# pg_dumpall

https://www.postgresql.org/docs/9.6/static/app-pg-dumpall.html

## Name

pg_dumpall -- extract a PostgreSQL database cluster into a script file

## Synopsis

`pg_dumpall` [`connection-option`...] [`option`...]

## Description

pg_dumpall is a utility for writing out ("dumping") all PostgreSQL databases of a cluster into one script file. The script file contains SQL commands that can be used as input to psql to restore the databases. It does this by calling pg_dump for each database in a cluster. pg_dumpall also dumps global objects that are common to all databases. (pg_dump does not save these objects.) This currently includes information about database users and groups, tablespaces, and properties such as access permissions that apply to databases as a whole.

Since pg_dumpall reads tables from all databases you will most likely have to connect as a database superuser in order to produce a complete dump. Also you will need superuser privileges to execute the saved script in order to be allowed to add users and groups, and to create databases.

The SQL script will be written to the standard output. Use the [-f|file] option or shell operators to redirect it into a file.

pg_dumpall needs to connect several times to the PostgreSQL server (once per database). If you use password authentication it will ask for a password each time. It is convenient to have a `~/.pgpass` file in such cases. See Section 32.15 for more information.

# 32.15. The Password File

The file `.pgpass` in a user's home directory or the file referenced by `PGPASSFILE` can contain passwords to be used if the connection requires a password (and no password has been specified otherwise). On Microsoft Windows the file is named `%APPDATA%\postgresql\pgpass.conf` (where `%APPDATA%` refers to the Application Data subdirectory in the user's profile).

This file should contain lines of the following format:

```
hostname:port:database:username:password
```

(You can add a reminder comment to the file by copying the line above and preceding it with #.) Each of the first four fields can be a literal value, or `*`, which matches anything. The password field from the first line that matches the current connection parameters will be used. (Therefore, put more-specific entries first when you are using wildcards.) If an entry needs to contain `:` or `\`, escape this character with `\`. A host name of `localhost` matches both TCP (host name `localhost`) and Unix domain socket (`pghost` empty or the default socket directory) connections coming from the local machine. In a standby server, a database name of `replication` matches streaming replication connections made to the master server. The `database` field is of limited usefulness because users have the same password for all databases in the same cluster.

On Unix systems, the permissions on `.pgpass` must disallow any access to world or group; achieve this by the command `chmod 0600 ~/.pgpass`. If the permissions are less strict than this, the file will be ignored. On Microsoft Windows, it is assumed that the file is stored in a directory that is secure, so no special permissions check is made.

# Options

The following command-line options control the content and format of the output.

`-a`
`--data-only`

Dump only the data, not the schema (data definitions).

`-c`
`--clean`

Include SQL commands to clean (drop) databases before recreating them. `DROP` commands for roles and tablespaces are added as well.

`-f filename`
`--file=filename`

Send output to the specified file. If this is omitted, the standard output is used.

`-g`
`--globals-only`

Dump only global objects (roles and tablespaces), no databases.

`-o`
`--oids`

Dump object identifiers (OIDs) as part of the data for every table. Use this option if your application references the OID columns in some way (e.g., in a foreign key constraint). Otherwise, this option should not be used.

`-O`
`--no-owner`

Do not output commands to set ownership of objects to match the original database. By default, pg_dumpall issues `ALTER OWNER` or `SET SESSION AUTHORIZATION` statements to set ownership of created schema elements. These statements will fail when the script is run unless it is started by a superuser (or the same user that owns all of the objects in the script). To make a script that can be restored by any user, but will give that user ownership of all the objects, specify `-O`.

`-r`
`--roles-only`

Dump only roles, no databases or tablespaces.

`-s`
`--schema-only`

Dump only the object definitions (schema), not data.

`-S username`
`--superuser=username`

Specify the superuser user name to use when disabling triggers. This is relevant only if `--disable-triggers` is used. (Usually, it's better to leave this out, and instead start the resulting script as superuser.)

`-t`
`--tablespaces-only`

Dump only tablespaces, no databases or roles.

`-v`
`--verbose`

Specifies verbose mode. This will cause pg_dumpall to output start/stop times to the dump file, and progress messages to standard error. It will also enable verbose output in pg_dump.

`-V`
`--version`

Print the pg_dumpall version and exit.

`-x`
`--no-privileges`
`--no-acl`

Prevent dumping of access privileges (grant/revoke commands).

`--binary-upgrade`

This option is for use by in-place upgrade utilities. Its use for other purposes is not recommended or supported. The behavior of the option may change in future releases without notice.

`--column-inserts`
`--attribute-inserts`

Dump data as `INSERT` commands with explicit column names (`INSERT INTO table (column, ...) VALUES ...`). This will make restoration very slow; it is mainly useful for making dumps that can be loaded into non-PostgreSQL databases.

`--disable-dollar-quoting`

This option disables the use of dollar quoting for function bodies, and forces them to be quoted using SQL standard string syntax.

