A framework for user centred privacy and security in the cloud

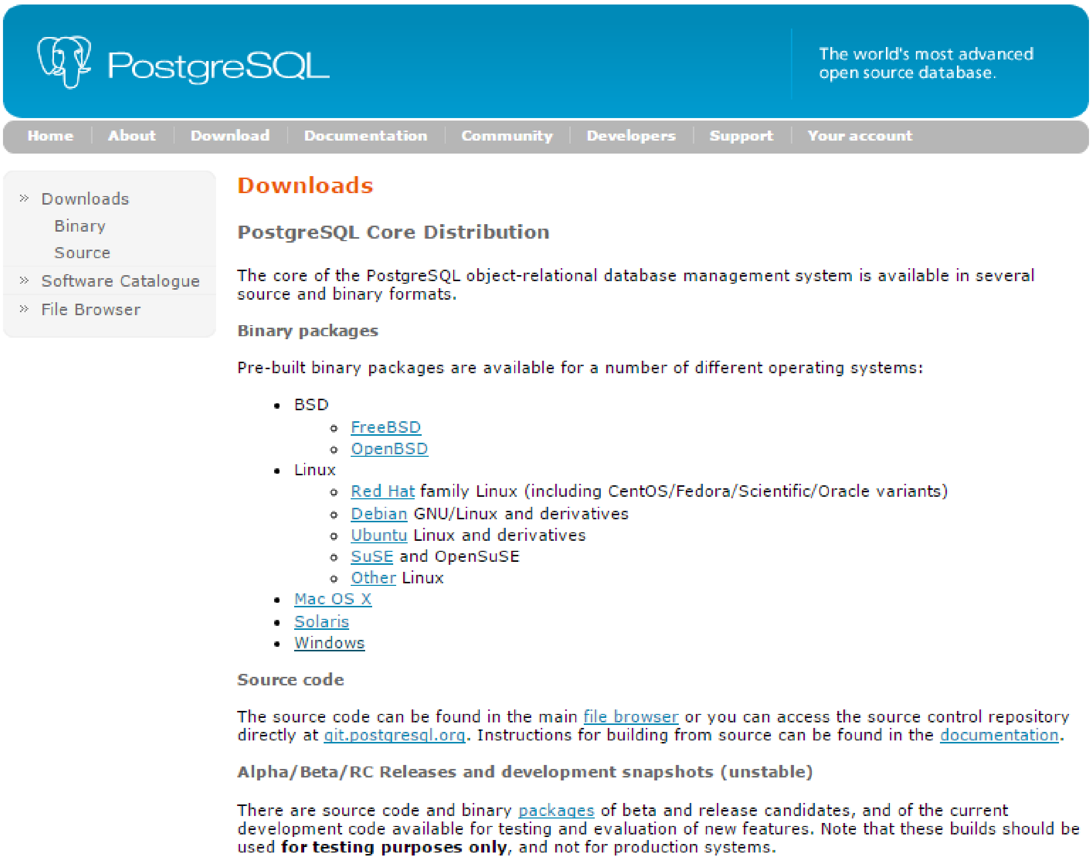
"CLARUS_GeoCollaboration" dataset

Author(s)	AKKA, URV
Document ID	CLARUS-DatasetGeoCollaboration-v1.0
Version	1.0
Date of Issue	15/06/2016
Document Distribution	Public
Abstract	This dataset actually includes two subdatasets: One subdataset represents a gas network and the other one marks some gas leaks on this gas network. The location data is fictive but attribute data are not. Format: ESRI Shapefile Projection: WGS 84 / EPSG: 4326.

Table of Contents

PostgreSQL DBMS installation

We will download the right PostgreSQL installer for the OS from the official site (http://www.postgresql.org/download/).



Downloads

PostgreSQL Core Distribution

The core of the PostgreSQL object-relational database management system is available in several source and binary formats.

Binary packages

Pre-built binary packages are available for a number of different operating systems:

- BSD
  - FreeBSD
  - OpenBSD
- Linux
  - Red Hat family Linux (including CentOS/Fedora/Scientific/Oracle variants)
  - Debian GNU/Linux and derivatives
  - Ubuntu Linux and derivatives
  - SuSE and OpenSuSE
  - Other Linux
- Mac OS X
- Solaris
- Windows

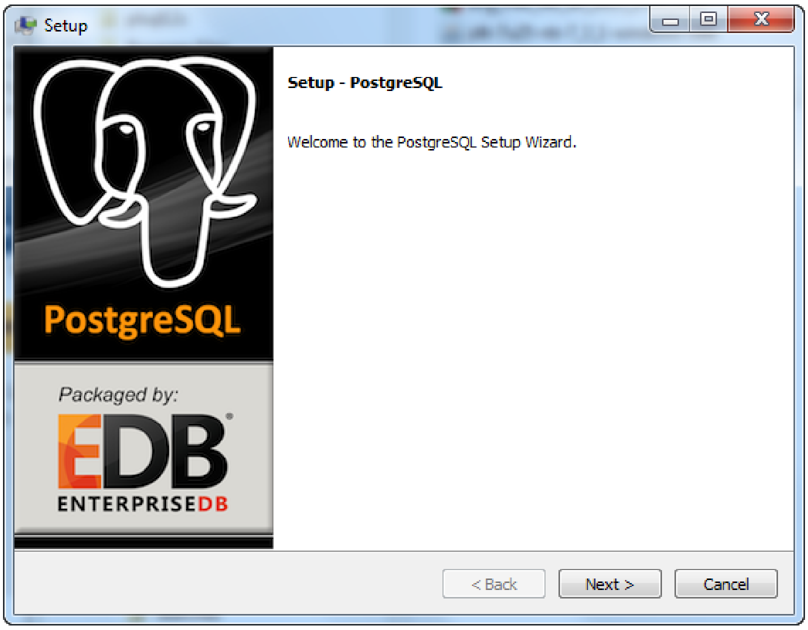
Source code

The source code can be found in the main file browser or you can access the source control repository directly at git.postgresql.org. Instructions for building from source can be found in the documentation.

