

## Super easy OSM-MapServer for Windows

With the Super easy OSM-MapServer you can set up your own WMS server for delivering OSM data data in ten minutes. The package contains a ready configured Apache web server, MapServer 6.0 WMS server, mapfiles for producing high-quality maps from OpenStreetMap data and sample data from Berlin in two different formats. Shapefiles from Geofabrik.de are included for giving a realistic experience of the speed of the service with either shapefile or PostGIS database. Spatialite database is included for demonstrating the flexibility of rendering from the database. Unfortunately the GDAL SQLite driver is a bit slow at the moment.

### Server

The WMS server included in the package is OSGeo Mapserver 6.0. Installation package contains almost unmodified MS4W package (<http://www.maptools.org/ms4w/>) with one exception. The included Apache http server is configured to start in port 8060 instead of the default port 80 which makes it possible to run the server without administrator rights for the computer.

### Mapfiles

Mapfiles are slightly modified from the mapfiles made by Thomas Bonfort as described in <http://trac.osgeo.org/mapserver/wiki/RenderingOsmData>

Mapfiles are edited to use Spatialite database as an input instead of PostGIS. This makes it possible to deliver the sample data as a one single file and users do not need to install PostgreSQL database and learn to administrate it.

Some changes are made to make is hopefully easier to understand how a pretty complicated rendering like this is done with Mapserver. Map is rendered from 38 layers and definitions for each layer are written to their own mapfiles. The major mapfile **osm\_wms.map** is collecting them together. Spatialite connection is stored into *osm\_sqlite\_connection\_i.map* file and reused by all the layers. The spatial extent of the map layers is also stored in one place in *wms\_extent.map* file. This makes it much faster to generate the WMS GetCapabilities document and thus WMS clients can make connection with the server faster.

The same mapfiles are modified slightly more for using the standard Geofabrik shapefiles as input. Because Geofabrik shapefiles does not contain all the attributes used for defining the original styles the rendered map will have a bit different look.

### OSM data

Sample data comes from a Geofabrik Berlin.osm.bz2 file from August 25, 2011 and from the Berlin shapefiles from the same date.

The Spatialite database was formed by importing data first into PostGIS with `osm2pgsql` and converted then to Spatialite format with these three `ogr2ogr` commands

```
ogr2ogr -dsco SPATIALITE=yes -t_srs epsg:3857 berlin.sqlite  
PG:dbname postgres berlin_point -nln osm_point
```

```
ogr2ogr -update berlin.sqlite PG:dbname postgres berlin_line -nln  
osm_line -t_srs epsg:3857
```

```
ogr2ogr -update berlin.sqlite PG:dbname postgres berlin_polygon  
-nln osm_polygon -t_srs epsg:3857
```

### Explanation of ogr2ogr options

- SPATIALITE=yes were used for creating a Spatialite database that is usable for other purposes that rendering with Mapserver.
- t\_srs makes the resulting database to be in the Spherical Mercator projection and to use the official EPSG:3857 code
- -nln parameter is used for renaming the tables which were named as berlin\_\* in PostGIS into osm\_\* in spatialite. Because of that the SQL used for database queries in the mapfiles does not need to be edited.

### Cleaning the database with OpenLite program

Unfortunately ogr2ogr does not create the most valid Spatialite database. Therefore it was first cleaned by using the OpenLite utility by copying the osm\_point, osm\_line, and osm\_polygon tables into a new empty Spatialite database.

### Installation

- Download the installation package from
- Extract the file into root of any Windows drive letter. After extraction there must be for example a directory [c:\ms4w](#) of [d:\ms4w](#)
- Open the Windows command window and go to directory \ms4w\Apache\bin
- Give command `httpd` and let the window stay open. If you do not see any error messages the Super easy WMS is now installed and running



```
C:\WINDOWS\system32\cmd.exe - httpd  
G:\ms4w\Apache\bin>httpd  
_
```

## Testing

Send the following test requests with a web browser. The result should look like in the images below the requests.

