resolve extensions

In this article

[Experimental] [Deep plumbing] Determine accessible extensions. This includes machine learning models and data extensions.

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 extensions <options>... -- <query|dir|suite|pack>...
```

Description

[Experimental] [Deep plumbing] Determine accessible extensions. This includes machine learning models and data extensions.

This plumbing command resolves the set of data extensions and GitHub-created machine learning models that are available to the query specifiers passed in as command line arguments.

Options

Primary Options

`<querysuite|pack>...`

[Mandatory] Queries to execute. Each argument is in the form `scope/name@range:path` where:

- `scope/name` is the qualified name of a CodeQL pack.
- `range` is a semver range.
- `path` is a file system path.

If a `scope/name` is specified, the `range` and `path` are optional. A missing `range` implies the latest version of the specified pack. A missing `path` implies the default query suite of the specified pack.

The `path` can be one of a `*.ql` query file, a directory containing one or more queries, or a `.qls` query suite file. If there is no pack name specified, then a `path` must be provided, and will be interpreted relative to the current working directory of the current process.

To specify a `path` that contains a literal `@` or `:`, use `path:` as a prefix to the argument, like this: `path:directory/with:and@/chars`.

If a `scope/name` and `path` are specified, then the `path` cannot be absolute. It is considered relative to the root of the CodeQL pack.

`--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 `;`).

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.)