Alpha/Beta/RC Releases and development snapshots (unstable)

There are source code and binary packages of beta and release candidates, and of the current development code available for testing and evaluation of new features. Note that these builds should be used **for testing purposes only**, and not for production systems.

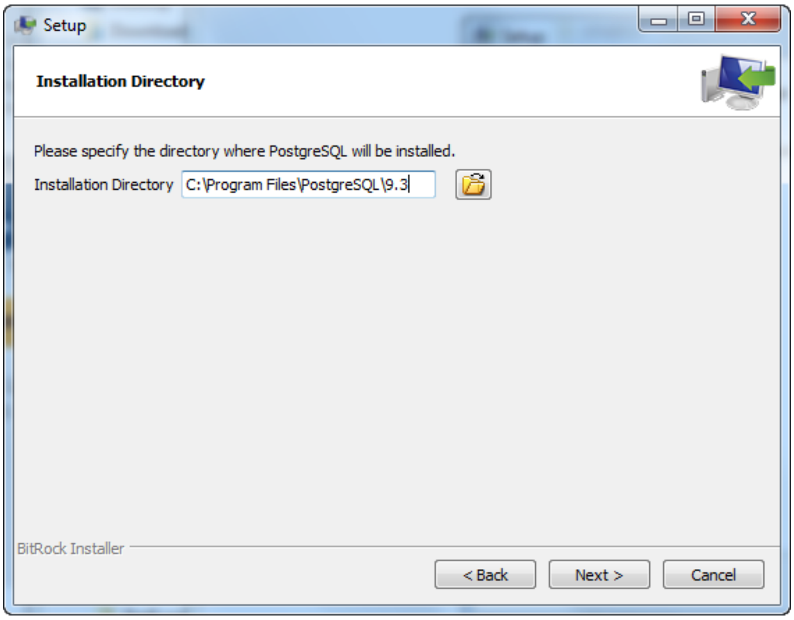
We will next install and configure the software.



Setup - PostgreSQL

Welcome to the PostgreSQL Setup Wizard.

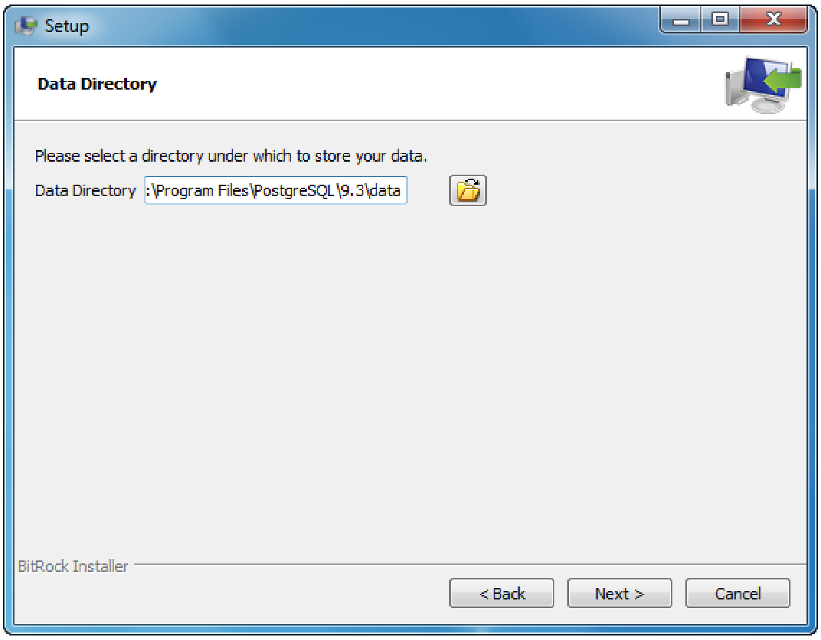
We will then choose a directory for PostgreSQL installation,



Please specify the directory where PostgreSQL will be installed.

Installation Directory C:\Program Files\PostgreSQL\9.3

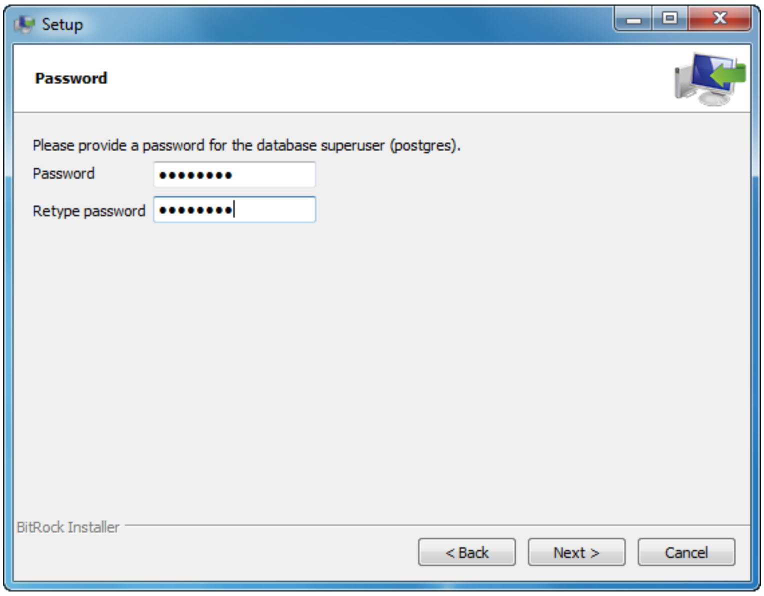
and a directory that will store all the data (including program configurations, database files and configurations, etc.).



Please select a directory under which to store your data.