### Test 1

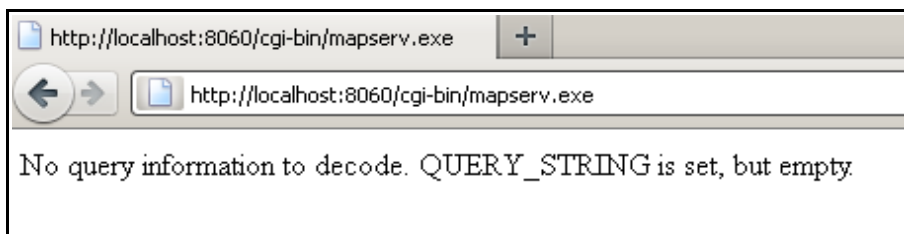
[Http://localhost:8060](http://localhost:8060)



If this page is shown it indicates that the Apache web server is running OK in localhost:8060

### Test 2

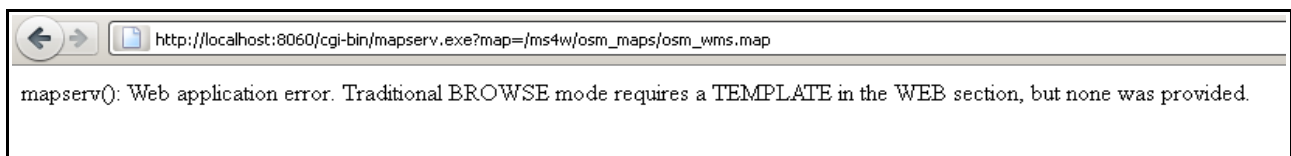
<http://localhost:8060/cgi-bin/mapserv.exe>



This message means that the Mapserver executable is found.

### Test 3

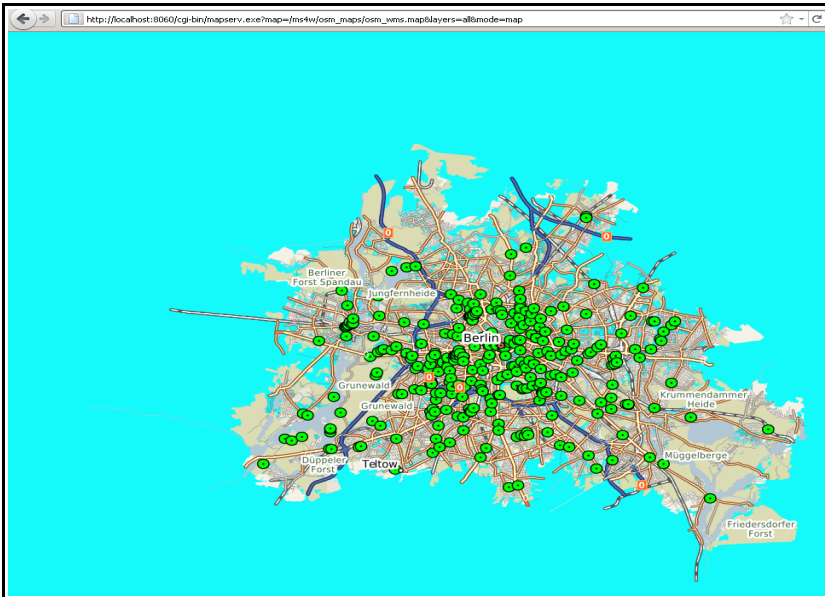
[http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/osm\\_maps/osm\\_wms.map](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/osm_maps/osm_wms.map)



This message means that also mapfile *osm\_wms.map* that will be used for rendering OSM data from Spatialite sample data is found and it does not have syntax errors.

## Test 4

[http://localhost:8060/cgi-bin/mapserv.exe?  
map=/ms4w/osm\\_maps/osm\\_wms.map&layers=all&mode=map](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/osm_maps/osm_wms.map&layers=all&mode=map)



Finally if this map is shown it means that the Spatialite database and the data are valid too. Unfortunately Mapserver is slow with Spatialite and it may take close to one minute to get the image on the browser.

