LearnOSM

osm2pgsql

In the previous chapter we saw how to set up Postgresql with PostGIS in Windows and how to set up a database and load it with shapefile data. In order to get OpenStreetMap data into a database, you could get the data in shapefile format and use the shapefile loader, but this may leave you without all the data that you want. In this chapter we will learn how to use **osm2pgsql**, a command-line program for loading raw OSM data into a PostGIS database.

We will go through the steps to set up osm2pgsql on Windows, though the steps should be roughly the same on another operating system, assuming you have set up your PostGIS database(s) correctly.

Get osm2pgsql

To download the windows version of osm2pgsql, navigate your web browser to http://wiki.openstreetmap.org/wiki/Osm2pgsql#Windows.
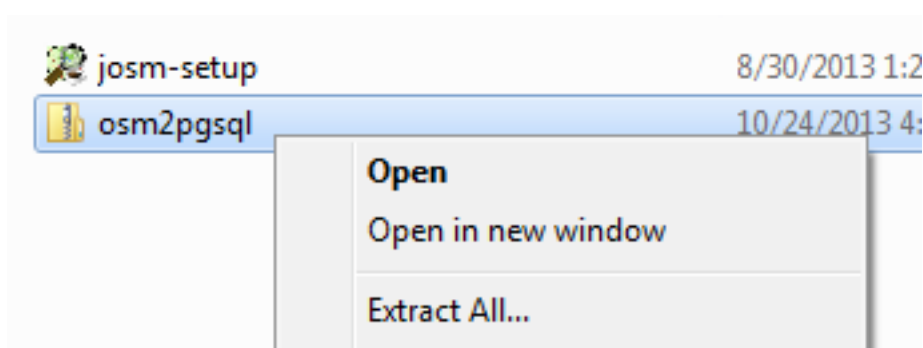
Windows

Binary

This is by far and away the easiest way to get Osm2pgsql running on Windows.

The newest windows binary is from February 6, 2013 at http://customdebug.com/osm/osm2pgsql.zip

This version does not appear to handle all objects properly (64-bit problem), checking git not all 32-bi when this was compiled. See Cygwin below for current workarounds.

It contains 32 and 64 bit binaries with 32 and 64 bit node id (four binaries). It supports pbf and all standard osn **This version has not been tested in a production environment so far.** The linked PostgreSQL version is 9 mapnik 2.

- Download the file named **osm2pgsql.zip**
- Unzip the file on your system. You should move the unzipped folder to a location where you will not move it later, because we need to add its location to the system path.
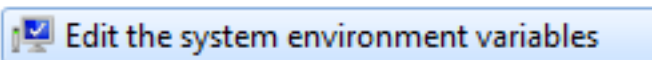
- In the osm2pgsql directory that you unzipped is a file called osm2pgsql.exe. This is a program that we will run to import the data, but in order for Windows to find it, we need to add its location to the system path. Click on the Start Menu and type "system path."
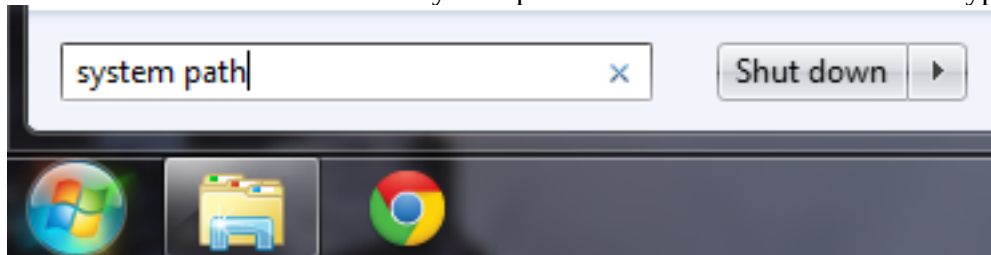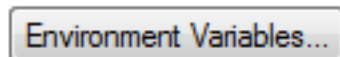