Data Directory :\Program Files\PostgreSQL\9.3\data

The installer will now ask for a password. This will be the password for *postgres* user (root)



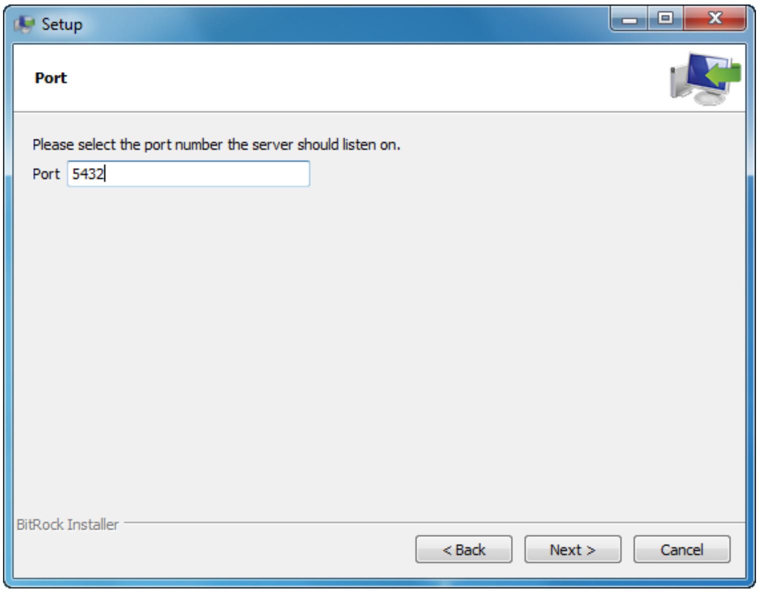
Password

Please provide a password for the database superuser (postgres).

Password ••••••••

Retype password ••••••••

We can also specify the port number in which the server will listen. We can use the default port if it's not being used on the computer we are carrying out the installation by any other service.



Port

Please select the port number the server should listen on.

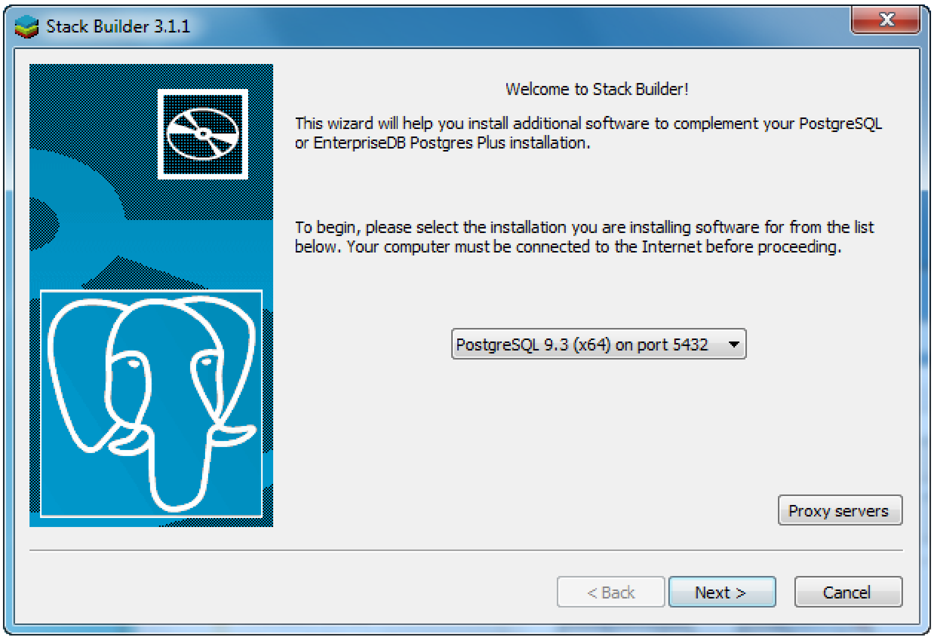
Port 5432

Once the installation is completed, we will be asked to launch a software named "Stack Builder". This software presents a GUI which allows to install plugins, drivers and new features for our PostgreSQL installation without ease.

PostGIS installation

In our case, we will use Stack Builder, but PostGIS can also be manually downloaded and installed from the official website (http://postgis.net/), which has all the necessary files.

When Stack Builder is executed, it will look for local valid PostgreSQL installations to install on it. It's worth noting that Stack Builder can also install on remote servers.



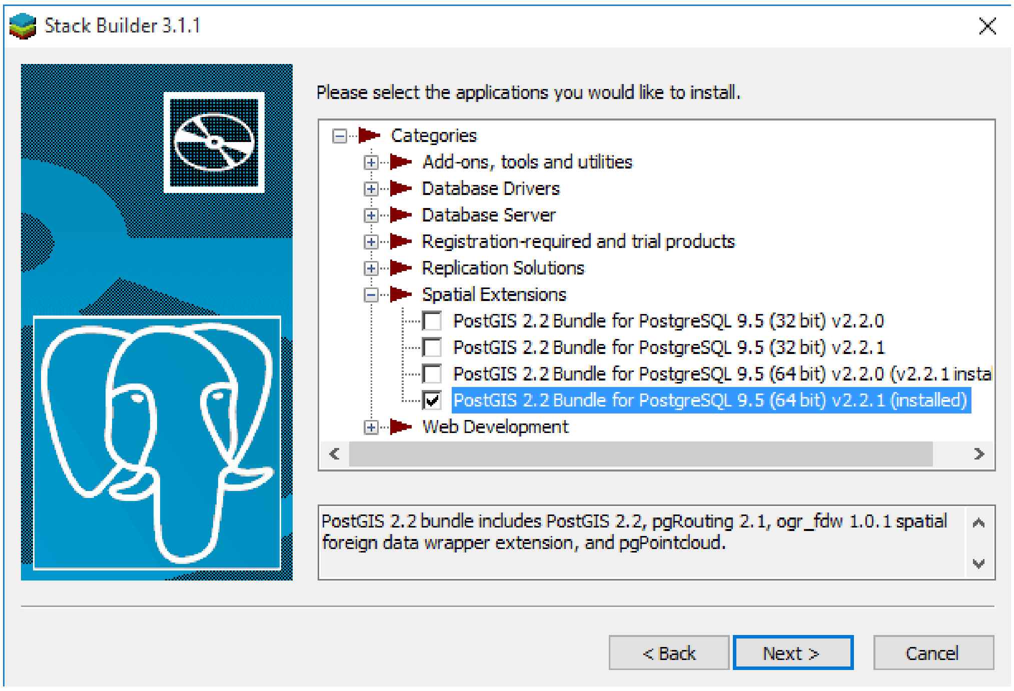
Welcome to Stack Builder!

