

Packaging PostgreSQL on Andrew Linux

Stephen Hathorne
Unix Systems Administrator
Infrastructure, Systems, Applications and Middleware
Carnegie Mellon
slh+@cmu.edu

This document is a tutorial on how to install PostgreSQL on an Andrew Linux System.

Assumptions

This document assumes that you have base familiarity with UNIX Systems Administration, that you have administrative privileges, (a root instance), on the system that need to be modified, and that you have passing familiarity with the Andrew Package System.

Conventions

`$>` Represents a system prompt do not type this.
Items listed after the `$>` prompt are to be typed:

Example:

`$> ls`

This example indicates for the reader to type the **ls** command after the `$>` prompt.

Items listed with borders and gray shaded represent file examples:

```
%define doesreallycoolstuff
```

Items listed in plain Arial Text 12 point font are explanatory in nature, and will comprise most of the document.

Italicized items represent filenames or special instructions

Bold elements generally represent section headings or important topics

Step-by-Step Installation

1. Make a directory `/var/db` and change the permissions to user database.
2. Modify `/etc/package.proto`
 - a. `%define does postgresql`
 - b. `D /var/db`
3. Run `mpp-package` to update the computer to run PostgreSQL
4. Make the `/usr/pgsql/data` directory and set the permissions so that the database user can create database files.
5. Become the database user
6. Start the database for the first time
 - a. `/usr/pgsql/bin/initdb -D/usr/pgsql/data`
 - b. `/usr/pgsql/bin/postmaster -D /usr/pgsql/data &`
7. Create a database user to test the installation
 - a. `/usr/pgsql/bin/createuser userid`
8. Become the new user and create a sample database to test the new installation.

I. Create `/var/db` and change the permissions to user database

The `/var/db` directory a directory required by the Andrew Package system. Once the directory is created we need to give ownership to the *database* user. Since PostgreSQL should not be run as root.

(note to Stephen find out why `/var/db` must exist)

First become root, (or your root instance), and type the following:

```
$>mkdir /var/db
$> chown /var/db
```

Step I is now complete.

II. Modify `/etc/package.proto`

`/etc/package.proto` is a file that is used by the Andrew Package System to update, modify and place files on an Andrew System. In order to have package load the Apache Web Server on an Andrew System we need to add the package define¹

```
%define doespostgresql
```

to `/etc/package.proto`

¹ See Appendix A: Additional Resources for more information on Package

Sample package.proto file:

```
%define doespostgresql  
D /var/db  
%define usesdepot  
%define localdepotdir  
%define contribdepotdir  
%define autolinuxspec  
%include /afs/andrew.cmu.edu/wsadmin/public/src/public.proto
```

Step II is now complete.

III. Run /etc/mpp-package

This will invoke the Andrew Package system to copy, move, and place files on the local disk. Based on your */etc/package.proto* file. Using the one above will cause package to load the Apache Web Server, and the PHP4 module.

```
$>/etc/mpp-package
```

Step III is now complete

IV. Make /usr/pgsql/data and set ownership and permissions

Now that the package update has been run we will need to create the PostgreSQL data directory, and set the ownership and permissions to enable the *database* user to write to this directory structure.

```
$> mkdir /usr/pgsql/data  
$> chown -R database.wheel /usr/pgsql/data  
$> chmod 700 /usr/pgsql/data
```

Step IV is now complete.

V. Become the database user and setup the database

Once the database engine has been installed, we need to initialize it for use. In order to do that, we will need to log in as the user database, and run a few simple programs to create the initial database, start up the database engine, and create a database user for our own purposes.

Note: You should still be using your root instance for this.

```
$> su database
```

```
# Notice the new prompt %
```

```
% /usr/pgsql/bin/initdb -D/usr/pgsql/data
```

This command initializes the database and uses `/usr/pgsql/data` as the data directory. You should notice at this time that this is the data directory that we created in a previous step and gave ownership to the user database to. At this time the database is creating its initial file elements in the named data directory.

```
$> /usr/pgsql/bin/postmaster -D /usr/pgsql/data &
```

This command starts the database engine. It should start again automatically on reboot, since part of the install puts a script in `/etc/init.d` (named `postgres`). This script should be able to start, stop, and restart the `postgres` database engine.

```
$> /usr/pgsql/bin/createuser userid
```

```
  Shall the new user be allowed to create databases (y/n) y
```

```
  Shall the new user be allowed to create more users (y/n) n
```

```
CREATE USER
```

The `createuser` command allows the administrator to create users and enable them in the PostgreSQL environment. The system will respond with two (2) questions. The first, "Shall the new user be allowed to create databases(y/n)". If you answer `y` to this question the newly created user will be able to create new PostgreSQL databases. Answering `n`, restricts them to current databases where they have permissions. (For more information on PostgreSQL database administration see Appendix A). The second question, "Shall the new user be allowed to create more users(y/n)" If you answer `y` to this question then the new user will be able to create new users. I generally allow new users to create databases, (depending on the application), but I very seldom allow them to create new users (again depending on the application). If the new user is created successfully, the system will respond with "CREATE USER" in all uppercase. Exit the database id, and your root instance.

```
$> exit;exit;
```

This will return you to your normal login shell.
Step V is now complete.

Vi. Test the new database installation

The final step in installing the PostgreSQL database engine on your system is to test your installation by creating a simple database, creating a table within that database, insert and select data from your newly created table.

```
$> /usr/pgsql/bin/createdb mydbname
CREATE DATABASE
$> /usr/pgsql/bin/psql mydbname
Welcome to psql 8.0.0, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
```

```
mydbname=> create table acme (name varchar(30), latin_name
varchar(30));
CREATE TABLE
mydbname=> insert into acme('Wiley Coyote','Carnivorous Vulgaris');
INSERT 17342 1
mydbname=> select * from acme;
  name  | latin_name
-----+-----
Wiley Coyote | Carnivoruous Vulgaris
(1 row)

mydbname=>\q
$>
```

Step VI is now complete and the PostgreSQL database engine has been installed.

Appendix A—Additional Resources

Additional information on package can be found at:

http://www.cmu.edu/computing/documentation/andrw_package/package.html

Additional information and documentation on Apache can be found at:

<http://acs-wiki.andrew.cmu.edu/twiki/bin/view/Andrewenv/SoftwareHowTos>

<http://www.postgresql.org>