

bqrs decode

In this article

- Synopsis
- Description
- Options

Convert result data from BQRS into other forms.

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 bqrs decode [--output=<file>] [--result-set=<name>] [--sort-key=<col>[, <col>...]] <options>... -- <file>
```

Description

Convert result data from BQRS into other forms.

The decoded output will be written to standard output, unless the `--output` option is specified.

Options

Primary Options

`<file>`

[Mandatory] BQRS file to decode.

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

The file to write the desired output to.

-r, --result-set=<name> 

Select a particular result set from the BQRS file to decode. The available results sets can be listed by [codeql bQRS info](#).

If no result set is selected, all result sets will be decoded, provided the selected output format and processing options support that. Otherwise an error results.

-k, --sort-key=<col>[,<col>...] 

Sort the selected result set by the indicated columns.

--sort-direction=<direction>[,<direction>...] 

Sort the selected result set using the indicated sort directions.

If sort directions are not specified, then ascending order will be used for all columns.

Output format options

--format=<fmt> 

Select output format. Choices include:

text (*default*): A human-readable plain text table.


csv : Comma-separated values.

json : Streaming JSON.

bQRS : BQRS. This must be used with **--output** . Most useful together with **--sort-key** .

--no-titles 

Omit column titles for **text** and **csv** formats

--entities=<fmt>[,<fmt>...] 

[Advanced] Control how result columns of entity type are shown. A comma-separated list of the following choices:

url : A URL referring to a source location, if the query was compiled to produce such URLs for entity types.

string : A string computed by the toString() method in QL, if the query was compiled to produce such strings for the column.

id : The internal ID of the entity, which may not be informative.

all : Show columns with all the information the BQRS file provides.

All the selected options are shown, if possible.

Options for pagination (for use by interactive front-ends)

--rows=<num> 

[Advanced] Output this many rows from the selected resultset, starting at the top, or at

the location given by `--start-at` .

`--start-at=<offset>` [↗](#)

[Advanced] Start printing the row defined at a particular byte offset in the BQRS file. The offset must be gotten from [codeql bqrs info](#), or from the "next" pointer found in JSON output from a previous invocation with `--rows` set. Other offsets are likely to produce nonsense output and/or explicit errors.

Must always be used together with `--rows` , and is incompatible with `--sort-key` .

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](#)