Rendering from the pre-installed Berlin shapefiles can be tested with this URL:

[http://localhost:8060/cgi-bin/mapserv.exe?  
map=/ms4w/Geofabrik\\_shapes\\_maps/osm\\_wms.map&layers=all&mode=map](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik_shapes_maps/osm_wms.map&layers=all&mode=map)

Result will be in the default EPSG:4326 projection.

## **Test with real WMS client**

The WMS-url for the WMS clients is for the Spatialite based service

[http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/osm\\_maps/osm\\_wms.map](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/osm_maps/osm_wms.map)

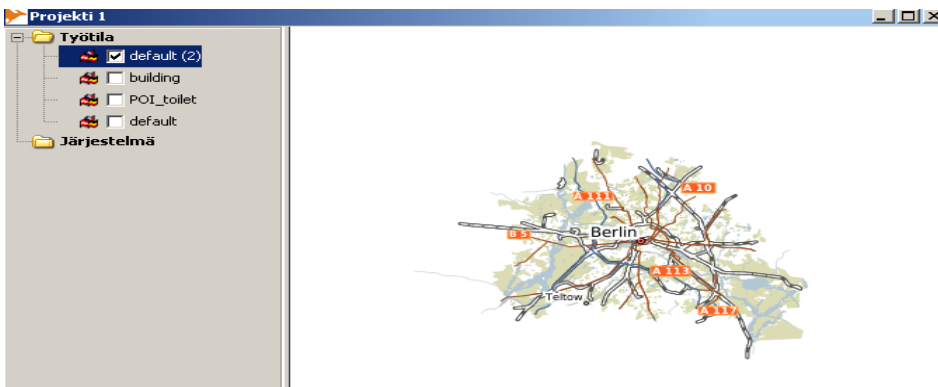
and for the shapefile based service

[http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik\\_shapes\\_maps/osm\\_wms.map](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik_shapes_maps/osm_wms.map)

The following screen captures are from OpenJUMP (<http://openjump.org>) . First image shows the layer listing of the Spatialite based service.



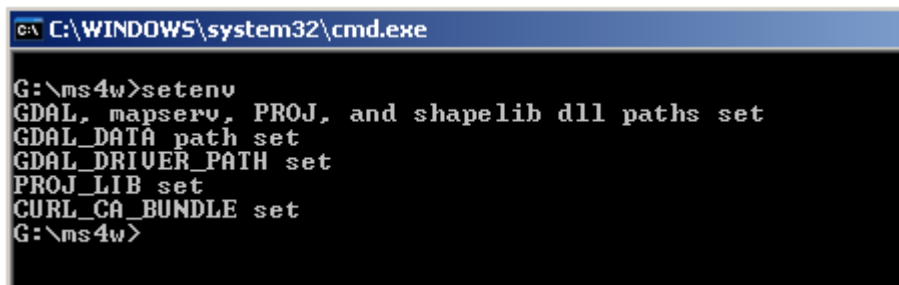
Select the grouping layer "default", right click and use the Zoom into the WMS layer extents option. **Note!** OpenJUMP cannot get enough information from the server for zooming into WMS extents with all the supported projections. If that is the case then it is possible to use the OpenJUMP Zoom to coordinates tool for getting the map window to a correct area.



## Using other data than berlin.sqlite or Berlin shapefiles

The Berlin shapefiles demo is easy to adapt for other Geofabrik.de shapefile datasets .

- Download shapefile archive from <http://download.geofabrik.de/osm/>
- Unzip the shapefiles into the existing shapefile directory `\ms4w\data\Geofabrik_shapes`. Consider making a backup of the Berlin demo shapefiles first.
- Open the Windows command window for getting some gdal tools available. Go to the `\ms4` directory and give command  
`setenv`

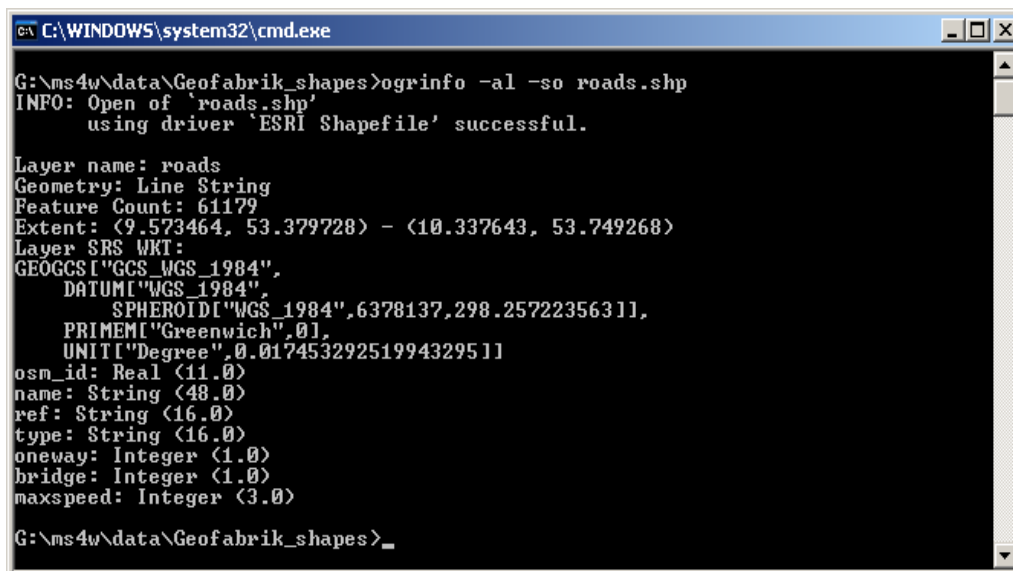


```
C:\WINDOWS\system32\cmd.exe

G:\ms4w>setenv
GDAL, mapserv, PROJ, and shapelib dll paths set
GDAL_DATA path set
GDAL_DRIVER_PATH set
PROJ_LIB set
CURL_CA_BUNDLE set
G:\ms4w>
```

- Next go to the `\ms4w\data\Geofabrik_shapes` directory and read the extents of the roads.shp shapefile with command

```
ogrinfo -al -so roads.shp
```



```
C:\WINDOWS\system32\cmd.exe

G:\ms4w\data\Geofabrik_shapes>ogrinfo -al -so roads.shp
INFO: Open of 'roads.shp'
      using driver 'ESRI Shapefile' successful.

Layer name: roads
Geometry: Line String
Feature Count: 61179
Extent: (9.573464, 53.379728) - (10.337643, 53.749268)
Layer SRS WKT:
GEOGCS["GCS_WGS_1984",
  DATUM["WGS_1984",
    SPHEROID["WGS_1984",6378137,298.25722356311,
    PRIMEM["Greenwich",0],
    UNIT["Degree",0.017453292519943295]]]
osm_id: Real (11.0)
name: String (48.0)
ref: String (16.0)
type: String (16.0)
oneway: Integer (1.0)
bridge: Integer (1.0)
maxspeed: Integer (3.0)

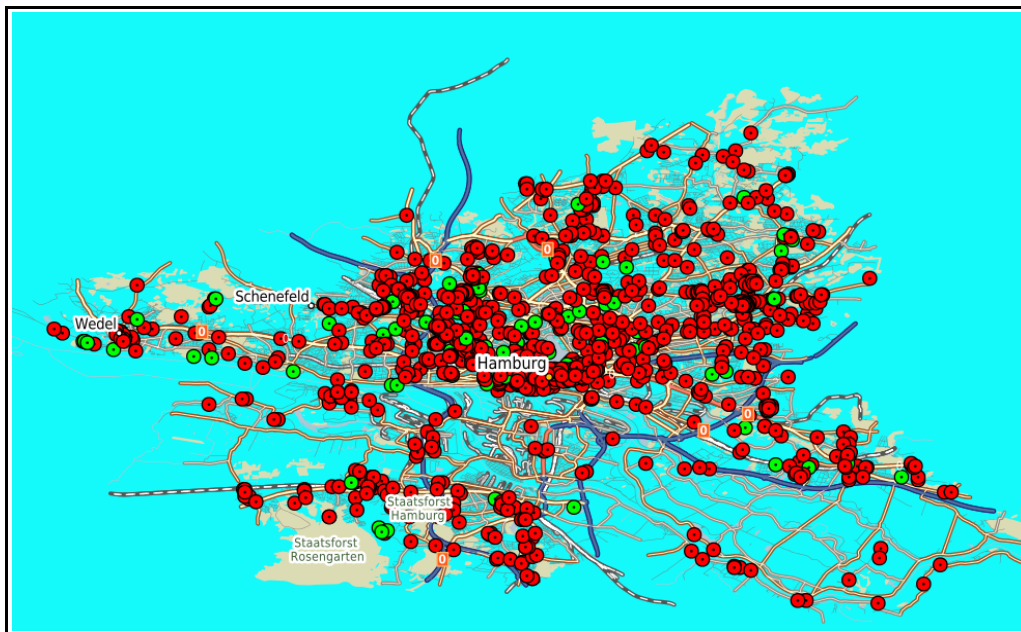
G:\ms4w\data\Geofabrik_shapes>_
```

