

Postgres OnLine Journal: November 2009 / December 2009

An in-depth Exploration of the PostgreSQL Open Source Database



Table Of Contents

From the Editors

Year in review

What's new and upcoming in PostgreSQL

PostGIS does Geography

Basics

An almost idiot's guide to Install and Upgrade to PostgreSQL 8.4 with Yum *Beginner*

Compiling PostGIS 1.5 and installing after Yum PostgreSQL Install *Intermediate*

Using PostgreSQL Contribs

PgAdmin III Plug-in Registration: PostGIS Shapefile and DBF Loader *Beginner*

Product Showcase

Wink: Making screencast tutorials

Reader Comments

A Product of Paragon Corporation

<http://www.paragoncorporation.com/>

<http://www.postgresonline.com/>

Year in review

This was a truly exciting year for us and the PostgreSQL project and perhaps a bit depressing for MySQL.

The following events happened:

- PostgreSQL 8.4 was released which had blow away features like Common Table Expressions (CTE) , Recursive CTEs, and Windowing Functions. This meant we could finally get some of our hard-core Oracle and SQL server friends really excited about PostgreSQL.
- This is the first year we got out of our shy mode and actually presented at conferences. We presented at PGCon 2009 and OSCON 2009.
- The **PostGIS project steering committee** was formed with Regina as one of the founding members
- We started writing our **PostGIS in Action** due out sometime in 2010. Sadly we are a bit behind schedule, but on the bright side, you can buy the book now and it will probably be a bit heftier than the 325 pages we had planned. To celebrate our upcoming book, we have launched our book promo site **PostGIS in Action: The Book where the adventure begins**. There you will find source code downloads, data, presentations as we put each together. You will also see a **brief description of chapters** , our progress with each chapter, what you can expect from each chapter, and related links to the chapter content. We are currently at what we hope is our last quarter sprint.
- We wrote a **DZone cheatsheet** which was confronted with mixed emotions.
- 2009 was also the year Oracle threatened to buy Sun and engulf MySQL in the process. Interestingly this was predictable in **someone's wildest dreams**. Is this the end of Open source databases as we know it? Only time will tell.

Plans for 2010

What are our plans for 2010?

- Get PostGIS 1.5 out the door some time in January 2010
- We hope in 2010 to present at at least one PGCon conference and hopefully make **FOSS4G 2010 in September 2010**. Our book better be written by then.
- Finish our book and hopefully soon.
- Increase the adoption of PostgreSQL and PostGIS significantly. To paraphrase our favorite Larry's famous words our strategy is to *Get big very fast*.
- Get PostGIS 2.0 out the door sometime in late 2010.

What will happen to the database industry in 2010

I usually try to keep my mouth shut on these topics. I must say that I have noticed a bit of animosity from some PostgreSQL people toward the whole MySQL/Oracle affair, comments like *He lives by the sword, he should die by the sword*. Other interesting conjectures as to what this means for Open Source databases, Is Monty right that the apparent rape of MySQL by Oracle is only bad and will cause countless pain and suffering for many. All I can say is "What..ever".

Some argue that Monty's fight is all about money and some don't that he is earnestly trying to save the world from Oracle. To me its a fight about a man who has spent half his life nurturing this child MySQL named after his own son. Of course he has quite a bit of emotional attachment to it, as many in the PostgreSQL clan have an equal attachment to PostgreSQL and we have an equal non-economic (as well as economic) attachment to PostGIS and PostgreSQL. Equally so I'm sure Larry Ellison has perhaps a stronger attachment to the namesake Oracle database named after a [CIA project he spent more than half his life nurturing](#).

So in short the motives on all side are clear and irrelevant to all except said people with said motives. In the end, what is relevant is what is relevant.

[Back to Table Of Contents](#) [Year in review](#) [Reader Comments](#)

PostGIS does Geography

The upcoming version of PostGIS - PostGIS 1.5 will be an exciting one. It has native geodetic support in the form of the new geography type, similar in concept to SQL Server's geography support. For windows users, we have [experimental binary builds](#) hot off the presses for PostgreSQL 8.3 and 8.4

[Paul](#) has more information about the technical details of it and an [example use case](#). At this point the geography portion of PostGIS 1.5 is pretty much complete and we are now stress testing the functionality we have in place. We have native area, length, perimeter, indexable proximity distance checks based on a geodetic model and a slightly more advanced spatial index support algorithm. So right now it takes care of the number one fundamental question of casual GIS users too short on time to learn about spatial reference systems, spheroids and geoids. That being *How do I get fairly accurate proximity and measurement information from long lat data and do it fast and quick?*. This also paves the way for PostGIS 2.0 where we will be extending 3D support since some of the lessons learned will be applied to that.