This wizard will help you install additional software to complement your PostgreSQL or EnterpriseDB Postgres Plus installation.

To begin, please select the installation you are installing software for from the list below. Your computer must be connected to the Internet before proceeding.

PostgreSQL 9.3 (x64) on port 5432

In this screen, extensions can be selected in order to install them. We can find PostGIS inside the "Spatial Extensions" category. We will choose the right version for the server (in our case, PostGIS 2.2.1, 64bit).

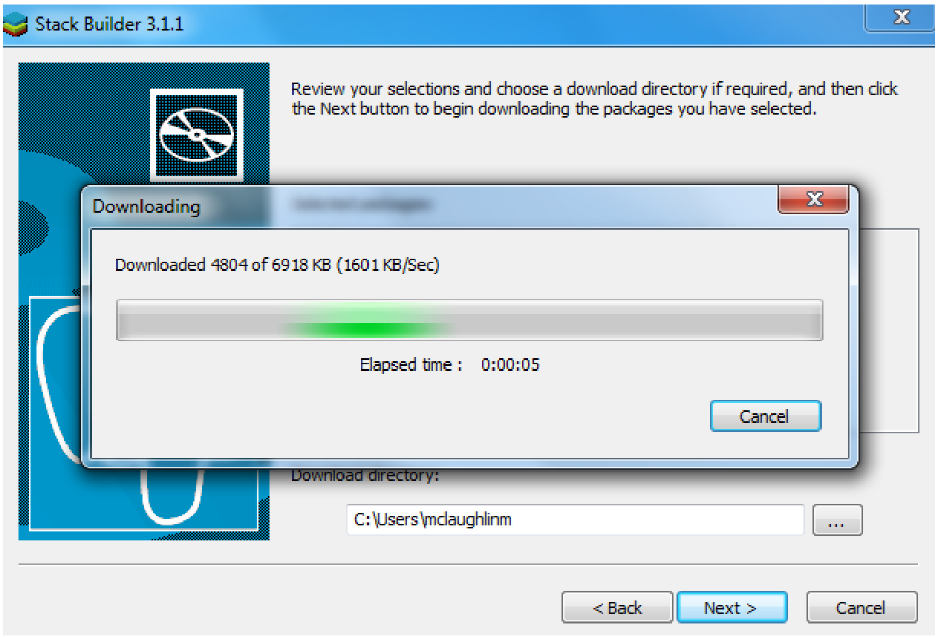


Please select the applications you would like to install.

- Categories
  - Add-ons, tools and utilities
  - Database Drivers
  - Database Server
  - Registration-required and trial products
  - Replication Solutions
  - Spatial Extensions
    - ☐ PostGIS 2.2 Bundle for PostgreSQL 9.5 (32 bit) v2.2.0
    - ☐ PostGIS 2.2 Bundle for PostgreSQL 9.5 (32 bit) v2.2.1
    - ☐ PostGIS 2.2 Bundle for PostgreSQL 9.5 (64 bit) v2.2.0 (v2.2.1 insta
    - ☑ PostGIS 2.2 Bundle for PostgreSQL 9.5 (64 bit) v2.2.1 (installed)
  - Web Development

PostGIS 2.2 bundle includes PostGIS 2.2, pgRouting 2.1, ogr_fdw 1.0.1 spatial foreign data wrapper extension, and pgPointcloud.

Once Next button is pressed, Stack Builder will start to download all checked extensions.



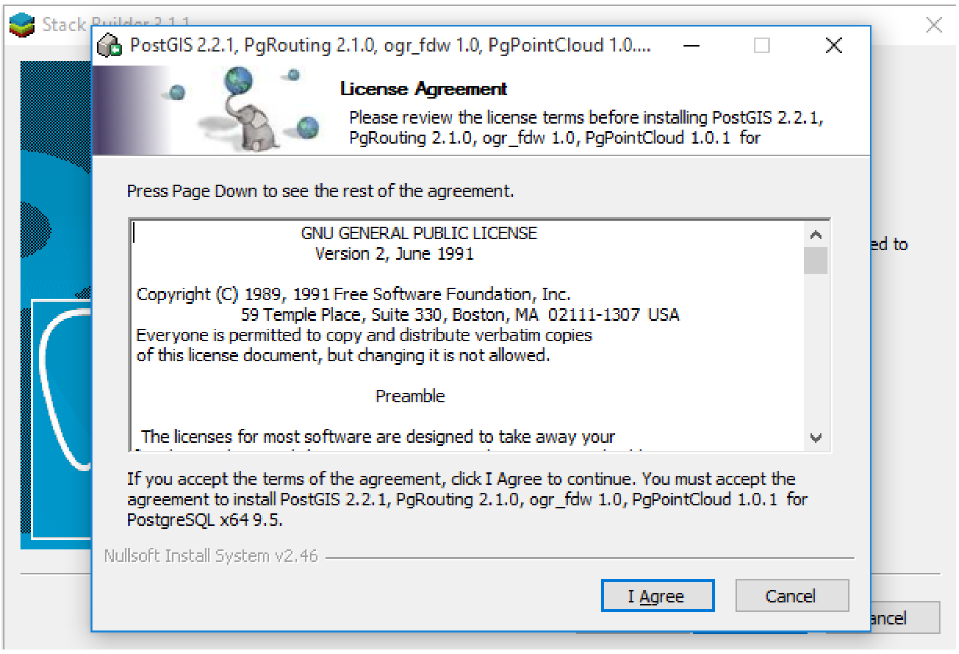
Review your selections and choose a download directory if required, and then click the Next button to begin downloading the packages you have selected.

