

NAME

`initdb` - create a new PostgreSQL database cluster

SYNOPSIS

`initdb` [*option...*] [`--pgdata` | `-D`] *directory*

DESCRIPTION

`initdb` creates a new PostgreSQL database cluster. A database cluster is a collection of databases that are managed by a single server instance.

Creating a database cluster consists of creating the directories in which the database data will live, generating the shared catalog tables (tables that belong to the whole cluster rather than to any particular database), and creating the `postgres`, `template1`, and `template0` databases. The `postgres` database is a default database meant for use by users, utilities and third party applications. `template1` and `template0` are meant as source databases to be copied by later **CREATE DATABASE** commands. `template0` should never be modified, but you can add objects to `template1`, which by default will be copied into databases created later. See Section 23.3 for more details.

Although `initdb` will attempt to create the specified data directory, it might not have permission if the parent directory of the desired data directory is root-owned. To initialize in such a setup, create an empty data directory as root, then use **chown** to assign ownership of that directory to the database user account, then **su** to become the database user to run `initdb`.

`initdb` must be run as the user that will own the server process, because the server needs to have access to the files and directories that `initdb` creates. Since the server cannot be run as root, you must not run `initdb` as root either. (It will in fact refuse to do so.)

For security reasons the new cluster created by `initdb` will only be accessible by the cluster owner by default. The **--allow-group-access** option allows any user in the same group as the cluster owner to read files in the cluster. This is useful for performing backups as a non-privileged user.

`initdb` initializes the database cluster's default locale and character set encoding. These can also be set separately for each database when it is created. `initdb` determines those settings for the template databases, which will serve as the default for all other databases. By default, `initdb` uses the locale provider `libc`, takes the locale settings from the environment, and determines the encoding from the locale settings. This is almost always sufficient, unless there are special requirements.

To choose a different locale for the cluster, use the option **--locale**. There are also individual options **--lc-*** (see below) to set values for the individual locale categories. Note that inconsistent settings for different locale categories can give nonsensical results, so this should be used with care.

Alternatively, the ICU library can be used to provide locale services. (Again, this only sets the default for subsequently created databases.) To select this option, specify `--locale-provider=icu`. To choose the specific ICU locale ID to apply, use the option `--icu-locale`. Note that for implementation reasons and to support legacy code, **initdb** will still select and initialize libc locale settings when the ICU locale provider is used.

When **initdb** runs, it will print out the locale settings it has chosen. If you have complex requirements or specified multiple options, it is advisable to check that the result matches what was intended.

More details about locale settings can be found in Section 24.1.

To alter the default encoding, use the `--encoding`. More details can be found in Section 24.3.

OPTIONS

-A *authmethod*

--auth=authmethod

This option specifies the default authentication method for local users used in `pg_hba.conf` (host and local lines). See Section 21.1 for an overview of valid values.

initdb will prepopulate `pg_hba.conf` entries using the specified authentication method for non-replication as well as replication connections.

Do not use `trust` unless you trust all local users on your system. `trust` is the default for ease of installation.

--auth-host=authmethod

This option specifies the authentication method for local users via TCP/IP connections used in `pg_hba.conf` (host lines).

--auth-local=authmethod

This option specifies the authentication method for local users via Unix-domain socket connections used in `pg_hba.conf` (local lines).

-D *directory*

--pgdata=directory

This option specifies the directory where the database cluster should be stored. This is the only information required by **initdb**, but you can avoid writing it by setting the **PGDATA** environment variable, which can be convenient since the database server (**postgres**) can find the database directory later by the same variable.

-E *encoding*

--encoding=*encoding*

Selects the encoding of the template databases. This will also be the default encoding of any database you create later, unless you override it then. The default is derived from the locale, if the libc locale provider is used, or UTF8 if the ICU locale provider is used. The character sets supported by the PostgreSQL server are described in Section 24.3.1.

-g

--allow-group-access

Allows users in the same group as the cluster owner to read all cluster files created by **initdb**. This option is ignored on Windows as it does not support POSIX-style group permissions.

--icu-locale=*locale*

Specifies the ICU locale ID, if the ICU locale provider is used.