Although the geography data type doesn't have all the jazzy geometric processing functions of the geometry type, casting to the geometry type is easy and, creating them for most common use cases is fairly trivial as Paul has demonstrated with his `ST_Buffer` example. We had lots of heated debates about this because these tricks don't work all the time and some of us (both on dev and long time postgis users) were concerned about the corner cases where these tricks start to show their sores and polluting our really good native geodetic functions with these questionable workarounds. [With Paul being on the side of "think of the children" and many of us being on the side of "think of the adults"](#). We finally did approach each other and come to some sort of truce that we'd allow Paul to put some of his frosting in and put aside the rest of it for people to decide how much of a helping they wanted.. We didn't quite get to the point of virtual fist fights this time around.

We'll be doing another comparison in the coming months on our [Boston GIS](#) site comparing the geometry and geodetic spatial support of Oracle, SQL Server, and PostgreSQL/PostGIS. I should say there is no clear winner in the geodetic front. SQL server 2008 has more native geodetic functions, but also more restrictions, and the fact you can't easily transform to another spatial ref puts it at a bit of a disadvantage to make use of the extra geometry functions it has. On the up side it has a lot more native geodetic functions that are not trivial to build. In so doing, it does put a restriction on the size of the geometry you can have which PostGIS doesn't. Oracle locator has a lot fewer functions period (and Oracle enterprise is just too pricey), but locator does do some things really really well.

[Simon Greener](#) did a talk at FOSS4G 2009 comparing Oracle and PostGIS/PostgreSQL. His video press should be coming out soon. [The accompanying video is here](#) <http://blip.tv/file/2795843/>, and he has posted his slides which are pretty interesting as they expose certain idiosyncracies of both, not just from the spatial side but from the core database functionality side as well. Its worth a read even if you don't care about spatial.

[Back to Table Of Contents](#) [PostGIS does Geography](#) [Reader Comments](#)

An almost idiot's guide to Install and Upgrade to PostgreSQL 8.4 with Yum *Beginner*

In this exercise, we'll go thru our steps for upgrading a Redhat Enterprise Linux 64-bit PostgreSQL box from PostgreSQL 8.3 to PostgreSQL 8.4. If you don't have any kind of PostgreSQL installed on your box, you can skip the Upgrade step.

Upgrading from PostgreSQL 8.* to PostgreSQL 8.4

If you are starting from scratch -- just skip this section.

If you are upgrading from 8.4.0 to 8.4.1 you can get away with a simple

```
yum update postgresql
```

and skip the rest of this article.

If you are upgrading from PostgreSQL 8.3 to 8.4, in theory you can use PgMigrator, but in practice, particularly with a Yum install, you are bound to run into obstacles. If you are running an older version, you must dump and restore.

PgMigrator difficulties

For starters we did try upgrading our database using the PgMigrator but were unsuccessful. Keep in mind these comments are not flames -- just things we see that are obstacles to upgrade and perhaps can be made less painful for future migration work and we consider important to increase PostgreSQL adoption.

This process turns out to be a bit more difficult if you are using Yum for a couple of reasons.

1. Yum and I think most distros always install in the same location -- the bin folder and bin/pgsql so its hard to tell which files you need and pgMigrator requires both the new binaries and the old binaries to complete. To work around that problem -- we did compile from scratch a PostgreSQL 8.3 that matched our 8.3 version by downloading the source and compile and setting --prefix== another location just to make the migrator happy.
2. pg_migrator was not happy with the location of pg_migrator.so -- this is probably a 64-bit issue and one we run into a lot. No biggies just symlink

```
ln -s /usr/lib64/pg_migrator.so /usr/local/pgsql84/lib/pg_migrator.so
```
3. The third obstacle we ran into was an insurmountable one. Or rather one we didn't really care to tackle because we figured trying to work around it would bite us in the future and also prevent us from using Yum which we really like using by the way. This is that the default date time storage between 8.3 and 8.4 has changed. So it seems to be able to migrate or at least on our 64-bit Linux box, we would need to recompile the PostgreSQL 8.4 to ignore which sounded like a future management nightmare waiting to happen. The error you get is this **Old and new pg_controldata date/time storage types do not match. You will need to rebuild the new server with configure --disable-integer-datetimes or get server binaries built with those options.**

I suspect this will become a non-issue in PostgreSQL 8.4 to 8.5

Please see [Greg's notes about suggestions for this issue](#)

Using the tried and true dump/restore but slower more space needed

We were upgrading 800 gb database which reduces down to a 80gb backup file. Space was a concern, but this is a dev box we wanted to blow out and have clean anyway. We had a good last night backup from production so we were in good shape to uninstall and just remove our data folder.

