Using the bq command-line tool

cloud.google.com/bigquery/docs/bq-command-line-tool

The `bq` command-line tool is a Python-based command-line tool for BigQuery. This page contains general information about using the `bq` command-line tool.

For a complete reference of all `bq` commands and flags, see the bq command-line tool reference.

Before you begin

Before you can use the `bq` command-line tool, you must use the Google Cloud Console to create or select a project.

1. In the Google Cloud Console, on the project selector page, select or create a Google Cloud project.

   Note: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.
   Go to project selector

2. BigQuery is automatically enabled in new projects. To activate BigQuery in a preexisting project, go to Enable the BigQuery API.
   Enable the API

3. Optional: Enable billing for the project. If you don't want to enable billing or provide a credit card, the steps in this document still work. BigQuery provides a sandbox to perform the steps.

Entering `bq` commands in Cloud Shell

You can enter `bq` command-line tool commands in Cloud Shell either from the Google Cloud Console or from the Cloud SDK.

- To use `bq` command-line tool from the Cloud Console, activate Cloud Shell:

  Activate Cloud Shell

- To use `bq` command-line tool from the Cloud SDK, install and configure the Cloud SDK.

Positioning flags and arguments

The `bq` command-line tool supports two kinds of flags:

- *Global flags* can be used in all commands.
- *Command-specific flags* apply to a specific command.

For a list of available global and command-specific flags, see bq command-line tool reference.

Place any global flags before the `bq` command, and then include command-specific flags. You can include multiple global or command-specific flags. For example:

```
bq --location=us mk --reservation --project_id=project reservation_name
```

You can specify command arguments in the following ways:

- `--FLAG ARGUMENT` (as shown in the previous examples)
- `--FLAG=ARGUMENT`
- `--FLAG='ARGUMENT'`
- `--FLAG="ARGUMENT"`
- `--FLAG 'ARGUMENT'`
- `--FLAG "ARGUMENT"`

Replace the following:

- `FLAG` : a global or command-specific flag
- `ARGUMENT` : the flag's argument

Some commands require the use of single or double quotes around arguments. This is often true when the argument contains spaces, commas, or other special characters. For example:

```
bq query --nouse_legacy_sql \
'SELECT
   COUNT(*)
 FROM
   `bigquery-public-data`.samples.shakespeare'
```

Flags with boolean values can be specified without an argument. If you specify `true` or `false` , then you must use the format `FLAG=ARGUMENT` .

For example, this command specifies false for the boolean flag `--use_legacy_sql` by placing `no` at the front of the flag:

```
bq query --nouse_legacy_sql \
'SELECT
   COUNT(*)
 FROM
   `bigquery-public-data`.samples.shakespeare'
```

Alternatively, to specify `false` as the flag's argument, you can enter the following:

```
bq query --use_legacy_sql=false \
'SELECT
   COUNT(*)
 FROM
   `bigquery-public-data`.samples.shakespeare'
```

Running queries from the `bq` command-line tool

To take a query that you've developed in the Google Cloud Console and run it from the `bq` command-line tool, do the following:

1. Include the query in a `bq query` command as follows: `bq query --use_legacy_sql=false 'QUERY'`. Replace `QUERY` with the query.
2. Replace any single quotes (`'`) in the query with double quotes (`"`).
3. Remove comments from the query.

For example, transform the following Google Cloud Console query:

```
-- count Shakespeare's use of the string "raisin"
SELECT
  word,
  SUM(word_count) AS count
FROM
  `bigquery-public-data`.samples.shakespeare
WHERE
  word LIKE '%raisin%'
GROUP BY
  word
```

into a `bq` command-line tool query as follows:

```
bq query --use_legacy_sql=false \
'SELECT
  word,
  SUM(word_count) AS count
FROM
  `bigquery-public-data`.samples.shakespeare
WHERE
  word LIKE "%raisin%"
GROUP BY
  word'
```

For more information, see Running interactive and batch query jobs.

Getting help

To get help for the `bq` command-line tool, you can enter the following commands:

- For the installed version of the `bq` command-line tool, enter `bq version`.
- For a full list of commands, enter `bq help`.
- For a list of global flags, enter `bq --help`.
- For help with a specific command, enter `bq help COMMAND`.
- For help with a specific command plus a list of global flags, enter `bq COMMAND --help`.

Replace `COMMAND` with the command that you need help with.

Setting default values for command-line flags

You can set default values for command-line flags by including them in the `bq` command-line tool's configuration file, `.bigqueryrc`. Before you configure your default options, you must first create a `.bigqueryrc` file. You can use your preferred text editor to create the

file. After you create the `.bigqueryrc` file, you can specify the path to the file using the `--bigqueryrc` global flag.

If the `--bigqueryrc` flag is not specified, then the `BIGQUERYRC` environment variable is used. If that is not specified, then the path `~/.bigqueryrc` is used. The default path is `$HOME/.bigqueryrc`.

Note: Creating a `.bigqueryrc` file using the `bq init` command is not recommended.