- You should see an option named "Edit the system environment variables." Click on it.
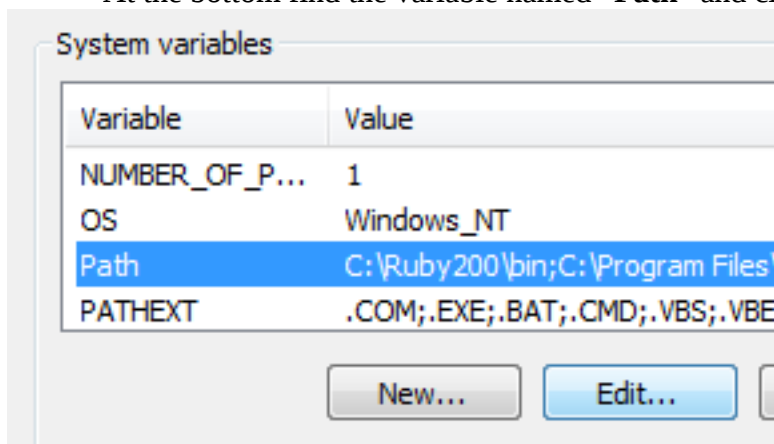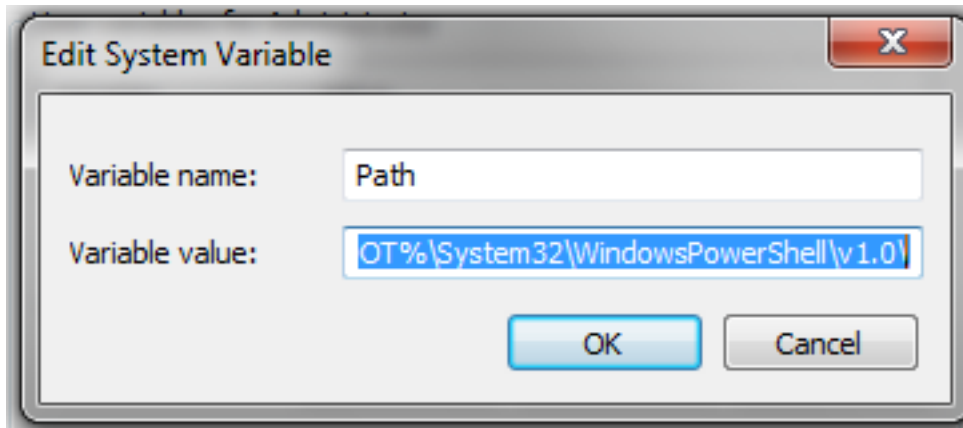

- Click on the button named "**Environment Variables**"

- At the bottom find the variable named "**Path**" and click "**Edit…**"