1. Note location of your old data cluster:

```
psql -h localhost -U postgres -p 5432
SELECT setting FROM pg_settings WHERE name = 'data_directory';
```

Gives something like:

```
          setting
-----
/var/lib/pgsql/data/
```

2. **Make sure to backup at least the .conf files to different location which are located in same folder as data and also make backup of users and other global settings.**

```
mkdir /pgbak
cp /var/lib/pgsql/data/*.conf /pgbak
pg_dumpall -h localhost -p 5432 -U postgres --globals-only > /pgbak/globals.sql
```

3. Make backup of each database or whole server. We liked compressed backups of each db since we don't always have space for a non-compressed and sometimes want to only selectively restore part of a db. Repeat the below for each db.

```
          /usr/bin/pg_dump -i -h localhost -U postgres -F c -b -v -f "/pgbak/
mydb_beforeupgrade.backup" mydb
```

4. Shut down the old service.

```
service postgresql stop
```

5. Uninstall it.

```
yum erase postgresql
```

You should get a screen that looks something like this and choose **y** to uninstall:

Dependencies Resolved

```
=====
Package                Arch      Version                Repository      Size
=====
Removing:
 postgresql            x86_64    8.3.7-1PGDG.rhel5     installed      4.7 M
Removing for dependencies:
 postgresql-contrib    x86_64    8.3.7-1PGDG.rhel5     installed      1.3 M
 postgresql-devel      x86_64    8.3.7-1PGDG.rhel5     installed      4.9 M
 postgresql-plperl     x86_64    8.3.7-1PGDG.rhel5     installed       68 k
 postgresql-server     x86_64    8.3.7-1PGDG.rhel5     installed      12 M
```

Transaction Summary

```
=====
Install      0 Package(s)
Update      0 Package(s)
Remove      5 Package(s)
```

Is this ok [y/N]: y

6. If you have space you can move your old cluster.

```
mv /var/lib/pgsql/data to /var/lib/pgsql/dataold
```

but if not and you know your backup is sound or this is a dev box just destroy it. **You are destroying data so proceed with caution**

```
rm -rf /var/lib/pgsql/data
```

Installing PostgreSQL 8.4.1 from Yum repository

1. Figure out which OS you are on. **Note the pgdn repository only works with CentOS and Redhat 4 and above, or Fedora.**

```
uname -a
```

If you see a el5 you are most likely running an enterprise linux. If you see x64, then you have a 64-bit installation.

```
vi /etc/redhat-release
```

May help narrow it down.

2. Download the right file - from <http://yum.pgsqlrpms.org/reporpms/8.4/>. In our case we were running a redhat EL distro so we downloaded.

```
cd /pgbak
wget http://yum.pgsqlrpms.org/reporpms/8.4/pgdg-redhat-8.4-1.noarch.rpm
```

3. Install the rpm

```
rpm -ivh pgdg-redhat-8.4-1.noarch.rpm
```

4. To get a listing of what is available in the PostgreSQL 8.4.1 yum

```
yum list | grep pgdg84
```

5. Install what you want: The developer package is needed to compile things like PostGIS if you are going to compile your own. the name would be different if you are on a 32-bit box.

```
yum install postgresql-devel.x86_64
yum install postgresql-server
yum install postgresql-contrib
yum install postgresql-plperl
```

6. If you decide to use non-default location for data, you need to edit the postgresql

service sysconfig file and change the PGDATA argument.

```
vi /etc/rc.d/init.d/postgresql
```

```
vi /etc/sysconfig/pgsql/postgresql
```

(For vi you do a I to insert a line -- ESC then :w to save and exit) NOTE: postgresql in sysconfig/pgsql may not exist so you may need to create it and put in a line denoting where you want the data -- something like below

```
PGDATA=/var/lib/pgsql/data
```

7. Create data cluster:

```
mkdir /var/lib/pgsql/data
chown postgres /var/lib/pgsql/data
su postgres
initdb -D /var/lib/pgsql/data
```

```
service postgresql initdb
```

8. Set postgresql service restart automatically on reboots.

```
su root
chkconfig --list (to see list of services)
chkconfig postgresql on
```

9. Start the service.

```
service postgresql start
```

10. Copy config, Restore accounts -- this is needed only if you were upgrading.

```
cp /pgbak/pg_hba.conf /yourdatacluster/
(NOTE: for postgresql.conf -- you are best manually editing that since its changed considerably
from 8.3 to 8.4)
vi /var/lib/pgsql/data/postgresql.conf (-- use your old file as reference)
psql -U postgres -d postgres -f /pgbak/globals.sql
```

11. Install pgadmin pack if you want to be able to change config settings directly from pgAdmin III

```
psql -U postgres -d postgres -f /usr/share/pgsql/contrib/adminpack.sql
```

12. Restart for whatever config changes you made to take effect

```
service postgresql restart
```

13. If you have database backups, first create a blank db for each and then restore with pg_restore.

[Back to Table Of Contents](#) [An almost idiot's guide to Install and Upgrade to PostgreSQL 8.4 with Yum Reader Comments](#)

Compiling PostGIS 1.5 and installing after Yum PostgreSQL Install *Intermediate*

A couple of people have asked this, so thought we would address the common issues people run into with compiling PostGIS.

The first question, some will ask is **Why don't you just take the packaged PostGIS that comes with the PostgreSQL Yum repository?**

There are two reasons for not installing PostGIS under using Yum and our general reasons for not.