-k

--data-checksums

Use checksums on data pages to help detect corruption by the I/O system that would otherwise be silent. Enabling checksums may incur a noticeable performance penalty. If set, checksums are calculated for all objects, in all databases. All checksum failures will be reported in the `pg_stat_database` view. See Section 30.2 for details.

--locale=*locale*

Sets the default locale for the database cluster. If this option is not specified, the locale is inherited from the environment that **initdb** runs in. Locale support is described in Section 24.1.

--lc-collate=*locale*

--lc-ctype=*locale*

--lc-messages=*locale*

--lc-monetary=*locale*

--lc-numeric=*locale*

--lc-time=*locale*

Like **--locale**, but only sets the locale in the specified category.

--no-locale

Equivalent to **--locale=C**.

--locale-provider={**libc**|**icu**}

This option sets the locale provider for databases created in the new cluster. It can be overridden in the **CREATE DATABASE** command when new databases are subsequently created. The

default is `libc`.

-N

--no-sync

By default, **initdb** will wait for all files to be written safely to disk. This option causes **initdb** to return without waiting, which is faster, but means that a subsequent operating system crash can leave the data directory corrupt. Generally, this option is useful for testing, but should not be used when creating a production installation.

--no-instructions

By default, **initdb** will write instructions for how to start the cluster at the end of its output. This option causes those instructions to be left out. This is primarily intended for use by tools that wrap **initdb** in platform-specific behavior, where those instructions are likely to be incorrect.

--pwfile=*filename*

Makes **initdb** read the database superuser's password from a file. The first line of the file is taken as the password.

-S

--sync-only

Safely write all database files to disk and exit. This does not perform any of the normal **initdb** operations. Generally, this option is useful for ensuring reliable recovery after changing `fsync` from off to on.

-T *config*

--text-search-config=*config*

Sets the default text search configuration. See `default_text_search_config` for further information.

-U *username*

--username=*username*

Selects the user name of the database superuser. This defaults to the name of the effective user running **initdb**. It is really not important what the superuser's name is, but one might choose to keep the customary name `postgres`, even if the operating system user's name is different.

-W

--pwprompt

Makes **initdb** prompt for a password to give the database superuser. If you don't plan on using password authentication, this is not important. Otherwise you won't be able to use password authentication until you have a password set up.

-X *directory*

--waldir=*directory*

This option specifies the directory where the write-ahead log should be stored.

--wal-segsize=*size*

Set the WAL segment size, in megabytes. This is the size of each individual file in the WAL log. The default size is 16 megabytes. The value must be a power of 2 between 1 and 1024 (megabytes). This option can only be set during initialization, and cannot be changed later.

It may be useful to adjust this size to control the granularity of WAL log shipping or archiving. Also, in databases with a high volume of WAL, the sheer number of WAL files per directory can become a performance and management problem. Increasing the WAL file size will reduce the number of WAL files.

Other, less commonly used, options are also available:

-d

--debug

Print debugging output from the bootstrap backend and a few other messages of lesser interest for the general public. The bootstrap backend is the program **initdb** uses to create the catalog tables. This option generates a tremendous amount of extremely boring output.

--discard-caches

Run the bootstrap backend with the `debug_discard_caches=1` option. This takes a very long time and is only of use for deep debugging.

-L *directory*

Specifies where **initdb** should find its input files to initialize the database cluster. This is normally not necessary. You will be told if you need to specify their location explicitly.

-n

--no-clean

By default, when **initdb** determines that an error prevented it from completely creating the database cluster, it removes any files it might have created before discovering that it cannot finish the job. This option inhibits tidying-up and is thus useful for debugging.

Other options:

-V

--version

Print the initdb version and exit.

-?

--help

Show help about initdb command line arguments, and exit.

ENVIRONMENT

PGDATA

Specifies the directory where the database cluster is to be stored; can be overridden using the **-D** option.

PG_COLOR

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

TZ

Specifies the default time zone of the created database cluster. The value should be a full time zone name (see Section 8.5.3).

This utility, like most other PostgreSQL utilities, also uses the environment variables supported by libpq (see Section 34.15).

NOTES

initdb can also be invoked via **pg_ctl initdb**.

SEE ALSO

pg_ctl(1), **postgres(1)**, Section 21.1