- Info that is needed is on the line  
Extent: (9.573464, 53.379728) - (10.337643, 53.749268)  
Remove commas and select the numbers  
9.573464 53.379728 10.337643 53.749268

- Open the main Geofabrik mapfile `\ms4w\Geofabrik_shapes_maps\osm_wms.map` and edit this place  
`EXTENT 13.062825 52.327947 13.763997 52.679994`  
`#These are the extents of Berlin shapefiles`  
so that the result will be  
`EXTENT 9.573464 53.379728 10.337643 53.749268`  
`#These are the extents of Hamburg shapefiles`

Now the test URL will show the map covering the whole area of the new Geofabrik Hamburg shapefiles.

[http://localhost:8060/cgi-bin/mapserv.exe?  
map=/ms4w/Geofabrik\\_shapes\\_maps/osm\\_wms.map&layers=all&mode=map](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik_shapes_maps/osm_wms.map&layers=all&mode=map)



### Optimising shapefiles for rendering

Creating spatial indexes for the shapefiles with ogr tools will give more speed for rendering small areas. When index exists Mapserver does not need to read always the whole shapefile but it can use index for getting the subset that it needs fast. Super easy WMS delivery contains the shptree tool for building shapefile index. Like in the case of using ogrinfo tool, go first into `\ms4w\` directory and set the environment with command `setenv`. Next go to the Geofabrik shapefile directory and run the shptree tool for all the shapefiles in the directory. The command to use is

```
shptree roads.shp
```

```
C:\WINDOWS\system32\cmd.exe
G:\ms4w\data\Geofabrik_shapes>shptree roads.shp
creating index of new LSB format
G:\ms4w\data\Geofabrik_shapes>_
```

## A few words about mapfiles and Spatialite

The mapfiles written by Thomas Bonfort have rather complicated rendering rules and scale dependent switching on/off for the layers. They have been written for PostGIS but they do work with Spatialite as well. However, the ogr SQLite driver that Mapserver is ineffective and and therefore Mapserver is pretty slow with Spatialite. PostGIS users should read the previously mentioned guide <http://trac.osgeo.org/mapserver/wiki/RenderingOsmData>

Spatialite can offer an extremely easy way to deliver a whole spatial database in a one file. There is no need to run a separate database engine or know anything about creating database users and so on. For a smallish datasets up to a few gigabytes in size The Spatialite format could be a good alternative for shapefiles and even osm.xml files. Spatialite can hold all the information of the so called Mapnik schema. Even tags from the hstore column can be preserved, albeit only as a text string which contains all the tag-value pairs.

