# Geographic platform configuration

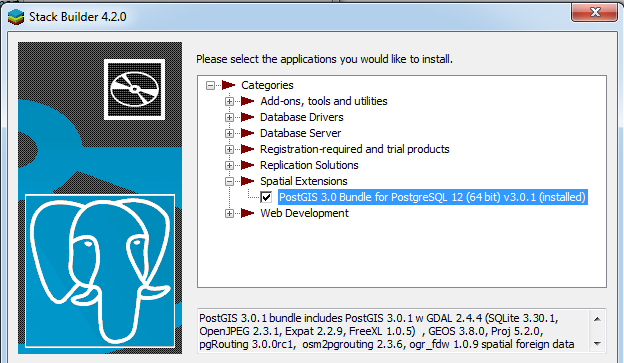
The following notes are meant to drive you through the building of a minimalistic architecture with actual working data as a proof of concept of the expected software and data architecture.

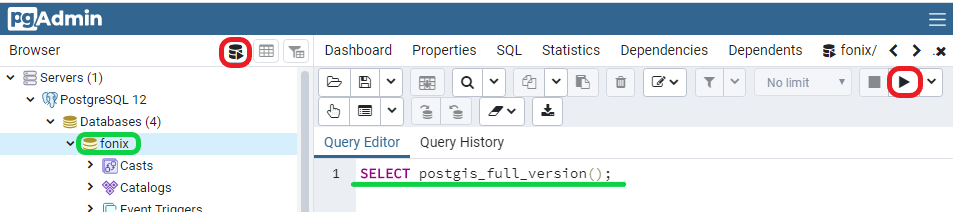
It’s based on Windows, either real or a Virtual Machine – any recent Windows version with decent resources will do.

We will build the system bottom up:

1. Data Layer: PostGIS spatial extensions to PostgreSQL Object-Relational Database Management System (ORDBMS);
2. Application Layer: MapServer mapping engine.
3. Presentation Layer: Leaflet/OpenLayers

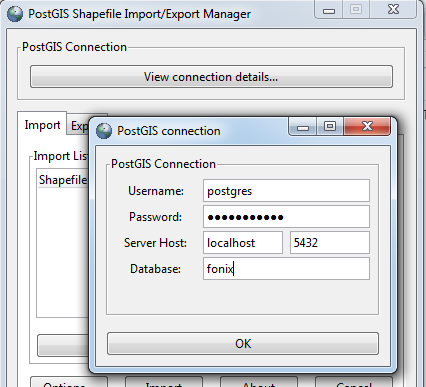
## D.1 – Data Layer: PostgreSQL / PostGIS

1. Install PosgreSQL from the installer at the web site. At the time of writing version 12.2 was current.
2. Install PostGIS spatial extensions. This is a separate install which is provided as an integrated plugin. The safest way to get this is by running the Stack Builder tool. Select PosGIS: download and installation will run automatically.  
   
3. Run the pgAdmin administration tool
4. Create Database “fonix”
5. Load the PostGIS spatial extension:
   * Activate the Tool > Query Tool panel and run command:   
     CREATE EXTENSION postgis;
   * Verify PostGIS version by running command:   
     SELECT postgis\_full\_version();

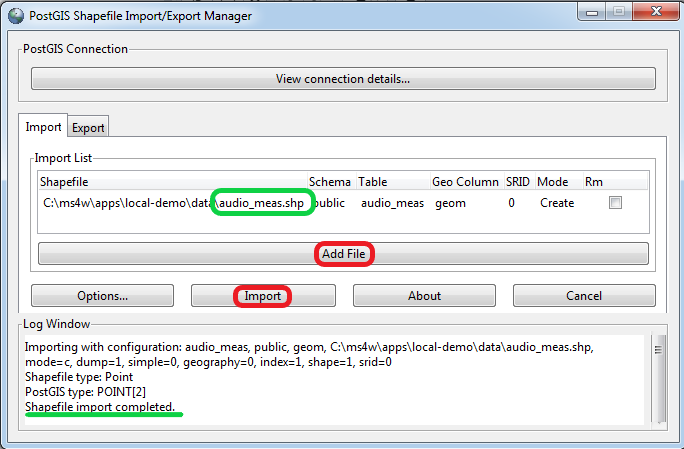


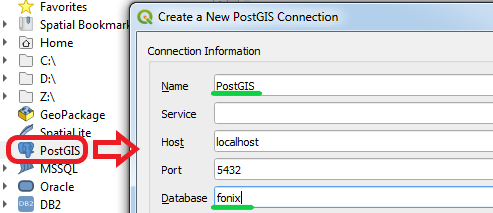
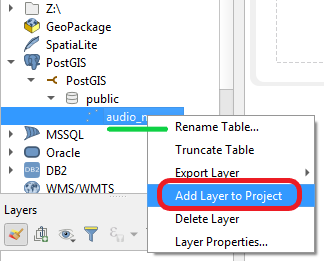
Please note the Play button, or F5 to execute the query.

