NAME

dropdb - remove a PostgreSQL database

SYNOPSIS

dropdb [*connection-option*...] [*option*...] *dbname*

DESCRIPTION

dropdb destroys an existing PostgreSQL database. The user who executes this command must be a database superuser or the owner of the database.

dropdb is a wrapper around the SQL command **DROP DATABASE**. There is no effective difference between dropping databases via this utility and via other methods for accessing the server.

OPTIONS

dropdb accepts the following command-line arguments:

dbname
Specifies the name of the database to be removed.

-e
--echo
Echo the commands that dropdb generates and sends to the server.

-f
--force
Attempt to terminate all existing connections to the target database before dropping it. See DROP DATABASE (**DROP_DATABASE**(7)) for more information on this option.

-i
--interactive
Issues a verification prompt before doing anything destructive.

-V
--version
Print the dropdb version and exit.

--if-exists
Do not throw an error if the database does not exist. A notice is issued in this case.

-?

--help
Show help about dropdb command line arguments, and exit.

dropdb 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 dropdb to prompt for a password before connecting to a database.

This option is never essential, since dropdb will automatically prompt for a password if the server demands password authentication. However, dropdb 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*
Specifies the name of the database to connect to in order to drop the target database. If not specified, the postgres database will be used; if that does not exist (or is the database being dropped), template1 will be used. This can be a connection string. If so, connection string parameters will override any conflicting command line options.

ENVIRONMENT

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 Section 34.15).

DIAGNOSTICS

In case of difficulty, see DROP DATABASE (**DROP_DATABASE**(7)) and **psql**(1) 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 destroy the database demo on the default database server:

```
$ dropdb demo
```

To destroy the database demo using the server on host eden, port 5000, with verification and a peek at the underlying command:

```
$ dropdb -p 5000 -h eden -i -e demo
Database "demo" will be permanently deleted.
Are you sure? (y/n) y
DROP DATABASE demo;
```

SEE ALSO

createdb(1), DROP DATABASE (**DROP_DATABASE**(7))