- You must add the directory where osm2pgsql.exe is located to the Path variable.
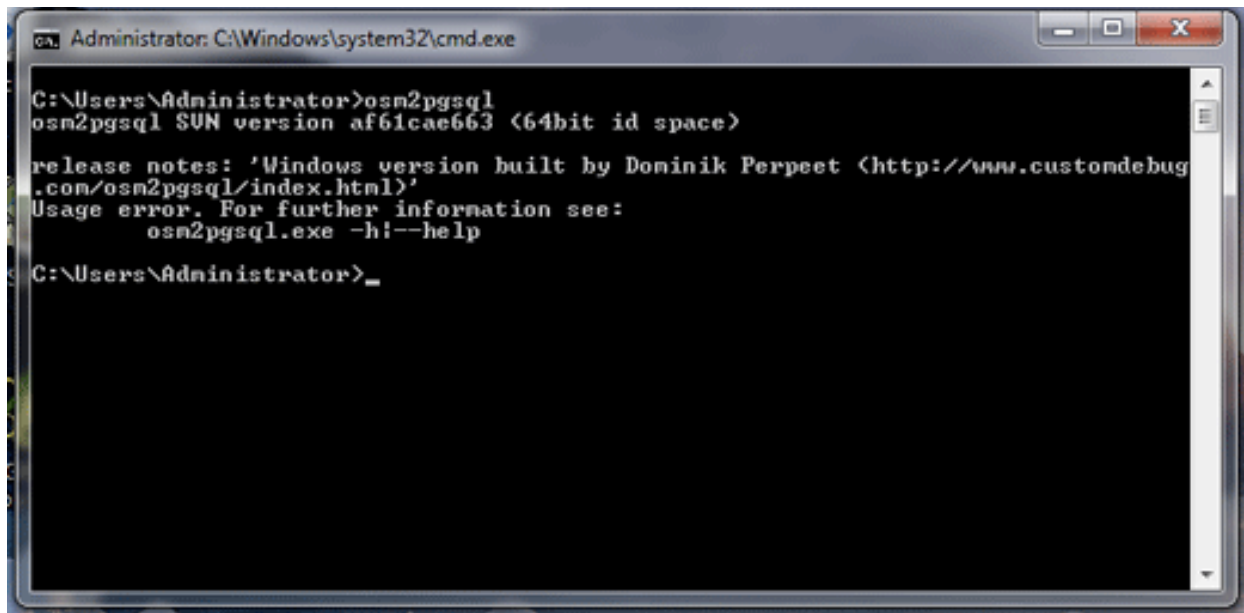
- Add a semicolon to the end of the previous directory and then type in the full directory path of osm2pgsql.exe. For example, if you put the **osm2pgsql** folder directly in the **C:** directory the path would be:

C:\osm2pgsql\Win32

- Click OK several times to save the new settings.
- **osm2pgsql** should be functioning now. Let's check it.
- Open the Windows Command Prompt. You can do this by clicking on the Start Menu and typing "**cmd**". The Command Prompt application will come up and you can press Enter or click on it.
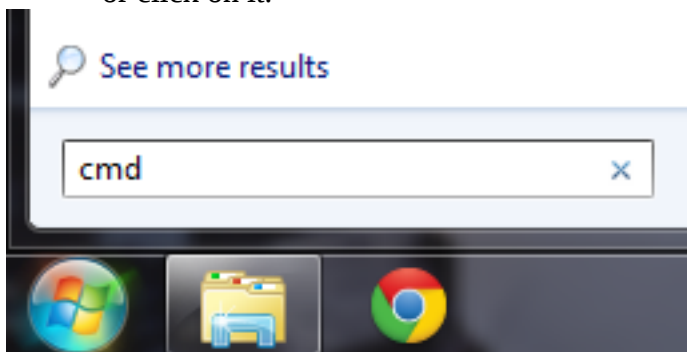
- In the black command window that opens, type:

osm2pgsql

- If everything is working right, you should get a message like this:

- If you don't see an error message like this, and it says that it cannot find the application **osm2pgsql**, then you may have entered the Path variable incorrectly.

Getting Raw OSM Data

Before we can run **osm2pgsql** we need to have some raw OSM data to import into a database. If you don't already have a **.osm** file that you can use, try downloading a file from http://metro.teczno.com/. This site hosts many OSM extracts for different cities. Find a city to import and download the PBF file for it. PBF files are compressed versions of the normal **.osm** files. You can use any of the extract services listed in the chapter on getting data, if you'd like the raw data for another area.

Get the Style File

osm2pgsql requires the use of a custom style file to define which tags are included in the database during import. You can download the default style file here.