1. Import sample measurement shapefile:
   * Run the “<drive>:\Program Files\PostgreSQL\<version.number>\bin\postgisgui\shp2pgsql-gui.exe”.  
     The “PostGIS Shapefile Import/Export Manager” will appear.
   * Connect to the newly created database “fonix”



* + Add the shape file and import:



1. Verify the data is in good… shape [pan intended] by opening the dataset in QGIS desktop application:
   * Open QGIS and create a connection to the “fonix” database.  
     
   * Use the “postgres” user – that will be changed later at some point, of course. The imported dataset “audio\_meas” will be listed as a point layer.
   * Right click on the “audio\_meas” and it as a layer to the current project  
     
   * Add some background, i.e. OpenStreetMap, and zoom to layer: this step is not needed but will help you check for geographic consistency: in our sample, a number of points located north of Rome pop up.  
     

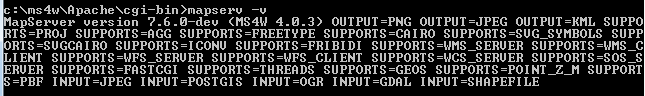
## D.2 – Application Layer: MapServer

An easy way to get a MapServer environment up and running is by getting MapServer for Windows (MS4W) installed on Windows OS.

MS4W is a very flexible bundle that includes:

* Apache Web Server, V2.4.41 at the time of writing;
* MapServer mapping engine, v7.6.0 at the time of writing;
* Lots of tools and libraries to facilitate data interconnection and management;
* Several application frameworks, which tease you to experiment with different client implementations.

The following step will lead to a small sample configuration of MapServer serving a map made of a few layers with different data sources, including the PostGIS configuration from D.1 above.

1. Download and install MS4W from the ms4w.com web site. Version 4.0.3 was current at the time of writing.
2. Install and restart MS4W by using the apache-install and apache-restart batch files from an administrative command line prompt.
3. You may want to run setenv.bat from an administrative command line prompt in order to activate the echo-system of tools, libraries, drivers available.
4. Verify MapServer is up by issuing *c:\ms4w\Apache\cgi-bin\mapserv -v* from a command prompt:  
   
5. Verify MapServer is accessible from the web side by calling the url [*http://localhost/cgi-bin/mapserv.exe*](http://localhost/cgi-bin/mapserv.exe)*?* that will cause an expected “No query information to decode” error.
6. Browsing to url *localhost* will show the landing page, that includes the default map - sample map of the world.
7. Subfolder c:\ms4w\apps\local-demo contains all resources for such map, in particular, in this specific example:
   1. The mapfile *local.map*: this is the map definition file, describing all layers and their configuration.
   2. A sqlite database – a very agile file-based spatial database. This is just an example – you may feed data to MapServer from any source, at least from any OGR/GDAL-compliant data sources.
8. Adding the sample dataset from section D.1 as a new layer in the map is straightforward, provided some caution to zoom layering: a point layer should be drawn on top of the other layers not to be overwritten. As the drawing order in a MapServer mapfile is top-to-bottom, the point layer will be added at the end of the mapfile.
9. Remembering that the sample dataset is hosted by PostGIS, a very basic sample configuration for the new layer in the mapfile would be:

LAYER

NAME "audio\_meas"

CONNECTIONTYPE POSTGIS

CONNECTION "host=localhost port=5432 dbname=fonix user=postgres password=something"

DATA "geom from audio\_meas using unique gid"

TYPE POINT

STATUS ON

CLASS

SYMBOL "square"

SIZE 8

COLOR 255 0 0

END

END # layer

1. Insert the code above in the map file.
2. The default sample client from MapServer, built around MapServer CGI with OpenLayers is one of the endless possibilities to show results. Browse to *localhost*, zoom to Rome [beware of coronavirus] and enjoy the map:

