

Extractor options

In this article

About extractors

About extractor options

Setting extractor options with the CodeQL CLI

Setting extractor options from files

You can use the CodeQL CLI to run CodeQL processes locally on software projects.

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 extractors

The CodeQL CLI uses special programs, called extractors, to extract information from the source code of a software system into a database that can be queried. You can customize the behavior of extractors by setting extractor configuration options through the CodeQL CLI.

About extractor options

Each extractor defines its own set of configuration options. To find out which options are available for a particular extractor, you can run `codeql resolve languages` or `codeql resolve extractor` with the `--format=betterjson` option. The `betterjson` output format provides the root paths of extractors and additional information. The output of `codeql resolve extractor --format=betterjson` will often be formatted like the following example:

```
{
  "extractor_root" : "/home/user/codeql/java",
  "extractor_options" : {
    "option1" : {
      "title" : "Java extractor option 1",
      "description" : "An example string option for the Java extractor.",
      "type" : "string",
      "pattern" : "[a-z]+"
    },
    "group1" : {
      "title" : "Java extractor group 1",
      "description" : "An example option group for the Java extractor.",
      "type" : "object",
      "properties" : {
        "option2" : {
          "title" : "Java extractor option 2",
          "description" : "An example array option for the Java
```

```

extractor",
    "type" : "array",
    "pattern" : "[1-9][0-9]*"
  }
}
}
}
}
}
}

```

The extractor option names and descriptions are listed under `extractor_options`. Each option may contain the following fields:

- `title` (required): The title of the option
- `description` (required): The description of the option
- `type` (required): The type of the option, which can be
 - `string`: indicating that the option can have a single string value
 - `array`: indicating that the option can have a sequence of string values
 - `object`: indicating that it is not an option itself, but a grouping that may contain other options and option groups
- `pattern` (optional): The regular expression patterns that all values of the option should match. Note that the extractor may impose additional constraints on option values that are not or cannot be expressed in this regular expression pattern. Such constraints, if they exist, would be explained under the description field.
- `properties` (optional): A map from extractor option names in the option group to the corresponding extractor option descriptions. This field can only be present for option groups. For example, options of `object` type.

In the example above, the extractor declares two options:

- `option1` is a `string` option with value matching `[a-z]+`
- `group1.option2` is an `array` option with values matching `[1-9][0-9]*`

Setting extractor options with the CodeQL CLI [↗](#)

The CodeQL CLI supports setting extractor options in subcommands that directly or indirectly invoke extractors. These commands are:

- `codeql database create`
- `codeql database start-tracing`
- `codeql database trace-command`
- `codeql database index-files`

When running these subcommands, you can set extractor options with the `--extractor-option` CLI option. For example:

- `codeql database create --extractor-option java.option1=abc ...`
- `codeql database start-tracing --extractor-option java.group1.option2=102 ...`

`--extractor-option` requires exactly one argument of the form `extractor_option_name=extractor_option_value`. `extractor_option_name` is the name of the extractor (in this example, `java`) followed by a period and then the name of the extractor option (in this example, either `option1` or `group1.option2`). `extractor_option_value` is the value being assigned to the extractor option. The value must match the regular expression pattern of the extractor option (if it exists), and it must not contain newline characters.

Using `--extractor-option` to assign an extractor option that does not exist is an error.

The CodeQL CLI accepts multiple `--extractor-option` options in the same invocation. If

you set a `string` extractor option multiple times, the last option value overwrites all previous ones. If you set an array extractor option multiple times, all option values are concatenated in order.

You can also specify extractor option names without the extractor name. For example:

- `codeql database create --extractor-option option1=abc ...`
- `codeql database start-tracing --extractor-option group1.option2=102 ...`

If you do not specify an extractor name, the extractor option settings will apply to all extractors that declare an option with the given name. In the above example, the first command would set the extractor option `option1` to `abc` for the `java` extractor and every extractor that has an option of `option1`, for example the `cpp` extractor, if the `option1` extractor option exists for that extractor.

Setting extractor options from files [🔗](#)

You can also set extractor options through a file. The CodeQL CLI subcommands that accept `--extractor-option` also accept `--extractor-options-file`, which has a required argument of the path to a YAML file (with extension `.yaml` or `.yml`) or a JSON file (with extension `.json`). For example:

- `codeql database create --extractor-options-file options.yml ...`
- `codeql database start-tracing --extractor-options-file options.json ...`

Each option file contains a tree structure of nested maps. At the root is an extractor map key, and beneath it are map keys that correspond to extractor names. Starting at the third level, there are extractor options and option groups.

In JSON:

```
{
  "extractor" : {
    "java": {
      "option1" : "abc",
      "group1" : {
        "option2" : [ 102 ]
      }
    }
  }
}
```

In YAML:

```
extractor:
  java:
    option1: "abc"
    group1:
      option2: [ 102 ]
```

The value for a `string` extractor option must be a string or a number (which will be converted to a string before further processing).

The value for an `array` extractor option must be an array of strings or numbers.

The value for an option group (of type `object`) must be a map, which may contain nested extractor options and option groups.

Each extractor option value must match the regular expression pattern of the extractor option (if it exists), and it must not contain newline characters.

Assigning an extractor option that does not exist is an error. You can make the CodeQL

CLI ignore unknown extractor options by using a special `__allow_unknown_properties` Boolean field. For example, the following option file asks the CodeQL CLI to ignore all unknown extractor options and option groups under `group1` :

```
extractor:
  java:
    option1: "abc"
    group1:
      __allow_unknown_properties: true
      option2: [ 102 ]
```

You can specify `--extractor-options-file` multiple times. The extractor option assignments are processed in the following order:

- 1 All extractor option files specified by `--extractor-options-file` are processed in the order they appear on the command line, then
- 2 All extractor option assignments specified by `--extractor-option` are processed in the order they appear on the command line

The same rules govern what happens when the same extractor option is set multiple times, regardless of whether the assignments are done using `--extractor-option` , using `--extractor-options-file` , or some combination of the two. If you set a `string` extractor option multiple times, the last option value overwrites all previous values. If you set an `array` extractor option multiple times, all option values are concatenated in order.

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)