

Background

This page is for collecting tips on tuning PostgreSQL, especially when using it with Chado. Feel free to place information from your own experience on this page. Unless otherwise indicated the tips on this page refer to PostgreSQL servers in the 8.x series.

General Tips

- [Tuning Your PostgreSQL Server](#)
- [Performance Optimization](#)
- [PostgreSQL performance FAQ](#)
- [PostgreSQL performance mailing list](#)
- [PostgreSQL Hardware Performance Tuning](#)
- <http://www.powerpostgresql.com/docs.html>
- [PostgreSQL performance checklist](#)
- [PostgreSQL annotated configuration option guide](#)

Database initialization

One step that can be taken to optimize your PostgreSQL database is to properly initialize it with initdb. If you are using a PostgreSQL server that came with your OS or have installed it from some package system it may have already been initialized. In these cases you are probably better off removing the PostgreSQL data directory and reinitializing it manually.

To do this you need to execute the following command as the postgres user, or whatever user you use to run postgres under.

```
initdb -D /var/lib/pgsql/data --locale=C
```

Please note that the directory location is entirely dependent on your system or your specific setup and should be substituted with whatever convention is used in your situation. What this command does is explicitly initialize the PostgreSQL database with a C locale. If you do not specify the C locale it defaults to whatever your system is set to, which is set to 'en_US.UTF8' on many linux distributions. In these cases, queries with "LIKE" will take a performance hit because PostgreSQL will not use the indices as you would expect it to. Setting it to the C locale from the start eliminates this.

From the Locale Support page in the manual:

The drawback of using locales other than C or POSIX in PostgreSQL is its performance impact. It slows character handling and prevents ordinary indexes from being used by LIKE. For this reason use locales only if you actually need them.

If for some reason you cannot reinitialize your database you can also drop specific indexes that involve text fields and recreate them using operator classes.

- <http://archives.postgresql.org/pgsql-performance/2007-05/msg00117.php>

One other option you should consider is what the default encoding for all your databases should be. If your data requires UTF-8 or just plain ASCII characters it might be a wise choice to specify that via the `initdb` command:

```
initdb -D /var/lib/pgsql/data --encoding=UTF8 --locale=C
or
initdb -D /var/lib/pgsql/data --encoding=SQL_ASCII --locale=C
```

If you do not do this you need to be sure that you specify the correct encoding when you create the database instance for Chado via `createdb`.

Configuration options

I recommend starting with [Tuning Your PostgreSQL Server](#). Take your time and READ all the options before you start tinkering. Tuning PostgreSQL configuration options requires some serious voodoo and patience. In addition to the links above you might find useful information on these pages.

- <http://www.postgresql.org/docs/current/static/runtime-config.html>
- <http://www.postgresql.org/docs/current/static/kernel-resources.html#SYSVIPC>

Vacuuming

Vacuuming your database after loading or after a large update is very important to database performance. To vacuum your database issue this command

```
vacuumdb -d DBNAME -f -z -v
```

See `vacuumdb --help` for more options. Doing a verbose vacuum is important because it will sometimes inform you when configuration options needs to be increased.

Auto vacuuming

PostgreSQL 8.1 and up has the ability to [automate vacuuming](#). Users who are in read/write environments should utilize this for maintaining performance over the long term.

This section should be expanded with information from Chado users who actually use auto vacuuming.

Powering GBrowse

Some useful email threads on improving performance when using [Chado](#) to power [GBrowse](#):

- [Full Text Search in PostgreSQL, 2010/05](#)

Installing Schema

This is technically not a postgres performance tip, but an ontology loading performance tip. For those who

regularly deploy chado database, [Precompiled Chado Databases](#) are available.