

# clusterdb

---



- [Description](#)
  - [Options](#)
  - [Environment](#)
  - [Diagnostics](#)
  - [Examples](#)
  - [See Also](#)
- 

clusterdb

clusterdb

1

Application

clusterdb

cluster a

PostgreSQL

database

clusterdb

connection-option

option

--table

-t

table

dbname

-a

--all

# Description

clusterdb is a utility for recluster tables in a PostgreSQL database. It finds tables that have previously been clustered, and clusters them again on the same index that was last used. Tables that have never been clustered are not affected.

clusterdb is a wrapper around the SQL command [???](#). There is no effective difference between clustering databases via this utility and via other methods for accessing the server.

# Options

clusterdb accepts the following command-line arguments:

`-a`; `--all`

Cluster all databases.

`-d dbname`; `--dbname=dbname`

Specifies the name of the database to be clustered, when `-a`/`--all` is not used. If this is not specified, the database name is read from the environment variable `PGDATABASE`. If that is not set, the user name specified for the connection is used. The `<dbname>` can be a [connection string](#). If so, connection string parameters will override any conflicting command line options.

`-e`; `--echo`

Echo the commands that clusterdb generates and sends to the server.

`-q`; `--quiet`

Do not display progress messages.

`-t table`; `--table=table`

Cluster `<table>` only. Multiple tables can be clustered by writing multiple `-t` switches.

`-v`; `--verbose`

Print detailed information during processing.

`-V`; `--version`

Print the clusterdb version and exit.

`-?;` `--help`

Show help about clusterdb command line arguments, and exit.

clusterdb also accepts the following command-line arguments for connection parameters:

`-h host;` `--host=host`

Specifies the host name of the machine on which the server is running. If the value begins with a slash, it is used as the directory for the Unix domain socket.

`-p port;` `--port=port`

Specifies the TCP port or local Unix domain socket file extension on which the server is listening for connections.

`-U username;` `--username=username`

User name to connect as.

`-w;` `--no-password`

Never issue a password prompt. If the server requires password authentication and a password is not available by other means such as a `.pgpass` file, the connection attempt will fail. This option can be useful in batch jobs and scripts where no user is present to enter a password.

`-W;` `--password`

Force clusterdb to prompt for a password before connecting to a database.

This option is never essential, since clusterdb will automatically prompt for a password if the server demands password authentication. However, clusterdb will waste a connection attempt finding out that the server wants a password. In some cases it is worth typing `-W` to avoid the extra connection attempt.

`--maintenance-db=dbname`

When the `-a`/`--all` is used, connect to this database to gather the list of databases to cluster. If not specified, the `postgres` database will be used, or if that does not exist, `template1` will be used.

This can be a [connection string](#). If so, connection string parameters will override any conflicting command line options. Also, connection string parameters other than the database name itself will be re-used when connecting to other databases.

# Environment

`PGDATABASE;` `PGHOST;` `PGPORT;` `PGUSER`

Default connection parameters

`PG_COLOR`

Specifies whether to use color in diagnostic messages. Possible values are `always`, `auto` and `never`

This utility, like most other PostgreSQL utilities, also uses the environment variables supported by libpq (see [???](#)).

# Diagnostics

In case of difficulty, see [???](#) and [???](#) for discussions of potential problems and error messages. The database server must be running at the targeted host. Also, any default connection settings and environment variables used by the libpq front-end library will apply.

# Examples

To cluster the database `test`:

```
$ clusterdb test
```

To cluster a single table `foo` in a database named `xyzyz`:

```
$ clusterdb --table=foo xyzyz
```

# See Also

---

`???` #2

`???` `???` 2026-10-31 14:08:34 UTC

`???` `???` 2026-10-31 14:08:35 UTC