Importing the Data

Open PgAdmin III and create a new database named **osm**, just as you did in the previous chapter. To import the data, we will run the **osm2pgsql** program via the command line.
- Click on the Start Menu and type "cmd" and Enter to open the Command Prompt.
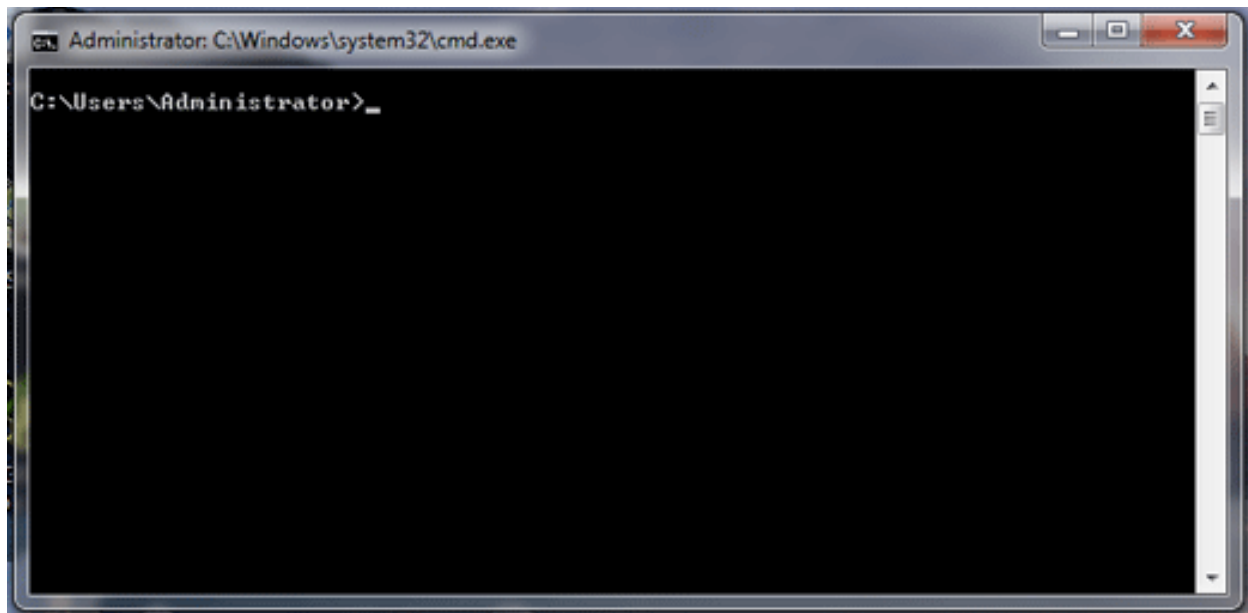
Here we will run the application **osm2pgsql** with several options. We need at least to supply it with:

- The location of the OSM Data File
- The name of the database, and the database username
- The style file which defines which OSM tags will be imported to the database

We have placed our OSM file into the **C:** directory to make this easier.

- Type the following command, replacing the location of the OSM file and style file with your own.

```
osm2pgsql -c -d osm -U postgres -H localhost -S C:\default.style
C:\bangkok.osm.pbf
```

- Press Enter. If all goes well, the process should begin running. It may take a few minutes for all of the data to load into the database.

```
C:\Users\Administrator>osm2pgsql -c -d osm -U postgres -H localhost -S C:\defaul
t.style C:\bangkok.osm.pbf
osm2pgsql SVN version af61cae663 (64bit id space)

release notes: 'Windows version built by Dominik Perpeet (http://www.customdebug
.com/osm2pgsql/index.html)'
Using projection SRS 900913 (Spherical Mercator)
Setting up table: planet_osm_point
NOTICE:   table "planet_osm_point" does not exist, skipping
NOTICE:   table "planet_osm_point_tmp" does not exist, skipping
Setting up table: planet_osm_line
NOTICE:   table "planet_osm_line" does not exist, skipping
NOTICE:   table "planet_osm_line_tmp" does not exist, skipping
Setting up table: planet_osm_polygon
NOTICE:   table "planet_osm_polygon" does not exist, skipping
NOTICE:   table "planet_osm_polygon_tmp" does not exist, skipping
Setting up table: planet_osm_roads
NOTICE:   table "planet_osm_roads" does not exist, skipping
NOTICE:   table "planet_osm_roads_tmp" does not exist, skipping
Allocating memory for sparse node cache
Node-cache: cache=800MB, maxblocks=0*2d, allocation method=102401
Mid: Ram, scale=100

!! You are running this on 32bit system, so at most
!! 3GB of RAM can be used. If you encounter unexpected
!! exceptions during import, you should try running in slim
!! mode using parameter -s.

Reading in file: C:\bangkok.osm.pbf
Processing: Node(864k 864.8k/s) Way(111k 12.37k/s) Relation(550 550.00/s)  parse
 time: 11s

Node stats: total(864790), max(2146144602) in 1s
Way stats: total(111347), max(227317345) in 9s
Relation stats: total(552), max(2997412) in 1s
Committing transaction for planet_osm_point
Committing transaction for planet_osm_line
Committing transaction for planet_osm_polygon
Committing transaction for planet_osm_roads

Writing way (111k)
Committing transaction for planet_osm_point
Committing transaction for planet_osm_line
Committing transaction for planet_osm_polygon
Committing transaction for planet_osm_roads
```