## OpenLayers and tiles

The document <http://trac.osgeo.org/mapserver/wiki/RenderingOsmData> contains sample OpenLayers code for adding a WMS layer. For this server the `singleTile:true` setting is recommended. If the speed of the WMS is not enough any tile cache software can be used in front of Mapserver.

## Need for speed?

Mapserver is very fast with shapefiles. If database is needed for flexibility and good speed is needed, use PostGIS. In addition, running Mapserver as fast-cgi instead of cgi is much faster

## Setting Mapserver to run in port 80 and as a Windows service

Port 8060 is configured because Windows users without admin rights may not be allowed to start a server in the default http port :80. No-admin users are usually not allowed to install Windows services either. Therefore it is advised to stars Apache server as an executable program from the command window. However, it is simple to change installation. The port can be changed by opening the file `\ms4w\Apache\conf\http.conf` and editing the following line  
`Listen 8060` into `Listen 80`.

If also the ip-address is added the server can be accessed from the internet.

Starting the Apache as a Windows service is done by running the Windows batch file `\ms4w\apache-install.bat` which comes always with MS4W.



## Debugging

Mapserver is writing error messages into a file that is set with the MS\_ERRORFILE variable. In the Super easy WMS it is set in the main mapfile *osm\_wms.map* in this place::

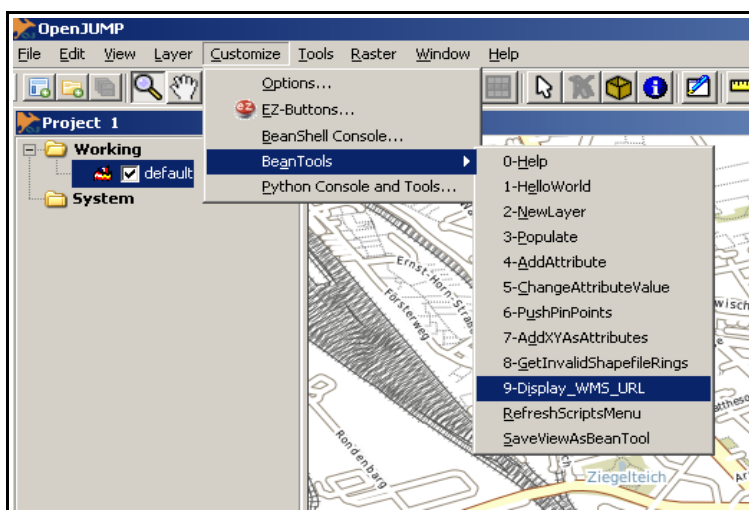
```
# Uncomment the DEBUG line for getting more debug into than just
errors to MS_ERRORFILE
# DEBUG 10
CONFIG "MS_ERRORFILE" "/ms4w/tmp/error.log"
```

