GeoMOOSE Exercise 9 – Understanding MapServer Mapfiles

Goals:

- Gain a general understanding of MapServer Configuration Files (.map)
- Learn how to debug syntax errors in a map file
- Learn how to create a mapfile by adding a point, line and polygon data sources.

Concepts - Understanding the MapServer Mapfile.

- Learning more about MapServer http://www.mapserver.org/documentation.html
- Text based, hierarchical
- Configuration files used to generate
  - Maps, legends, scalebars, reference maps
  - Defines cartography, labeling and queries
  - GeoMOOSE uses them to render mapserver layer types
- MapFile Reference Guide
  - http://mapserver.org/mapfile/index.html#mapfile
- MapServer Symbology
  - http://mapserver.org/mapfile/symbology/construction.html
- GeoMOOSE stores these files in the "maps" folder
- Case insensitive (with few exceptions)
- Paths for files are given relative to the map file location
- Strings must start with a letter and can contain a-z, A-Z, 0-9, - and _. All other strings must be quoted.
- Order of layers is significant – First In First Out (FIFO). So put your point layers at the bottom of a map file.
- GeoMoose uses MapServer layers as a WMS server. See more about MapServer WMS server at http://mapserver.org/ogc/wms_server.html.

Exercise Steps

1) Download three shapefiles in the Week 9 exercises folder from the Moodle class website. They are shapfiles inside the "metrogis" folder. They are major shopping centers (points - shopping_centers.shp), growth management (polygons - gmpa_a.shp) and regional trails (lines - trails_regional.shp). All of the data layers were downloaded from http://www.datafinder.org/catalog/index.asp

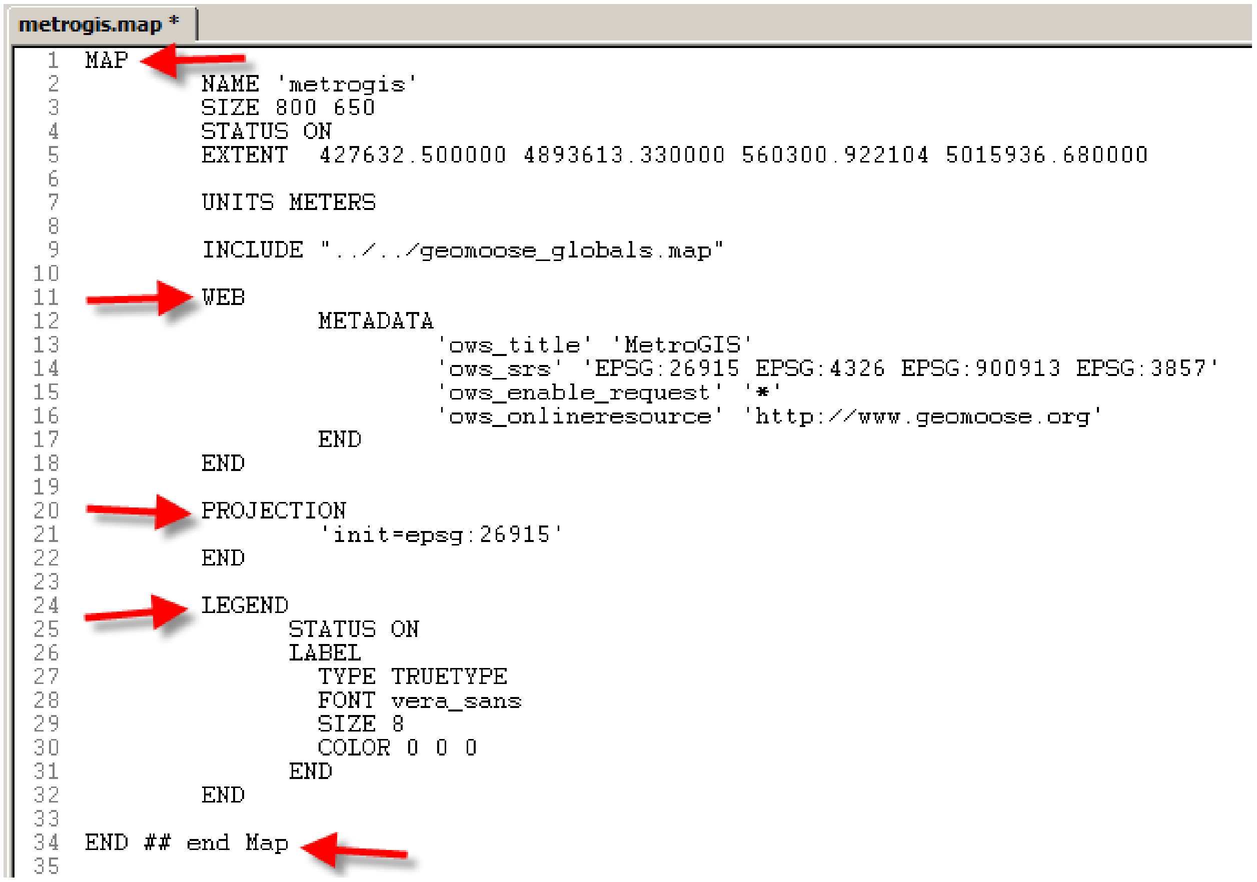
   - Major Shopping Centers is a point layer that we will label.
   - Growth Management Areas is a polygon layer we will split into classes.
   - Regional Trails is a line layer we will label.