Adding flags to `.bigqueryrc`

To add default values for command-line flags to `.bigqueryrc`:

- Place global flags at the top of the file without a header.
- For command-specific flags, enter the command name (in brackets) and add the command-specific flags (one per line) after the command name.

For example:

```
--apilog=stdout
--format=prettyjson
--location=
US


[query]
--use_legacy_sql=false
--max_rows=100
--maximum_bytes_billed=10000000
[load]
--
destination_kms_key=projects/myproject/locations/mylocation/keyRings/myRing/cryptoKeys
```

The preceding example sets default values for the following flags:

- The global flag `--apilog` is set to `stdout` to print debugging output to the Cloud Console.
- The global flag `--format` is set to `prettyjson` to display command output in a human-readable JSON format.
- The global flag `--location` is set to the `US` multi-region location.
- The `query` command-specific flag `--use_legacy_sql` is set to `false` to make standard SQL the default query syntax.

  Note: You cannot use `--nouse_legacy_sql` in `.bigqueryrc`.
- The `query` command-specific flag `--max_rows` is set to `100` to control the number of rows in the query output.
- The `query` command-specific flag `--maximum_bytes_billed` is set to 10,000,000 bytes (10 MB) to fail queries that read more than 10 MB of data.
- The `load` command-specific flag `--destination_kms_key` is set to `projects/myproject/locations/mylocation/keyRings/myRing/cryptoKeys/myKey`.

Running the `bq` command-line tool in an interactive shell

You can run the `bq` command-line tool in an interactive shell where you don't need to prefix the commands with `bq`. To start interactive mode, enter `bq shell`. After launching the shell, the prompt changes to the ID of your default project. To exit interactive mode, enter `exit`.

Running the `bq` command-line tool in a script

You can run the `bq` command-line tool in a script, as you would run a gcloud command-line tool command. The following is an example of `gcloud` and `bq` commands in a bash script:

```
#!/bin/bash
gcloud config set project myProject
bq query --use_legacy_sql=false --destination_table=myDataset.myTable \
'SELECT
   word,
   SUM(word_count) AS count
 FROM
   `bigquery-public-data`.samples.shakespeare
 WHERE
   word LIKE "%raisin%"
 GROUP BY
   word'
```

Running `bq` commands from a service account

To run `bq` commands using a service account, you must authorize access to Google Cloud from the service account. For more information, see gcloud auth activate-service-account.

Examples

You can find command-line examples throughout the How-to guides section of the BigQuery documentation. Below are links to common command-line tasks such as creating, getting, listing, deleting, and modifying BigQuery resources.

Creating resources

For information about using the `bq` command-line tool to create resources, see the following:

- Creating a dataset
- Creating an empty table with a schema definition
- Creating a table from a query result
- Creating an ingestion-time partitioned table
- Creating a view

For examples of creating a table using a data file, see Loading data.

Getting information about resources

For information about using the `bq` command-line tool to get information about resources, see the following:

- Getting information about datasets
- Getting information about tables
- Getting information about views

Listing resources

For information about using the `bq` command-line tool to list resources, see the following:

- Listing datasets
- Listing tables
- Listing views

Updating resources

For information about using the `bq` command-line tool to update resources, see the following:

Loading data

For information about using the `bq` command-line tool to load data, see the following:

Querying data

For information about using the `bq` command-line tool to query data, see the following:

Using external data sources

For information about using the `bq` command-line tool to query data in external data sources, see the following:

Exporting data

For information about using the `bq` command-line tool to export data, see the following:

Exporting data stored in BigQuery

Using the BigQuery Data Transfer Service

For information about using the `bq` command-line tool with the BigQuery Data Transfer Service, see the following:

Troubleshooting the `bq` command-line tool

This section shows you how to resolve issues with `bq` command-line tool.

Keep your Cloud SDK up to date

If you are using the `bq` command-line tool from the Cloud SDK, then make sure that you have the latest functionality and fixes for the `bq` command-line tool by keeping your Cloud SDK installation up to date. To see whether you are running the latest version of the Cloud SDK, enter the following command in Cloud Shell:

```
gcloud components list
```

The first two lines of the output display the version number of your current Cloud SDK installation and the version number of the most recent Cloud SDK. If you discover that your version is out of date, then you can update your Cloud SDK installation to the most recent version by entering the following command in Cloud Shell:

```
gcloud components update
```

Debugging

You can enter the following commands to debug the `bq` command-line tool:

- **See requests sent and received.** Add the `--apilog=PATH_TO_FILE` flag to save a log of operations to a local file. Replace `PATH_TO_FILE` with the path that you want to save the log to. The `bq` command-line tool works by making standard REST-based API calls, which can be useful to see. It's also useful to attach this log when you're reporting issues. Using `-` or `stdout` instead of a path prints the log to the Google Cloud Console. Setting `--apilog` to `stderr` outputs to the standard error file.

- **Troubleshoot errors.** Enter the `--format=prettyjson` flag when getting a job's status or when viewing detailed information about resources such as tables and datasets. Using this flag outputs the response in JSON format, including the `reason` property. You can use the `reason` property to look up troubleshooting steps.