Downloaded 4804 of 6918 KB (1601 KB/Sec)

Elapsed time : 0:00:05

C:\Users\mclaughlinm

After the completion of the required downloads, Stack Builder will launch the selected installers. From now on, the steps described in this section will be exactly the same as though we downloaded the official installer from PostGIS website.


Stack Builder 3.1.1
PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0....
License Agreement
Nullsoft Install System v2.46
I Agree
Cancel
ed to
ancel

Please review the license terms before installing PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 for

Press Page Down to see the rest of the agreement.

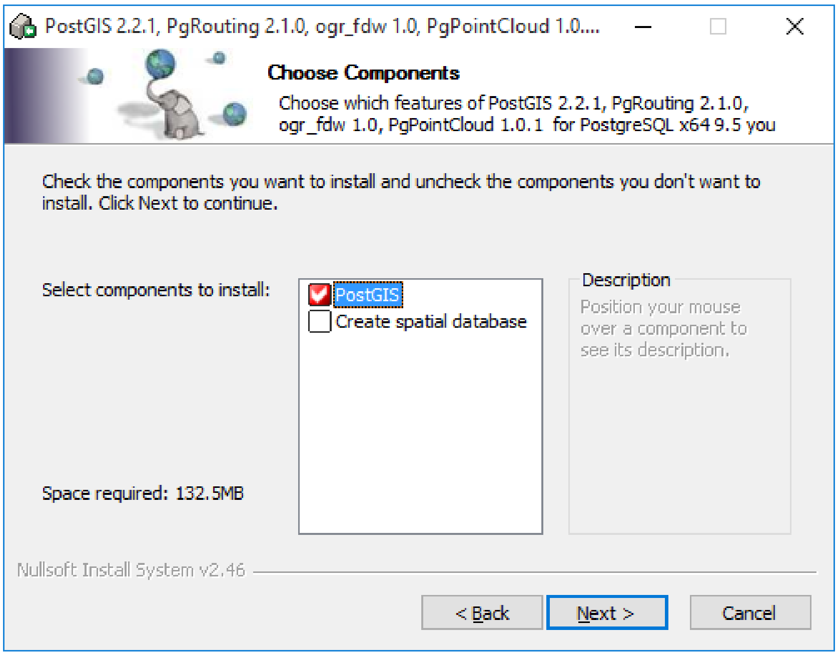
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 for PostgreSQL x64 9.5.



PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0....

Choose Components

Choose which features of PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 for PostgreSQL x64 9.5 you

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ PostGIS

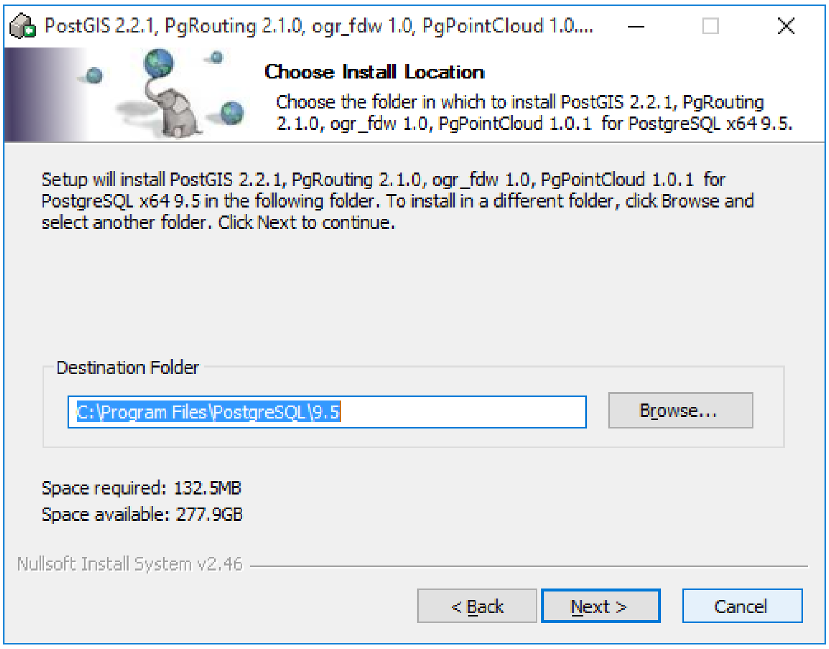
☐ Create spatial database

Description

Position your mouse over a component to see its description.

Space required: 132.5MB

In addition to installing PostGIS, an option for creating a spatial database appears. After that, we will be asked for the PostgreSQL installation directory, and the installation will start right away.



PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0....

Choose Install Location

Choose the folder in which to install PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 for PostgreSQL x64 9.5.