2) Copy the 3 shapefiles into C:\ms4w\apps\gm_class\maps\demo\metrogis_shapefiles. *Note: you need these files to make up a valid shapefile (.shp, .shx, .dbf). You also will need to create a metrogis_shapefiles folder.*

3) Next create a new file called metrogis.map in textpad. This is a MapServer mapfile. Save this file into C:\ms4w\apps\gm_class\maps\demo\metrogis_shapefiles' It will be empty for now. For geomoose applications we store the mapfiles in this folder which are used to configure the cartography for mapserver type layers in GeoMOOSE. This is because the MapServer program is actually generating the images being displayed in the web browser from the shapefiles. You can learn more about mapfiles at http://www.mapserver.org/mapfile/index.html.

4) Create the start of the mapfile as shown below in the figure. Notice there is four main parameters called MAP, WEB, PROJECTION and LEGEND. Each has an ending END parameter. The MAP parameter is required in a mapfile. Also noticed there was an include file on line #9 called geomoose_globals.map. Please open this file to see what is in there. MapServer allows for files to be included in others.

   - Learn more about the MAP parameter by reading http://www.mapserver.org/mapfile/map.html.
   - Learn about the WEB object here: http://mapserver.org/mapfile/web.html
   - Learn about the Projection object here: http://mapserver.org/mapfile/projection.html
   - Learn about the Legend object here: http://mapserver.org/mapfile/legend.html
   - Finally notice the "METADATA" object within the WEB object. This is where you config MapServer to turn on the layers in this mapfile as a WMS server. Read more about those configuration parameters here: http://mapserver.org/ogc/wms_server.html


metrogis.map *

```
MAP
        NAME 'metrogis'
        SIZE 800 650
        STATUS ON
        EXTENT  427632.500000 4893613.330000 560300.922104 5015936.680000

        UNITS METERS

        INCLUDE "../../geomoose_globals.map"

        WEB
                METADATA
                        'ows_title' 'MetroGIS'
                        'ows_srs' 'EPSG:26915 EPSG:4326 EPSG:900913 EPSG:3857'
                        'ows_enable_request' '*'
                        'ows_onlineresource' 'http://www.geomoose.org'
                END
        END

        PROJECTION
                'init=epsg:26915'
        END

        LEGEND
              STATUS ON
              LABEL
                TYPE TRUETYPE
                FONT vera_sans
                SIZE 8
                COLOR 0 0 0
              END
        END

END ## end Map
```

5) Next add a LAYER definition for the regional trails. Notice this is a line shapefile that is going to be labeled. You can learn more about the LAYER parameter at http://www.mapserver.org/mapfile/layer.html.