`--disable-triggers`

This option is relevant only when creating a data-only dump. It instructs pg_dumpall to include commands to temporarily disable triggers on the target tables while the data is reloaded. Use this if you have referential integrity checks or other triggers on the tables that you do not want to invoke during data reload.

Presently, the commands emitted for `--disable-triggers` must be done as superuser. So, you should also specify a superuser name with `-S`, or preferably be careful to start the resulting script as a superuser.

`--if-exists`

Use conditional commands (i.e. add an `IF EXISTS` clause) to clean databases and other objects. This option is not valid unless `--clean` is also specified.

`--inserts`

Dump data as `INSERT` commands (rather than `COPY`). This will make restoration very slow; it is mainly useful for making dumps that can be loaded into non-PostgreSQL databases. Note that the restore might fail altogether if you have rearranged column order. The `--column-inserts` option is safer, though even slower.

`--lock-wait-timeout=timeout`

Do not wait forever to acquire shared table locks at the beginning of the dump. Instead, fail if unable to lock a table within the specified `timeout`. The timeout may be specified in any of the formats accepted by `SET statement_timeout`. Allowed values vary depending on the server version you are dumping from, but an integer number of milliseconds is accepted by all versions since 7.3. This option is ignored when dumping from a pre-7.3 server.

`--no-security-labels`

Do not dump security labels.

`--no-tablespaces`

Do not output commands to create tablespaces nor select tablespaces for objects. With this option, all objects will be created in whichever tablespace is the default during restore.

`--no-unlogged-table-data`

Do not dump the contents of unlogged tables. This option has no effect on whether or not the table definitions (schema) are dumped; it only suppresses dumping the table data.

`--quote-all-identifiers`

Force quoting of all identifiers. This option is recommended when dumping a database from a server whose PostgreSQL major version is different from pg_dumpall's, or when the output is intended to be loaded into a server of a different major version. By default, pg_dumpall quotes only identifiers that are reserved words in its own major version. This sometimes results in compatibility issues when dealing with servers of other versions that may have slightly different sets of reserved words. Using `--quote-all-identifiers` prevents such issues, at the price of a harder-to-read dump script.

`--use-set-session-authorization`

Output SQL-standard `SET SESSION AUTHORIZATION` commands instead of `ALTER OWNER` commands to determine object ownership. This makes the dump more standards compatible, but depending on the history of the objects in the dump, might not restore properly.

`-?`
`--help`

Show help about pg_dumpall command line arguments, and exit.

The following command-line options control the database connection parameters.

`-d connstr`
`--dbname=connstr`

Specifies parameters used to connect to the server, as a connection string. See Section 32.1.1 for more information.

The option is called `--dbname` for consistency with other client applications, but because pg_dumpall needs to connect to many databases, database name in the connection string will be

ignored. Use `-l` option to specify the name of the database used to dump global objects and to discover what other databases should be dumped.

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

Specifies the host name of the machine on which the database server is running. If the value begins with a slash, it is used as the directory for the Unix domain socket. The default is taken from the `PGHOST` environment variable, if set, else a Unix domain socket connection is attempted.

`-l dbname`
`--database=dbname`

Specifies the name of the database to connect to for dumping global objects and discovering what other databases should be dumped. If not specified, the `postgres` database will be used, and if that does not exist, `template1` will be used.

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

Specifies the TCP port or local Unix domain socket file extension on which the server is listening for connections. Defaults to the `PGPORT` environment variable, if set, or a compiled-in default.

`-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 pg_dumpall to prompt for a password before connecting to a database.

This option is never essential, since pg_dumpall will automatically prompt for a password if the server demands password authentication. However, pg_dumpall 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.

Note that the password prompt will occur again for each database to be dumped. Usually, it's better to set up a `~/.pgpass` file than to rely on manual password entry.

`--role=rolename`

Specifies a role name to be used to create the dump. This option causes pg_dumpall to issue a `SET ROLE rolename` command after connecting to the database. It is useful when the authenticated user (specified by `-U`) lacks privileges needed by pg_dumpall, but can switch to a role with the required rights. Some installations have a policy against logging in directly as a superuser, and use of this option allows dumps to be made without violating the policy.

## Environment

`PGHOST PGOPTIONS PGPORT PGUSER`

Default connection parameters

This utility, like most other PostgreSQL utilities, also uses the environment variables supported by libpq (see Section 32.14).

## Notes

Since pg_dumpall calls pg_dump internally, some diagnostic messages will refer to pg_dump.

Once restored, it is wise to run `ANALYZE` on each database so the optimizer has useful statistics. You can also run `vacuumdb -a -z` to analyze all databases.

pg_dumpall requires all needed tablespace directories to exist before the restore; otherwise, database creation will fail for databases in non-default locations.

## Examples

To dump all databases:

```
$ pg_dumpall > db.out
```

To reload database(s) from this file, you can use:

```
$ psql -f db.out postgres
```

(It is not important to which database you connect here since the script file created by pg_dumpall will contain the appropriate commands to create and connect to the saved databases.)

## See Also

Check pg_dump for details on possible error conditions.

---