a pack that also appears in the default path. On the other hand, it is *not recommended* to override this option in a config file; some internal actions will add this option on the fly, overriding any configured value.

(Note: On Windows the path separator is `;`).

`--library-path=<dir>[:<dir>...]` [↗](#)

[Advanced] An optional list of directories that will be added to the raw import search path for QL libraries. This should only be used if you're using QL libraries that have not been packaged as QL packs.

(Note: On Windows the path separator is `;`).

--dbscheme=<file> [↗](#)

[Advanced] Explicitly define which dbscheme queries should be compiled against. This should only be given by callers that are extremely sure what they're doing.

--compilation-cache=<dir> [↗](#)

[Advanced] Specify an additional directory to use as a compilation cache.

--no-default-compilation-cache [↗](#)

[Advanced] Don't use compilation caches in standard locations such as in the QL pack containing the query or in the CodeQL toolchain directory.

Options for configuring the CodeQL package manager [↗](#)

--registries-auth-stdin [↗](#)

Authenticate to GitHub Enterprise Server Container registries by passing a comma-separated list of <registry_url>=<token> pairs.

For example, you can pass

```
https://containers.GHEHOSTNAME1/v2/=TOKEN1,https://containers.GHEHOSTNAME2/v2/=TOKEN2
```

 to authenticate to two GitHub Enterprise Server instances.

This overrides the CODEQL_REGISTRIES_AUTH and GITHUB_TOKEN environment variables. If you only need to authenticate to the github.com Container registry, you can instead authenticate using the simpler **--github-auth-stdin** option.

--github-auth-stdin [↗](#)

Authenticate to the github.com Container registry by passing a github.com GitHub Apps token or personal access token via standard input.

To authenticate to GitHub Enterprise Server Container registries, pass **--registries-auth-stdin** or use the CODEQL_REGISTRIES_AUTH environment variable.

This overrides the GITHUB_TOKEN environment variable.

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](#)