- We like running a newer GEOS than what is generally packaged with Yum because certain PostGIS functionality is not enabled with older versions of GEOS. For example GEOS 3.1 introduced a couple of new functions like ST_Covers and ST_CoveredBy, ST_ContainedProperly the **Cascaded Union** and **prepared geometry** speed enhancements require GEOS 3.1 +. GEOS 3.2 brought more robust support for dealing with topological exceptions, **new buffering features like one-sided buffer** (which the upcoming PostGIS 1.5 exposes if and only if you are running GEOS 3.2+), faster buffering, and several memory leak cleanups. We highly recommend using GEOS 3.2.0. It is just a more robust product than prior GEOS versions.
- We generally like to run newer versions of PostGIS than what Yum provides and also run multiple versions of PostGIS on the same server. This is a bit tricky to do with Yum since it only exposes one and generally an older one than what we care for. PostGIS 1.5 will be especially great since it will have the **geodetic support** and cool functions like **ST_ClosestPoint** and **other major goodies**.

NOTE: PostGIS 1.3, PostGIS 1.4, PostGIS 1.5 can coexist on the same PostgreSQL daemon service as long as they are installed in different databases. They will all share the same Proj and GEOS. So installing PostGIS 1.5 will not break your PostGIS 1.4 or 1.3 installs. The newer GEOS 3.2.0 C-API is backward compatible with older GEOS C-API and the C-API is what PostGIS uses.

Now that we answered the why, the next is **How?**. For this How section, we'll outline the things you need to compile first which are **NOT** packaged with PostGIS. We will also be focusing on the upcoming PostGIS 1.5. We will also outline the common issues people run into. Issues are most common on a 64-bit Linux Centos/Redhat EL.

Basic overview steps

1. Install PostgreSQL using Yum as we outlined in **PostgreSQL 8.4 Yum Install**. Also remember to yum install postgresql-dev.
Do not yum install postgis.
2. download, compile, install proj 4.6+, don't forget nad datum shift folder
3. download, compile, install geos 3.2 from <http://trac.osgeo.org/geos/> . Note seem to run into fewer linking issues if you download from SVN, but then you have to do a yum install svn to install subversion client.
4. download, compile, install libxml 2.6+ - <ftp://xmlsoft.org/libxml2/libxml2-2.7.6.tar.gz> (this is a new dependency for PostGIS 1.5 to support the ST_GeomFromGML and ST_GeomFromKML functions). It is not required for older versions of PostGIS.
5. download, compile, install postgis <http://www.postgis.org/downloads>
6. Fix geos / proj linking errors if you run into them. This seems to be more of an issue with 64-bit Linux than any other we have run into. We'll go over how to discover missing links and fix them.

Preprep

Yum Installs you need. As mentioned you need to Yum install PostgreSQL and postgresql-devel.

Missing G++, GCC

If you run into issues like complaints about no gcc+ compiler, then you probably need to:

```
yum install gcc-c++
```

```
mkdir /sources
cd /sources
```

Download and install Proj

```
cd /sources
mkdir proj
cd proj
wget http://download.osgeo.org/proj/proj-4.6.1.tar.gz
tar -xvf proj-4.6.1.tar.gz
wget http://download.osgeo.org/proj/proj-datumgrid-1.4.zip
unzip proj-datumgrid-1.4.zip -d proj-4.6.1/nad
cd proj-4.6.1
./configure
make
make install
ldconfig
```

Download and install GEOS

```
cd /sources
mkdir geos
cd geos
wget http://download.osgeo.org/geos/geos-3.2.0.tar.bz2
tar -xvf geos-3.2.0.tar.bz2
cd geos-3.2.0
./configure
make & make install
make check
ldconfig
```

Download and install libxml

This is only needed for PostGIS 1.5+

```
cd /sources
mkdir libxml
cd libxml
wget ftp://xmlsoft.org/libxml2/libxml2-2.7.6.tar.gz
tar -xvf libxml2-2.7.6.tar.gz
cd libxml2-2.7.6
./configure
make & make install
ldconfig
```

Download and install PostGIS 1.5 SVN

You may swap this out with PostGIS 1.4.1 for mission critical production use, but we are demonstrating PostGIS 1.5 SVN. This is going to be a really exciting release once [we get all our ducks in a row](#) hopefully by early to Mid January 2010.

```
cd /sources
mkdir postgis
cd postgis
wget http://www.postgis.org/download/postgis-1.5.0SVN.tar.gz
```

```
tar -xvf postgis-1.5.0SVN.tar.gz
cd postgis-1.5.0SVN
./configure
make
make install
make check
ldconfig
```

Problems

Its very common to run into linking issues particularly with GEOS or Proj on a 64-bit that its either pointing at an old version or non-existent location. To detect said linking issues do the following:

```
ldd -d /usr/lib64/pgsql/postgis-1.5.so (to show all the dependencies (or ldd -d /usr/lib/pgsql/
postgis-1.5.so if you are on 32-bit)
ldd -d /usr/lib64/pgsql/postgis-1.5.so | grep libgeos (or ldd -d /usr/lib/pgsql/postgis-1.5.so
if you are on 32-bit)
ldd -d /usr/lib64/pgsql/postgis-1.5.so | grep libproj (or ldd -d /usr/lib/pgsql/postgis-1.5.so
if you are on 32-bit)
```