Setup will install PostGIS 2.2.1, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 for PostgreSQL x64 9.5 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

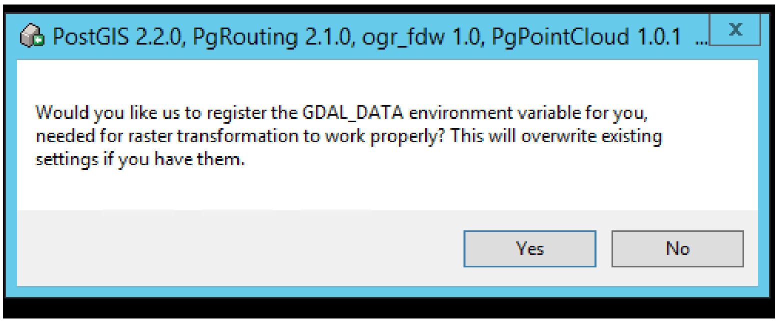
Destination Folder: C:\Program Files\PostgreSQL\9.5

Space required: 132.5MB

Space available: 277.9GB

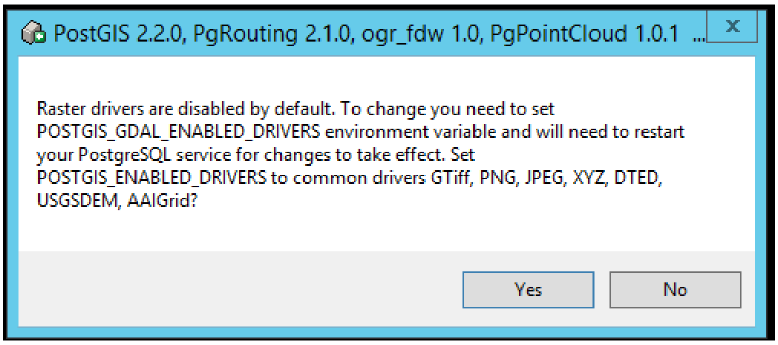
When the installation is over, we will be presented with 3 dialog boxes which enable some configurations of PostGIS plugin.

The first one will ask to register an environment variable (GDAL_DATA) for us. This variable is needed for converting the vector information into images. It is recommended to let it register the variable in the first installation.


PostGIS 2.2.0, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 ...

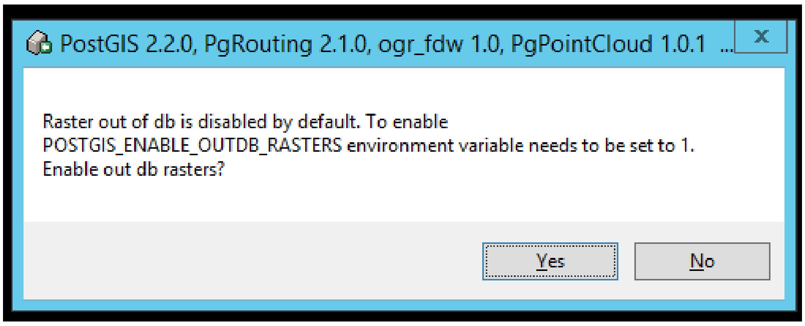
Would you like us to register the GDAL_DATA environment variable for you, needed for raster transformation to work properly? This will overwrite existing settings if you have them.

The next dialog can enable the raster drivers. These drivers are, according to the documentation, safe (meaning that they don't access external web services for rastering the images, and refers to functions such as *st_aspng()*)


PostGIS 2.2.0, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 ...

Raster drivers are disabled by default. To change you need to set POSTGIS_GDAL_ENABLED_DRIVERS environment variable and will need to restart your PostgreSQL service for changes to take effect. Set POSTGIS_ENABLED_DRIVERS to common drivers GTiff, PNG, JPEG, XYZ, DTED, USGSDEM, AAIGrid?

The last dialog will enable the out of database rastering. We will activate it if needed.


PostGIS 2.2.0, PgRouting 2.1.0, ogr_fdw 1.0, PgPointCloud 1.0.1 ...

Raster out of db is disabled by default. To enable POSTGIS_ENABLE_OUTDB_RASTERS environment variable needs to be set to 1. Enable out db rasters?

These functionalities (and some additional drivers not shown in the second dialog) can be enabled at any time if needed.

Creation of PostGIS enabled databases

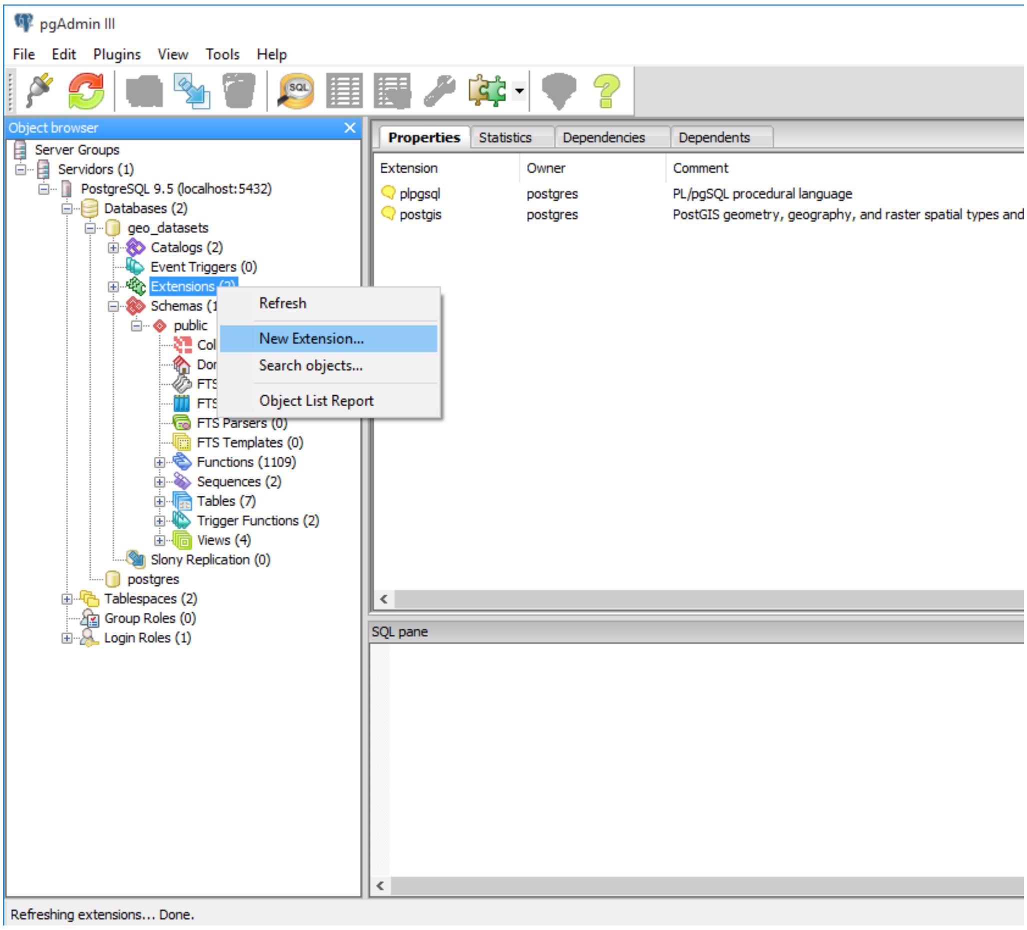
The fastest and easiest way to create a PostGIS enabled database is by using SQL queries. An example is presented as follows:

```
-- Create a new database named “geo_datasets”
CREATE DATABASE geo_datasets;
-- Connect to our newly created database
\connect geo_datasets
-- Enabling PostGIS plugin
CREATE EXTENSION postgis;
```