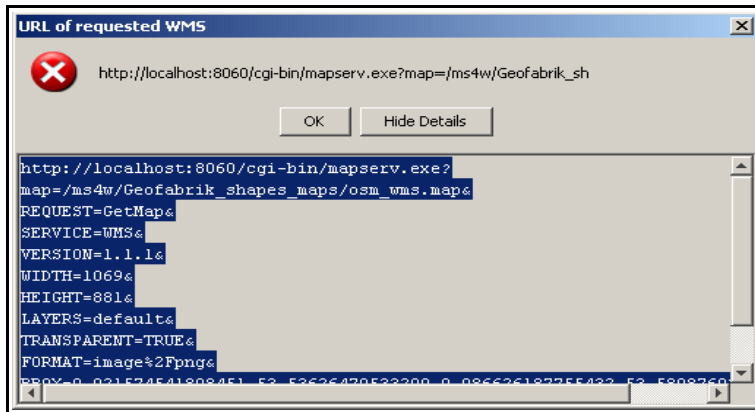
By removing the comment sign # from the beginning of the DEBUG 10 line detailed debugging information about all the rendered layers will be written to the error.log file. Example of the logged data is as follows. **Note!** The values are real values recorded when zooming in with the indexed Hamburg shapefiles.

```
[Tue Sep 13 12:27:28 2011].393000 CGI Request 1 on process 1620
[Tue Sep 13 12:27:28 2011].403000 msDrawMap(): rendering using outputformat named png (AGG/PNG).
[Tue Sep 13 12:27:28 2011].403000 msDrawMap(): WMS/WFS set-up and query, 0.000s
[Tue Sep 13 12:27:28 2011].413000 msDrawMap(): Layer 0 (landuse_layer4), 0.010s
[Tue Sep 13 12:27:28 2011].413000 msDrawMap(): Layer 1 (landuse_layer5), 0.000s
[Tue Sep 13 12:27:28 2011].423000 msDrawMap(): Layer 2 (landuse_layer6), 0.010s
[Tue Sep 13 12:27:28 2011].433000 msDrawMap(): Layer 6 (railways_03), 0.010s
[Tue Sep 13 12:27:28 2011].654000 msDrawMap(): Layer 14 (roadsclose_03), 0.221s
[Tue Sep 13 12:27:28 2011].664000 msDrawMap(): Layer 23 (places_01), 0.010s
[Tue Sep 13 12:27:28 2011].934000 msDrawMap(): Drawing Label Cache, 0.270s
[Tue Sep 13 12:27:28 2011].934000 msDrawMap() total time: 0.531s
[Tue Sep 13 12:27:29 2011].615000 msSaveImage(stdout) total time: 0.681s
[Tue Sep 13 12:27:29 2011].615000 mapserv request processing time (msLoadMap not incl.): 1.222s
[Tue Sep 13 12:27:29 2011].615000 msFreeMap(): freeing map at 01F8B690.
```

## Advanced output options Part 1. Geotiff and PDF output

For familiarizing oneself with the advanced output options of Mapserver it is recommended to install the recent OpenJUMP 1.4.2 version which has a tool for showing the WMS GetMap URL that was used for getting the map that is shown on the screen from a server. This tool can be used for copying the GetMap URL into text editor. The edited URL can finally be send with a web browser.





For example, when the format is changed into image/tiff the server will send a georeferenced GeoTIFF image which can be saved on a disk and used afterwards in any GIS program.

```
http://localhost:8060/cgi-bin/mapserv.exe?
map=/ms4w/Geofabrik_shapes_maps/osm_wms.map&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=500&HEIGHT=400&L
AYERS=default&TRANSPARENT=TRUE&FORMAT=image
%2Ftiff&BBOX=9.921574541898451,53.53626470533209,9.986626187755432,53.589876024321796&SRS=EPSG:4326&STYLES=
```

In a similar way format=application/x-pdf will send the map as a PDF file with vector graphics. Because of this images tolerates zooming and both the lines and texts remain sharp.