You should get something of the form below which would tell me my geos is pointing at an older version

```
libgeos_c.so.1 => /usr/lib64/libgeos_c.so.1
libgeos-3.1.0.so => /usr/local/lib/libgeos-3.1.0.so (0x00110000)
```

To fix, may not be the best of solutions, but I destroy the symlinks and relink them something of the form:

```
rm /usr/lib/libgeos_c.so.1
ln -s /usr/local/lib/libgeos_c.so.1.6.0 /usr/lib64/libgeos_c.so.1
ln -s /usr/local/lib/libproj.so /usr/lib64/libproj.so.0
```

Installing PostGIS in a db

The insall path will be changing shortly -- but you can always find the sql files in the source folder: Here we are creating a template_postgis15.

```
createdb template_postgis15 -U postgres
cd /sources/postgis/postgis-1.5.0SVN
psql -d template_postgis15 -U postgres -c "CREATE LANGUAGE plpgsql"
psql -d template_postgis15 -U postgres -f postgis/postgis.sql
psql -d template_postgis15 -U postgres -f spatial_ref_sys.sql
#this installs the help description for each function you can search via psql or pgadmin
psql -d template_postgis15 -U postgres -f doc/postgis_comments.sql
```

To verify all is good, run the below command in the database you just created using psql or PgAdminIII

```
SELECT postgis_full_version();
```

You should get an output something like:

```
POSTGIS="1.5.0SVN" GEOS="3.2.0-CAPI-1.6.0" PROJ="Rel. 4.6.1, 21 August 2008" LIBXML="2.7.6"
USE_STATS
```

[Back to Table Of Contents](#) [Compiling PostGIS 1.5 and installing after Yum PostgreSQL Install Reader Comments](#)

PgAdmin III Plug-in Registration: PostGIS Shapefile and DBF Loader *Beginner*

PgAdmin 1.9+ has a simple Plug-In architecture which makes it relatively simple to introduce new plugins. It is pretty much all controlled by the file *plugins.ini*. In that file you can register any executable you want in there. If you want the executable to get passed database configuration settings, there is an option for that and you just have to build your executable to accept commandline switches.

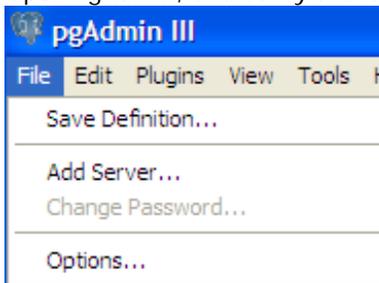
In PostGIS 1.4, [Paul Ramsey](#) created this nice Graphical User Interface to the PostGIS shapefile loader, because well, not everyone likes command-line interfaces, particularly people new to PostGIS. All was great except I couldn't get the thing to compile under windows. After much whining on my part, Paul being the gracious guy he is got off his [Mac soapbox](#) and disgusted himself by debugging this thing in his slow and clunky windows VM.

After a lot of sweat and tears, we have a GUI for the upcoming PostGIS 1.5 that not only is an easy to use non-threatening to newbies interface, but that also fits in nicely as a Plug-in for PgAdmin III (thanks to my whining). **PostGIS 1.5 is eminent once we stop fighting over the whole PostgreSQL PGXS annoyance and how to hack or not hack around it. This whole argument really deserves its own Dilbert comic strip.. Although the GUI is designed for PostGIS 1.5, you can use it to load PostGIS 1.3 and PostGIS 1.4 data as well. Mark Cave-Ayland threw in a nice loading touch to it too. It also adds to the mix, ability to load into the new geography type introduced in PostGIS 1.5. The commandline version also has geography type support.**

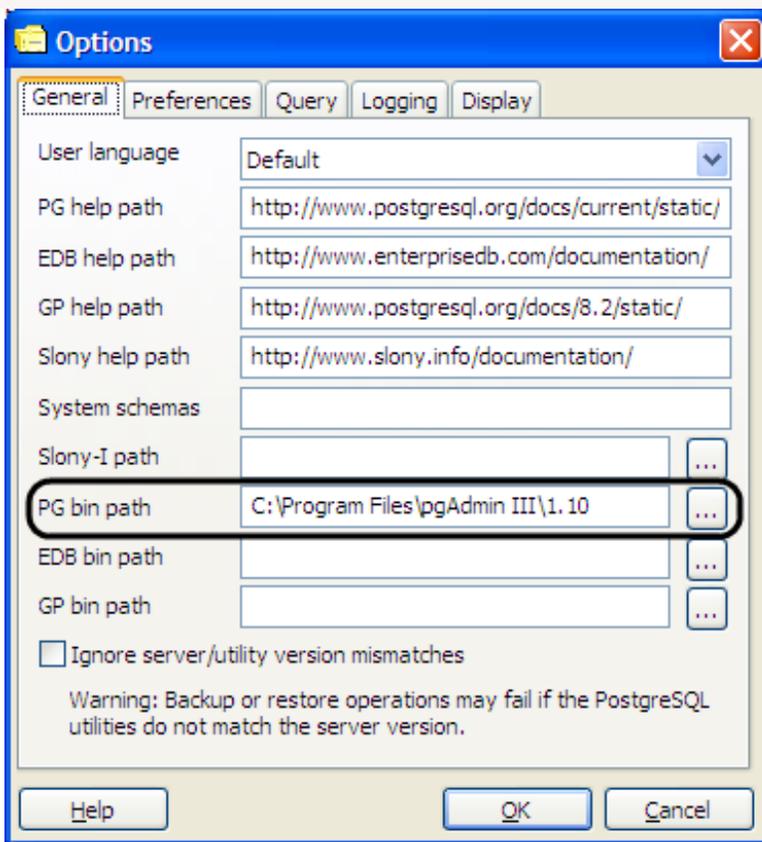
You can download the windows version from <http://www.postgis.org/download/windows/experimental.php>. If you just want it without the PostGIS 1.5 binaries -- just download the one labeled *PostGIS ESRI Shapefile GUI*.

