

Preparing Fedora Core 3 (FC3) to Run CONCEPT

Last update: January 16, 2007

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:

- Un-select all postgres related (like the ones installed from source) packages.
- 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 packages:

postgresql source from

<http://wwwmaster.postgresql.org/download>

geos source from

<http://geos.refrations.net>

proj source from

<ftp://ftp.remotesensing.org/pub/proj>

postgis source from

<http://postgis.refrations.net>

DBD-Pg source from

<http://www.cpan.org/pub/CPAN/modules/by-module/DBD/>

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

As root, if it is not already there, 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-8.1.4.tar.gz
cd postgresql-8.1.4
./configure
make
cd ..
```

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

```
tar -xzvf DBD-Pg-1.49.tar.gz
cd DBD-Pg-1.49
perl Makefile.PL
make
make test
cd ..
```

```

tar -xjvf geos-2.2.2.tar.bz2
cd geos-2.2.2
./configure
make
cd ..

cd /pg_extras
tar -xzvf postgis-1.1.1.tar.gz
mv postgis-1.1.1 postgresql-8.1.4/contrib/

```

Become root, and execute the following commands:

```

cd /pg_extras
cd postgresql-8.1.4
make install
cd ../proj-4.4.9
make install
cd ../geos-2.2.2
make install
/sbin/ldconfig
cd ../DBD-Pg-1.49
make install

```

Configure the PostgreSQL Installation

Assume the userid you want to be when running CONCEPT. If the userid does not automatically enter into the bash shell upon log-in then edit (as root) the /etc/passwd file to make 'bash' the default userid shell. Otherwise the commands as listed in this section will not work correctly.

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
export PGDATA=/data/concept
export CONCEPT_HOME=/s1/concept/concept
export CONCEPT_PROJECTS=/s1/concept/projects
export POSTGIS_DIR=/usr/local/pgsql/share/contrib/
export PATH=$PGHOME:$CONCEPT_HOME:$PATH:$POSTGIS_DIR

```

After saving the updated .bashrc, type ". .bashrc" to make the changes take effect, or just log-out and log back in and the shell script will automatically be invoked.

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

The next step is to install POSTGIS. It likes to have all the POSTGRES environment variables set before it installs correctly.

Enter these commands as CONCEPT user:

```
cd /pg_extras/postgresql-8.1.4/contrib/postgis-1.1.1
./configure
vi Makefile.config (*)
```

* Edit the Makefile.config to point to the PROJ and GEOS directories, which are set to be =/usr/local on our system. Make sure all PROJ and GEOS variables that are set to 0 are switched to 1.

Enter these commands as root:

```
make
make install
```

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".

The commands below are necessary to set up concept and PostgreSQL for your userid (in this example 'johndoe') and project (in this example 'test'):

```
psql -c 'create group concept' test
createuser johndoe (not superuser)
concept add_user -n test -u johndoe
```

You should now be ready to copy the CONCEPT source code and input data and run the model.

***Notes:**

As newer versions of these packages are released it is important to keep in mind the directory structure and commands may change.

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, then un-install all of them. When you build and install PostgreSQL from sources, you'll get everything that those packages provided.

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.