

Setting Up An OpenStreetmap Server

March 15, 2010 / Gavin Jackson



Introduction

Recently I had a project at work to build a standalone Openstreetmap server (think of this as a locally hosted google maps server). Here are my notes on how I got it up and running (with postgis, mapserver, openlayers, Tilecache and FGS).

These steps should work on most Linux distros (I got this working on Ubuntu Karmic and Centos 5.4). Note that this renders tiles for Australia only (it is of course possible to generate tiles for the complete world).

Ironically, we didn't end up using this as the final mapping solution (we ended up using Openlayers against Yahoo! maps - generating georss xml for the data we wanted to overlay).

Thanks Special thanks goes to Scott Penrose ([home page](#)) for pointing me in the right direction, and Thomas Bonfort ([toms tutorial](#)) for answering my stupid questions.

Links FGS <http://www.maptools.org/fgs/> Tutorial

<http://trac.osgeo.org/mapserver/wiki/RenderingOsmData> OpenLayers <http://www.openlayers.org/>

OpenStreetMap <http://www.openstreetmap.org/> MapServer <http://mapserver.org/> Tilecache

<http://tilecache.org/> FGS (standalone mapserver installation)

<http://www.maptools.org/fgs/>

Install in the default location (/opt/fgs). FGS ships with postgis (I didn't use this version)

You need to initialise the database first: `initdb -D /opt/fgs/apps/postgresql/data -L /opt/fgs/apps/postgresql/share`
POSTGIS ETC

Follow the installation guide here: <http://trac.osgeo.org/mapserver/wiki/RenderingOsmData>

Note that the australia.osm.gz linked from the article does not want to load (primary key violation) I had better luck using the osm file located at <http://www.osmaustralia.org/> Creating your map file I placed the contents of svn checkout <http://mapserver-utils.googlecode.com/svn/trunk/> mapserver-utils-read-only into /osm/ Modify the osmtemplate.map Change the extent line to be your bounding box (I removed the other extents in the file): EXTENT 12569721 -5410502 17711538 -1030958.0625 Also modify the CONFIG "PROJ_LIB" _proj_lib line to point to the location of the epsg file.

```
"/opt/fgs/share/proj/" Uncomment the following line (by removing the ## "+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +no_defs" Change the following line from (they appear twice in the template): PROJECTION "init=epsg:4326"
```

to: PROJECTION "init=epsg:900913" Coastline data should be changed to (this is downloaded during the make process): DATA "/osm/coastlines/processed_p.shp" Update the dbconnection file (to connect to the postgresql database) define _db_connection "host=localhost dbname=osm user=www-data password=www-data port=5432" run the makefile this generates the osm-mapserver.map file Finally perform a find replace (on osm-mapserver.map) of 4326 and replace with 900913 otherwise it will not project the data correctly.

SETTING UP AN OPENLAYERS TEST FILE (This is very similar to the main tutorial). Create a file in /opt/fgs/www/htdocs (I called mine test.html)

With the following content:

```
<br /> body {<br /> width:100%;<br /> height:100%;<br />border: 0px;<br />margin: 0px;<br />padding: 0px;<br />font:sans-serif;<br />font-size:small;<br />}<br />#map {<br />width: 800px;<br />height: 600px;<br />border: 1px solid black;<br />padding: 0px;<br />margin:0px auto;<br />margin-top:30px;<br />position:relative;<br />}<br />
```

```
<br />var lon = 16831422;<br />var lat=-4012608;<br />var zoom = 3;<br />var map, layer;<br />function init(){<br /> map = new OpenLayers.Map( 'map' );<br /> var wms = new OpenLayers.Layer.WMS( "WMS",<br /> "http://localhost/cgi-bin/mapserv?",<br /> {map: '/osm/osm-mapserver.map',<br /> layers: 'default',<br /> format: 'aggpng24',<br /> transparent: 'off'},<br /> {maxExtent: new OpenLayers.Bounds(11548635, -5889094, 18604187, -597430),<br /> scales: [5000,10000,25000,50000,100000,250000,500000,<br /> 1000000,2500000,5000000,10000000,25000000,50000000,100000000],<br /> units: 'm',<br /> projection:new OpenLayers.Projection("EPSG:900913"),<br /> gutter:0,<br /> ratio:1,<br /> wrapDateLine: true,<br /> isBaselayer:true,<br /> singleTile:true,<br /> transitionEffect:'resize'} );<br /> map.addLayers([wms]);<br /> if(!map.getCenter())<br /> map.setCenter(new OpenLayers.LonLat(lon, lat), zoom);<br /> map.addControl(new OpenLayers.Control.Scale('scale'));<br /> map.addControl(new OpenLayers.Control.Permalink('permalink'));<br /> map.addControl(new OpenLayers.Control.MousePosition());<br /> }<br />
```

lien direct

Note the path to the mapserv cgi executable (this ships with fgs). Note the path to the osm.map file (/osm/osm-mapserver.map) Setting Up Tilecache This took a while to figure out, the following post helped me make sense of this process: <http://n2.nabble.com/Configuration-of-MapServer-Tilecache-and-Openlayers-Together-SOLVED-td4438807.html> Modify /opt/fgs/apps/tilecache-2.10/tilecache.cfg

```
# Configuration for MC TileCache
[cache]
type=Disk
base=/opt/fgs/tmp/tilecache/cache

[vmap0]
type=WMSLayer
layers=default
maxresolution=13780.375
bbox=11548635, -5889094, 18604187, -597430
url=http://localhost/cgi-bin/mapserv?MAP=/osm/osm-mapserver.map
mapfile=/osm/osm-mapserver.map
extent_type=loose
srs=EPSG:900913
```

Note that the maxresolution is a magical number it is derived from the bounding box $\max(X) - \min(x)$ /512 This number is used to help calculate the zoom levels. Next step is to create an openlayers html file that uses tilecache (this one is a gav special)

```
<br /> #map {<br /> width: 100%;<br /> height: 99%;<br /> border: 1px solid black;<br /> }<br />
body {<br /> padding:0px;<br /> margin:2px<br /> }<br /> #labs {<br /> position:absolute;<br />
bottom:15px;<br /> left:7px;<br /> font-size:smaller;<br /> z-index: 5000;<br /> }<br />

<br /> <!--<br /> var map, layer;<br /> function init(){<br /> map = new OpenLayers.Map( $('map'),
{projection: new OpenLayers.Projection("EPSG:900913"), units: 'm'});<br /> layer = new
OpenLayers.Layer.WMS( "VMap0",<br /> "tilecache.cgi?", {layers: 'vmap0', format: 'image/png'
},{isBaseLayer: true, projection: 'EPSG:900913',units: 'm', maxExtent: new
OpenLayers.Bounds(11548635, -5889094, 18604187, -597430), maxResolution: 13780.375} );<br />
map.addLayer(layer);<br /> map.addControl(new OpenLayers.Control.Permalink());<br /> if
(!map.getCenter()) map.zoomToMaxExtent();<br /> }<br /> // --><br />
```

From MetaCarta Labs.

Note the projection, resolution and bounding box are all defined. Pre generating tiles using tilecache
Once the tilecache config has been set up, you can pre-generate tiles using the tilecache_seed.py
command: Note that this generates tiles for 30 zoom levels (which may take some
time)./tilecache_seed.py vmap0 1 30 -b 11548635,-5889094,18604187,-597430