To enable said plugin, do the following:

1. Open PgAdmin, and verify the location of the bin folder by choosing Options from the File menu



2. You may want to change the location if its pointing at an old PostgreSQL or PgAdmin III install.



3. Copy the postgisgui into the bin folder noted in the above dialog.
4. You can manually edit the plugins.ini file in your PgAdminIII install folder by adding these lines which are in the readme or overwrite the plugins.ini with the packaged one assuming you have no custom items in plugins.ini:

```

;
; PostGIS shp2pgsql-gui (Windows):
;
Title=PostGIS Shapefile and DBF loader
Command="$$PGBINDIR\postgisgui\shp2pgsql-gui.exe" -h "$$HOSTNAME" -p $$PORT -U "$$USERNAME" -d "$$DATABASE" -W "$$PASSWORD"
Description=Open a PostGIS ESRI Shapefile or Plain dbf loader console to the current database.
KeyFile=$$PGBINDIR\postgisgui\shp2pgsql-gui.exe
Platform=windows
ServerType=postgres
Database=Yes
SetPassword=Yes

```

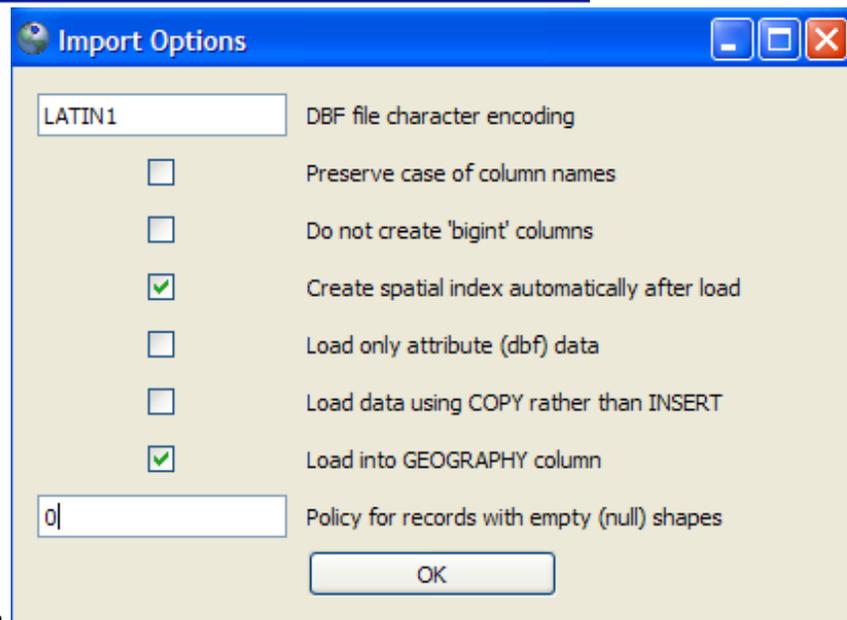
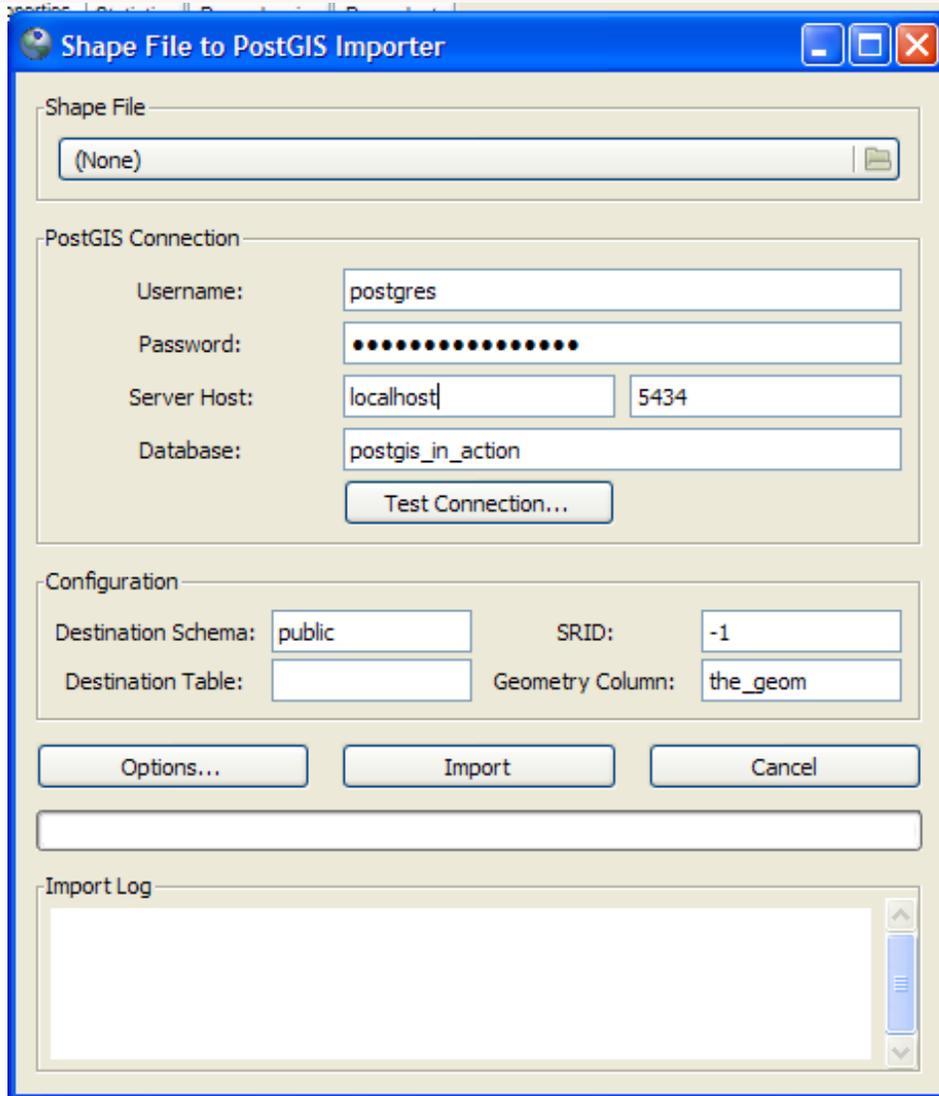


```

testpostgis15_geo
10846783
postgres

```

5. Reopen PgAdmin and your Plugins menu should look like: `ce` `pg_default`
6. If you don't have a database selected, the menu options will be disabled, but once you have a database selected, then you should be able to click the loader icon and get a screen like this.



7. If you click the Options button, you can fine tune things a little, like for long lat data choosing to load in the new geography data type.

NOTE: That for **geography** your data must already be in WGS 84 long lat (4326) which is the only spatial ref supported for geography since transform support is not currently built into the PostGIS loaders. For data you get from **Tiger** and **Census boundary files** (although they are in NAD 83 long lat (4269), you can get away with lying and calling it 4326 for most use cases, since they are almost identical in spheroid, datum etc to the WGS 84 one)

Wink: Making screencast tutorials

Debugmode Wink is a freeware piece of software for both business as well as personal use for doing [screencasts](#) and incorporating sound into your screen captures. The main useful format it outputs to is macromedia flash though you can output to PDF for handouts and so forth. While its not a PostgreSQL related item per se, it can be useful for making all sorts of tutorials including PostgreSQL tutorials that involve showing people how to do things on screen. We've started to experiment with tutorials of this format and hope to get into doing more hands on like tutorials. We'll probably be doing a using PostgreSQL in OpenOffice tutorial in this issue since it lends itself well to screen casting.

Where to get Wink and OS Support

You can get Wink from <http://www.debugmode.com/wink/>

It comes in two versions. One for Window and one for Linux. Haven't tried the Linux version so can't speak for its merits. From reading it seems the Linux version is a little behind in functionality of the windows one, though hopefully that will change.

Debug mode site has some other interesting video slicing, morphing and framing software also released as freeware.

Wink Features

These are the features of the Windows version 2.0 build

- Fairly intuitive to use and it comes with flash videos generated with Wink showing how to use Wink.
- Screen recording and voice recording. We like to do our screens first and then cut in voice afterward, though you can record at the same time
- The project screen once you have done a recording shows all the frames and you can save as a wink project.
- You can cut out frames you don't like
- Add text boxes, cutesy arrows, add next previous pause buttons that stop the movie at that point waiting for user to click next
- Dub in voice snippets at any frame
- Render to flash and to keep the size down, it allows you to designate the size of color palette and build the ideal color palette.
- If you have two or more wink projects you have done, you can copy frames from one to another simply with copy and paste and thus stitch together many screencasts to make a more lengthy video. I'm thinking this is a great feature if you want to cater your tutorials to a specific audience, you make short reels and stitch together ones that are relevant for each group.

Below is a simple link of a screencast we made for PgCon2009 just in case our computer decided to break on us. [OpenJump demo](#)
This one we didn't have any jazzy sound or other fancy features of Wink. Later tutorials we plan to explore the more advanced features of Wink like sound dubbing and bubble windows and so forth.