With this, the database that was just created can already use functions and data type from PostGIS.

But in our case, where we import data from a shapefile, as in the case of a new database creation, we will need to enable the PostGIS plugin (per database) in order to be capable of using “geom” data type, and the functions that process that type of data.

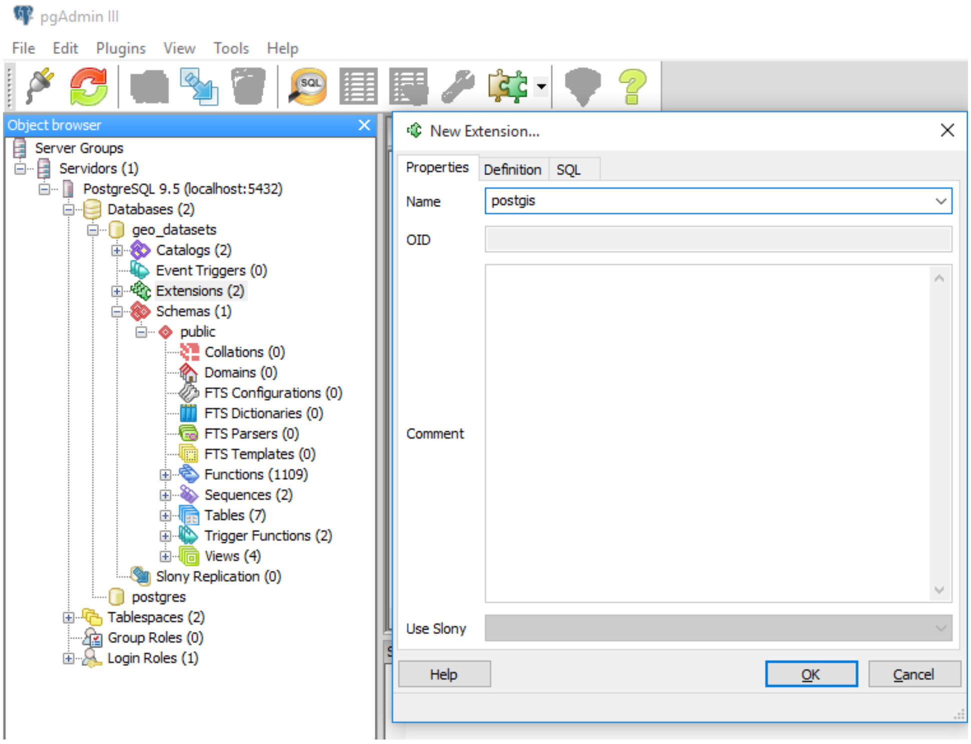
We can also add the plugin using pgAdmin, PostgreSQL’s GUI administration tool, by right clicking on the extensions section of our desired database, which will bring up this contextual menu:



Extension	Owner	Comment
plpgsql	postgres	PL/pgSQL procedural language
postgis	postgres	PostGIS geometry, geography, and raster spatial types and

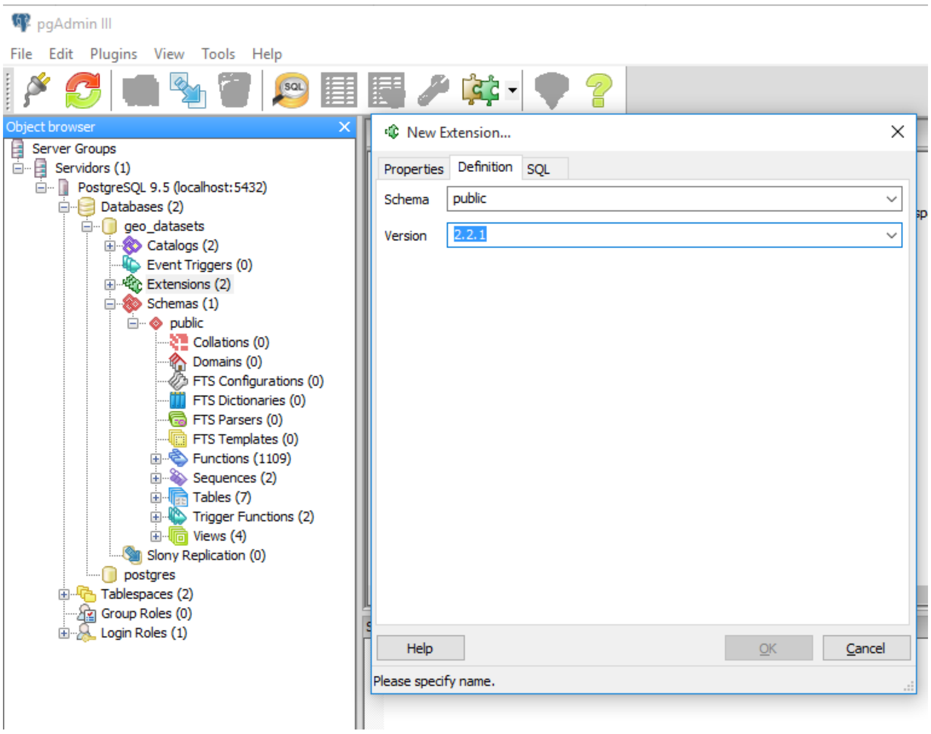
Refreshing extensions... Done.