- If your raw OSM file is large, you may need to add additional memory to the osm2pgsql import process. To do this, add the following to the command:
  `--cache 600`

Testing It

We can test that the import was successful and view the data in our database using QGIS.

- Open QGIS and click on the "Add PostGIS Layers" button.
- Under "Connections" at the top, click "New."
- Give the new connection a name. Under database type **osm** (the name of your database).
- Enter the username postgres and your password below.

- Click OK to save the connection settings. Then click "Connect" to connect to your PostgreSQL server.
- Click next to "public" to see all of the layers (tables) in your database. Notice that osm2pgsql creates a separate table for different object types - points, lines, and polygons. It also creates a roads table, which contains only major roads.
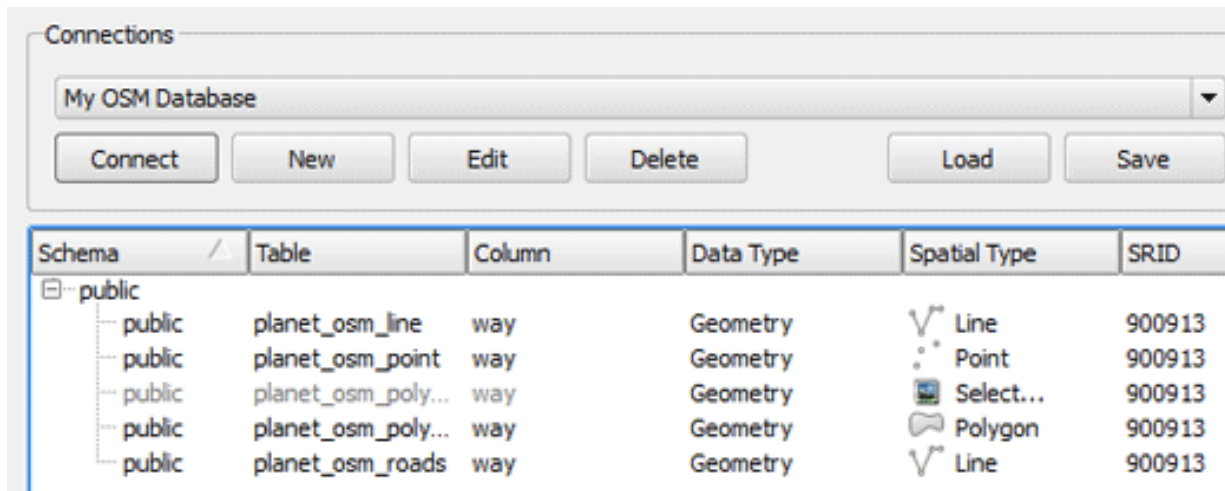
- Select one or more of the layers and click "Add." If asked, choose WGS84 as the CRS.
- If everything is successful, you will see the layers you selected displayed in QGIS.
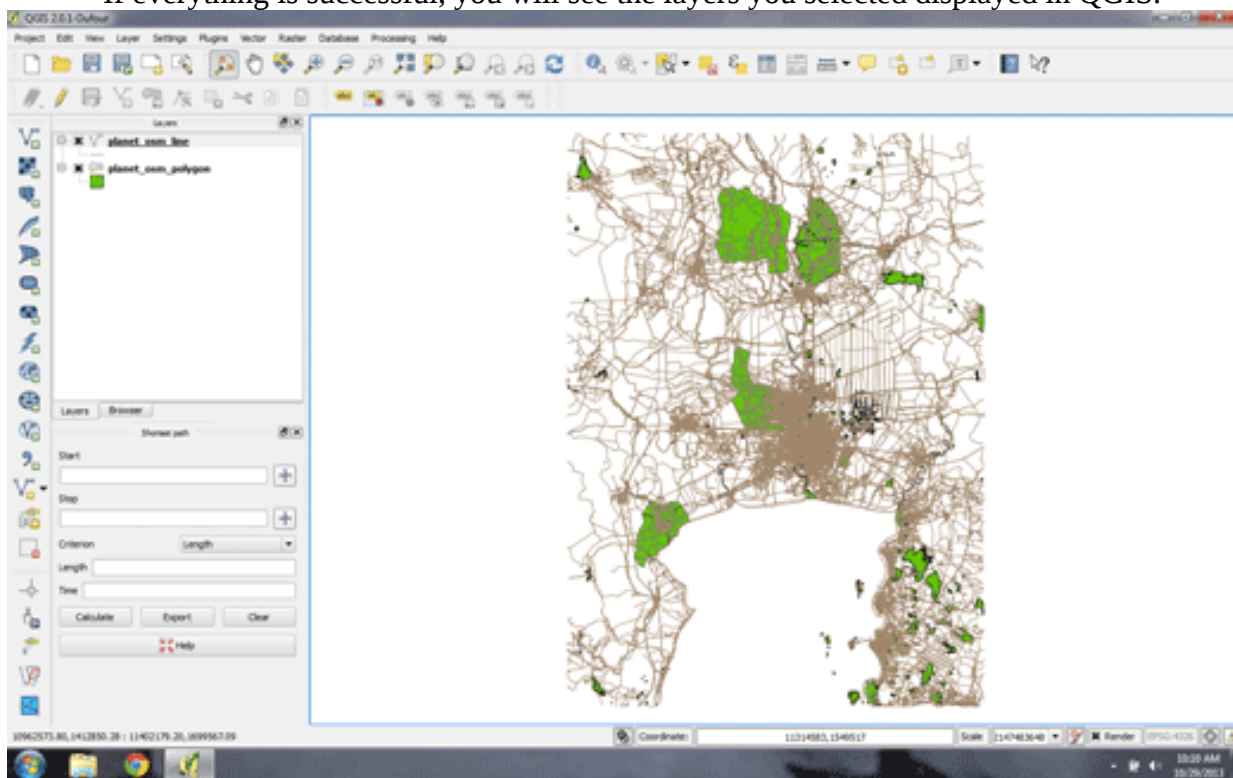
If you look at the attribute tables of the layers, you will see that the attributes are mapped to OSM tags. The specific tags that are imported are defined during the **osm2pgsql** import process. In order to add specific tags that are not included by default, you can edit the *style* file that osm2pgsql references to define the data schema.

Summary

When you want to import OpenStreetMap data into your own database, **osm2pgsql** is a great

8

tool. It can be extremely useful when you need to be able to get the most up-to-date OSM data and customize the attributes you want, or when working on more complex projects.

Another import tool has been developed recently, called imposm, and offers some speed and other improvements over osm2pgsql, although as of this writing it lacks other key functions which are promised in imposom version 3.

For more information on osm2pgsql, refer to the OSM Wiki - http://wiki.openstreetmap.org/wiki/Osm2pgsql.

Was this chapter helpful? Let us know and help us improve the guides!

- learnosm@hotosm.org
- @learnOSM
- Hosted on Github

CC0

Official HOT OSM learning materials

Humanitarian
OpenStreetMap
Team