[Back to Table Of Contents](#) [Wink: Making screencast tutorials](#) [Reader Comments](#)

Year in review

theusnewsdot.com

PostGIS does Geography

topsy.com

Paul Ramsey

Simon's video is now online: <http://blip.tv/file/2795843/>

dustymugs

I'm wondering if you will eventually do a comparison of the differences between geometry and geography. Basically, why geography if you're already using geometry?

If this is something to be discussed in your book (great thus far), that'd be great to.

Leo Hsu and Regina Obe

Dusty,

Thanks. Actually that should probably be included as part of the PostGIS official reference manual (or at least FAQ wiki) since I think it will be a common question.

Yes we'll be discussing it in our book as well. We have started to cut in geography in our book.

PostGIS

PostGIS 1.5.0 is now available for download. It includes a new feature: support for PostGIS 1.5.0. ...

gis-lab.info

An almost idiot's guide to Install and Upgrade to PostgreSQL 8.4 with Yum

Gurjeet Singh

I think you forgot to advise users to use `pg_dumpall` for dumping global objects; users, tablespaces, etc.

Greg Smith

Step (6) in your procedure is better replaced with this:

```
service postgresql initdb
```

That does the same thing, but it's less typing and you can be sure the directories created will have the right permissions.

Step (7) is really not a good idea. If you want to relocate PGDATA, you want to edit `/etc/sysconfig/postgresql/postgresql` and set it there. Look at the init script and you can see how it includes that file after setting a default value for it in the script. Editing the init script will put you into a position where you can't accept updates to the init script itself from future versions without doing a manual merge. If you just set it in the `sysconfig` file instead, this problem goes away.

As for the integer datetime switch, the only good way to handle this if you want to keep your 8.3 database is to download the source RPMs from the yum repo, tweak them to use the old format, and then build your own RPMs. It's possible to automate that into a script, so while not quite as easy as yum it's at least possible to manage in a way that includes upgrades without too much overhead. For people with a database small enough that they can dump and reload, you're right that they should just suck it up and covert now rather than go through that.

Leo and Regina

Thanks for the tips -- we'll update to reflect.

To Gurjeet I think we did include `pg_dumpall` note in step 2 but perhaps we should put that in bold.

www.pythian.com

The Pythian Blog

Hello, and welcome to the 171st edition of Log Buffer, the weekly review of database blogs. Let's get it going this week with . . . Oracle Uwe Hesse, the Oracle Instructor look at result cache, another brilliant 11g new feat...

tony

thanks great article

Compiling PostGIS 1.5 and installing after Yum PostgreSQL Install

Tom Blackmore

Thanks for the great post. I ran into problems with libxml2 and postgis-1.5. I found a post on another site saying postgis-1.5 can use libxml2-devel. So I yummed it and everything seems to work. It's quite an old version of libxml2 (2.6.26) do you think this will cause problems?

Thanks again for a great post.

Tom

Regina

Should be fine. We aren't using any functionality beyond libxml 2.5 as I recall. Anyrate 2.6+ is definitely okay.

PgAdmin III Plug-in Registration: PostGIS Shapefile and DBF Loader

gis-lab.info

Wink: Making screencast tutorials

sysadmin.camerblog.net

Cd-MaN

I personally prefer BB Flashback Express. It is completely free and worked very well on all the systems I've used it: <http://www.bbsoftware.co.uk/bbflashbackexpress/home.aspx>

www.kotihost.com

madman

What's wrong with plain old video, again?

Noo, we can't use that, it would be JUST TOO COMPATIBLE. We must spread the Flash disease everywhere.

Cd-MaN

What's wrong with "plain old video"? For example the fact that it doesn't offer any controls (like start / stop / pause / volume controls) by default (just see the plain [video] tag from HTML5). Or the video codec "compatibility" (or utter lack of it). For better or worse, Flash is a de facto standard which works almost everywhere (including Linux).

madman

Yeah, these days Flash Mostly Almost Works on non-Windows platforms, if you ignore the occasional rendering quirks. And hey, it even compiles for x86_64!

And WHEN it works, it makes sure to strip you of any privacy, security and features that your browser tries so hard to provide. It makes sure that any web developer touching Flash will very likely screw up because it uses similar, but subtly different, programming practices. And god forbid if your user might want to right click and download the video for later.

But speaking of HTML5, have you actually ever seen a "plain video tag from HTML5"? Adding controls is just 8 keystrokes away, you need to type "controls"! And if you want to support recent browsers (bar IE), you need to add two sources in two different codecs -- just add two elements. The code is still MUCH simpler than a single Flash element (wtf... object? param? embed? and 100 ad hoc attributes that you'll never remember...)

Now obviously I'm overdramatising all of this. Yes, Flash has more market penetration (thank you MSIE!), but if you cater to open source developers then almost anyone will prefer open standards. Unlike your average user, we're happy with [link to download video] if all else fails. If Flash fails, all you get is a crashed browser.