

Preparing Fedora Core 3 (FC3) to Run CONCEPT

Install/Configure Linux

Either at install time via the "Customize Software Packages to be installed" radio button, or via the Add/Remove Applications tool later, do the following:

- Select *only* the perl-DBD-Pg selection from the PostgreSQL Database group. Un-select the rest.
- Check which Development Tools packages are selected. FC3 appears to install most of the needed tools by default. The Proj and Geos tools will use a Fortran compiler if it's available, so select gcc-g77. A workstation (as opposed to desktop, server or custom) installation of FC3 will make the right choices here automatically. Otherwise, see the *Notes at the end of the document.

Build and Install the Tools Used by CONCEPT

Besides the CONCEPT model, you'll need the following:

postgresql 8.0.1* source from

<http://wwwmaster.postgresql.org/download/mirrors-ftp?file=v8.0.1%2Fpostgresql-8.0.1.tar.gz>

geos 2.1.1 source from

<http://geos.refractive.net/geos-2.1.1.tar.bz2>

proj 4.4.9 source from

<ftp://ftp.remotesensing.org/pub/proj/proj-4.4.9.tar.gz>

postgis 0.9.1* source from

<http://postgis.refractive.net/postgis-0.9.1.tar.gz>

The instructions below assume you saved these 4 files in the directory /pg_extras.

As root, add add the line "/usr/local/lib" (without the quotes) to the file /etc/ld.so.conf

At the command line, as a regular user, execute the following commands:

```
cd /pg_extras

tar -xzvf postgresql-8.0.1.tar.gz
cd postgresql-8.0.1
./configure
make
cd ..

tar -xzvf proj-4.4.9.tar.gz
cd proj-4.4.9
./configure
make
cd ..

tar -xjvf geos-2.1.1.tar.bz2
cd geos-2.1.1
./configure
make
cd ..
```

Become root, and execute the following commands:

```
cd /pg_extras
cd postgresql-8.0.1
make install

cd ../proj-4.4.9
make install

cd ../geos-2.1.1
make install
/sbin/ldconfig
```

Become a regular user again and execute the following:

```
cd /pg_extras
tar -xzvf postgres-0.9.1.tar.gz
mv postgres-0.9.1 postgresql-8.0.1/contrib/
cd postgresql-8.0.1/contrib/postgis-0.9.1/
make
```

And as root again, ...

```
cd /pg_extras/postgresql-8.0.1/contrib/postgis-0.9.1/
make install
```

Configure the PostgreSQL Installation

Assume the userid you want to be when running CONCEPT. Find or create an empty directory with lots of free space where you want PostgreSQL to store its data.

Modify your ~/.bashrc to set the environment variable PGDATA to the data directory you chose, and set PGHOME to /usr/local/pgsql/bin (the install described here will put the PostgreSQL binaries there). You can also set the environment variables CONCEPT_HOME, CONCEPT_PROJECTS and update your PATH now. Here are the lines from my .bashrc as an example.

```
export PGHOME=/usr/local/pgsql/bin      # the install directory for PostgreSQL
export PGDATA=/data/pgdata             # the directory where the PostgreSQL data files reside
export CONCEPT_HOME=/data/concept_new/concept # the concept subdirectory of the CONCEPT model
export CONCEPT_PROJECTS=/data/concept_new/concept_projects # the concept_projects subdirectory
export PATH=$PGHOME:$CONCEPT_HOME:$PATH
```

After saving the updated .bashrc, type ". .bashrc" to make the changes take effect.

Test by typing "echo \$PGHOME". You should get back "/usr/local/pgsql/bin".

Still working as the CONCEPT user, type "initdb" to prepare PostgreSQL for use. You should get a message indicating success and telling you how to start the database server. The directory corresponding to your \$PGDATA environment variable should appear in the start message.

Again, taking advantage of having set PGDATA, and assuming you want your Postgres logfile named pg.log in your home directory, start the database server by typing "pg_ctl -l ~/pg.log start".

Test the server by typing "createdb test". You should get the message "CREATE DATABASE".

You should now be ready to install and run CONCEPT.

*Notes:

Since first preparing this document, PostgreSQL 8.0.3 and PostGIS 1.0.1 have been released. I have tested PostgreSQL 8.0.3, which seems to work fine with CONCEPT. I have not tested the newer PostGIS. Based on changed file locations in the PostGIS 1.0 betas, I'd expect to have to make some adjustments.

For other Linux distributions, a similar install process should work. The distribution will likely have a number of PostgreSQL packages. Don't install any of them. If they are already installed, un-install all of them. When you build and install PostgreSQL from sources, you'll get everything that those packages provided. The perl interface and driver, perl-DBI and perl-DBD, would be the possible exceptions here.

Watch the output from the "./configure" steps for missing tools. Distributions with a fairly complete basic installation, like SuSE, won't leave too much for you to have to add. On the other hand, a distribution like Kubuntu will make you add lots of packages to the standard install before everything will build. In either case, the warnings and errors from the ./configure steps will tell you what you need. Once it's added, pick up the build and install process where you left off.