Excerpt from Mapserver PDF output with zoom level 100%



Excerpt from Mapserver PDF output with zoom level 400%

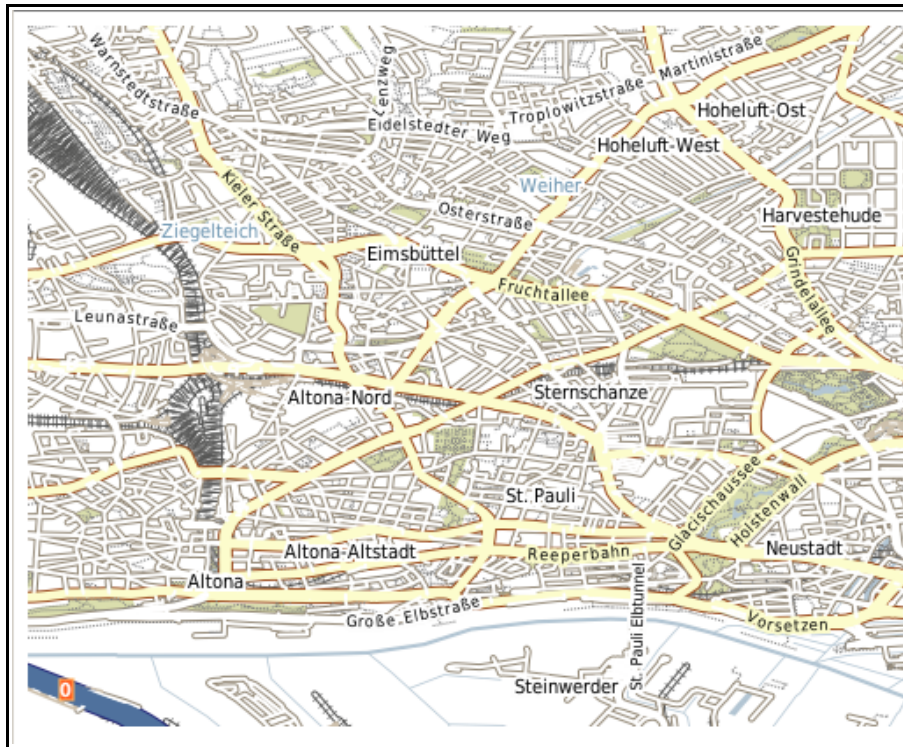


### Advanced output options Part 1. Map resolution control

The following three maps are demonstrating the Mapserver map resolution control. A map that looks good on a computer screen has too coarse resolution for printing. Better resolution for printing can be obtained by taking the same geographical area from the WMS server but with more pixels. While this works fine with aerial images it usually does not give a desired result with maps rendered from vectors. Increasing the pixel count for a certain area makes the server to believe that user is zooming in and wants to see more details. The printed map will be blurry with tiny little texts.

This is a typical WMS GetMap request for showing a map on a computer screen.

[http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik\\_shapes\\_maps/osm\\_wms.map&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=500&HEIGHT=400&LAYERS=default&TRANSPARENT=TRUE&FORMAT=image%2Fpng&BBOX=9.921574541898451,53.53626470533209,9.986626187755432,53.589876024321796&SRS=EPSG:4326&STYLES=](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik_shapes_maps/osm_wms.map&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=500&HEIGHT=400&LAYERS=default&TRANSPARENT=TRUE&FORMAT=image%2Fpng&BBOX=9.921574541898451,53.53626470533209,9.986626187755432,53.589876024321796&SRS=EPSG:4326&STYLES=)



This is the same request but pixel count of the resulting image file is doubled in both width and height directions. Notice the typical effect on texts

[http://localhost:8060/cgi-bin/mapserv.exe?  
map=/ms4w/Geofabrik\\_shapes\\_maps/osm\\_wms.map&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=1000&HEIGHT=800&  
LAYERS=default&TRANSPARENT=TRUE&FORMAT=image  
%2Fpng&BBOX=9.921574541898451,53.53626470533209,9.986626187755432,53.589876024321796&SRS=EPSG:4326&STYLES=](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik_shapes_maps/osm_wms.map&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=1000&HEIGHT=800&LAYERS=default&TRANSPARENT=TRUE&FORMAT=image%2Fpng&BBOX=9.921574541898451,53.53626470533209,9.986626187755432,53.589876024321796&SRS=EPSG:4326&STYLES=)



The following GetMap URL is utilising the Mapserver map\_resolution control. The map appearance is the same than on the computer screen, it is just drawn with more pixels.

[http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik\\_shapes\\_maps/osm\\_wms.map&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=1000&HEIGHT=800&LAYERS=default&TRANSPARENT=TRUE&FORMAT=image%2Fpng&BBOX=9.921574541898451,53.53626470533209,9.986626187755432,53.589876024321796&SRS=EPSG:4326&STYLES=&map\\_resolution=150](http://localhost:8060/cgi-bin/mapserv.exe?map=/ms4w/Geofabrik_shapes_maps/osm_wms.map&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=1000&HEIGHT=800&LAYERS=default&TRANSPARENT=TRUE&FORMAT=image%2Fpng&BBOX=9.921574541898451,53.53626470533209,9.986626187755432,53.589876024321796&SRS=EPSG:4326&STYLES=&map_resolution=150)