In "name" field, we will write or select the extension we want to enable from the dropdown (postgis).



Name: postgis

Inside the definition tab, we will configure the extension to be used on our desired schema (the schema where the geolocalization data tables will be stored), and the plugin version to use.



Schema: public

Version: 2.2.1

Please specify name.

By doing this, the database will be now ready to receive the imported data from the shapefiles.

Importing a Shapefile to database

The next step will be to import the datasets to a table. PostgreSQL provides the *shp2pgsql* to do that.

Shp2pgsql

shp2pgsql generates an SQL script from ESRI shape and DBF files suitable for loading into a PostGIS enabled database.

shp2pgsql utility is located in the bin directory of the postgreSQL directory, e.g. c:\PostgreSQL\9.5\bin

```
USAGE: shp2pgsql [<options>] <shapefile> [[<schema>.]<table>]
OPTIONS:
  -s [<from>:]<srid> Set the SRID field. Defaults to 0.
      Optionally reprojects from given SRID (cannot be used with -D).
 (-d|a|c|p) These are mutually exclusive options:
     -d  Drops the table, then recreates it and populates
         it with current shape file data.
     -a  Appends shape file into current table, must be
         exactly the same table schema.
     -c  Creates a new table and populates it, this is the
         default if you do not specify any options.
     -p  Prepare mode, only creates the table.
  -g <geocolumn> Specify the name of the geometry/geography column
      (mostly useful in append mode).
  -D  Use postgresql dump format (defaults to SQL insert statements).
  -e  Execute each statement individually, do not use a transaction.
      Not compatible with -D.
  -G  Use geography type (requires lon/lat data or -s to reproject).
  -k  Keep postgresql identifiers case.
  -i  Use int4 type for all integer dbf fields.
  -I  Create a spatial index on the geocolumn.
  -m <filename>  Specify a file containing a set of mappings of (long) column
     names to 10 character DBF column names. The content of the file is one
or
     more lines of two names separated by white space and no trailing or
     leading space. For example:
         COLUMNNAME DBFFIELD1
         AVERYLONGCOLUMNNAME DBFFIELD2
  -S  Generate simple geometries instead of MULTI geometries.
  -t <dimensionality> Force geometry to be one of '2D', '3DZ', '3DM', or '4D'
  -w  Output WKT instead of WKB.  Note that this can result in
      coordinate drift.
  -W <encoding> Specify the character encoding of Shape's
      attribute column. (default: "UTF-8")
  -N <policy> NULL geometries handling policy (insert*,skip,abort).
  -n  Only import DBF file.
  -T <tablespace> Specify the tablespace for the new table.
      Note that indexes will still use the default tablespace unless the
      -X flag is also used.
  -X <tablespace> Specify the tablespace for the table's indexes.
      This applies to the primary key, and the spatial index if
      the -I flag is used.
  -?  Display this help screen.
```

PSQL Connection options

```
  -h, --host=HOSTNAME      database server host or socket directory
  -p, --port=PORT          database server port number
  -U, --username=NAME      connect as specified database user
  -W, --password           force password prompt (should happen
automatically)
  -e, --exit-on-error      exit on error, default is to continue
```

If no input file name is supplied, then standard input is used.

Example

Load data into PostgreSQL from the ESRI shape file Gas Leaks:

```
$ shp2pgsql -s 4326 -d -I -S -g geometry my_path\gasleak_clarus_demo.shp
public.gasleak > my_path\gasleak.sql
```

Then run the SQL script with the psql utility:

```
$ psql -h my_server -d my_db_name -U my_db_user -f my_path\gasleak.sql
```

Exporting a database to Shapefile

This section explains how to export to a shape file from PostGIS (PostgreSQL with the PostGIS extension). PostgreSQL provides the *pgsql2shp* utility to do that.

pgsql2shp

pgsql2shp dumps a PostGIS database table, view or SQL query to ESRI shape file format.

pgsql2shp utility is located in the bin directory of the PostgreSQL directory, e.g. c:\PostgreSQL\9.5\bin

```
USAGE: pgsql2shp [<options>] <database> [<schema>.]<table>
       pgsql2shp [<options>] <database> <query>

OPTIONS:
  -f <filename>  Use this option to specify the name of the file to create.
  -h <host>  Allows you to specify connection to a database on a
     machine other than the default.
  -p <port>  Allows you to specify a database port other than the default.
  -P <password>  Connect to the database with the specified password.
  -u <user>  Connect to the database as the specified user.
  -g <geometry_column> Specify the geometry column to be exported.
  -b Use a binary cursor.
  -r Raw mode. Do not assume table has been created by the loader. This would
     not unescape attribute names and will not skip the 'gid' attribute.
  -k Keep PostgreSQL identifiers case.
  -m <filename>  Specify a file containing a set of mappings of (long) column
     names to 10 character DBF column names. The content of the file is one
or
     more lines of two names separated by white space and no trailing or
     leading space. For example:
         COLUMNNAME DBFFIELD1
         AVERYLONGCOLUMNNAME DBFFIELD2
  -? Display this help screen.
```

Example

Export data to an ESRI shape file Gas Leaks :

```
$ pgsql2shp -u my_db_user -P my_db_password -h my_server -g geometry -f
my_path\gasleak_clarus_demo.shp my_db_name  public.gasleak
```