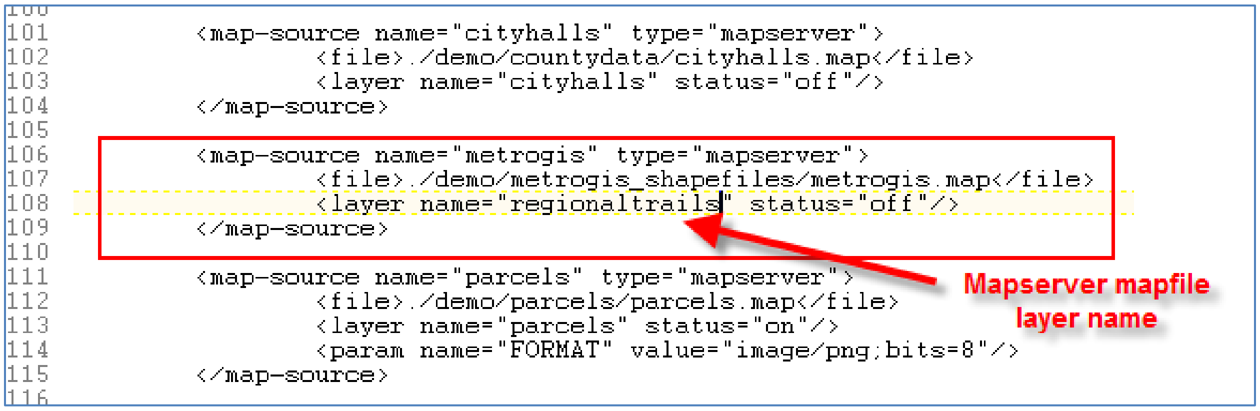
```
LAYER

        NAME 'regionaltrails'

        DATA 'trails_regional' #shapefile Name

        TYPE LINE

        STATUS ON

        CLASSITEM 'CATEGORY_S'  #shapefile attribute

        LABELITEM 'NAME' #shapefile attribute

        LABELCACHE ON

        LABELMAXSCALE 48000

        CLASS

         NAME 'Regional and State Existing'

         EXPRESSION /Regional Existing|State Existing/   #Example of Regular Expression

         COLOR 255 69 5
```

```
 SYMBOL "dashed1"
 SIZE 4
 LABEL
          TYPE TRUETYPE
          FONT vera_sans
          SIZE 8
          ANTIALIAS TRUE
          COLOR 0 0 0
          OUTLINECOLOR 100 100 100
          BUFFER 4
          OFFSET 4 4
          PARTIALS FALSE
          ANGLE FOLLOW
 END
END
CLASS
 NAME 'Regional Proposed'
 EXPRESSION 'Regional Proposed'
 COLOR 100 100 150
 SYMBOL "dashed2"
 SIZE 4
 LABEL
          TYPE TRUETYPE
          FONT vera_sans
          SIZE 8
          ANTIALIAS TRUE
          COLOR 0 0 0
          OUTLINECOLOR 100 100 100
          BUFFER 4
          OFFSET 4 4
          PARTIALS FALSE
          ANGLE FOLLOW
 END
```

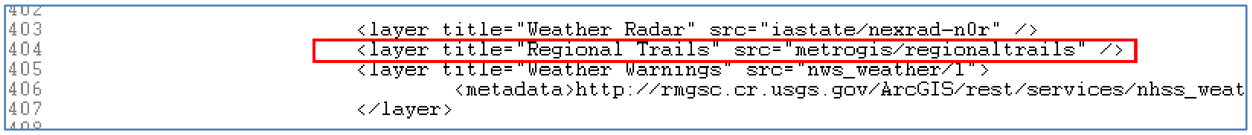
```
          END
          CLASS
           NAME 'Regional Planned'
           EXPRESSION 'Regional Planned'
           COLOR  65 105 225
           SYMBOL "dcircle"
           SIZE 4
           LABEL
                    TYPE TRUETYPE
                    FONT vera_sans
                    SIZE 8
                    ANTIALIAS TRUE
                    COLOR 0 0 0
                    OUTLINECOLOR 100 100 100
                    BUFFER 4
                    OFFSET 4 4
                    PARTIALS FALSE
                    ANGLE FOLLOW
           END
          END
END #Layer
```

6) Notice that the classitem (on line 82 in the graphic above) is an attribute that is used to define 3 different classes of trails. Each class is uniquely defined with an expression. Also notice each class has a corresponding END parameter. Each class also has a LABEL parameter defined that tells MapServer to label that class using the LABELITEM. You can learn more about CLASS parameter at http://www.mapserver.org/mapfile/class.html. In Mapserver there are also 3 different ways to define class expressions. See http://mapserver.org/mapfile/expressions.html#expressions for more information.

7) Next add the definition for the new map layer to the map book. You need to add a <map-source> that tells GeoMOOSE the layer is a mapserver type and references the new metrogis.map file. Notice we called the map source name metrogis. You cannot have duplicate map-source names in the map book. Also the layer name has to match the layer name in the mapserver mapfile. These are also ***case sensitive***!

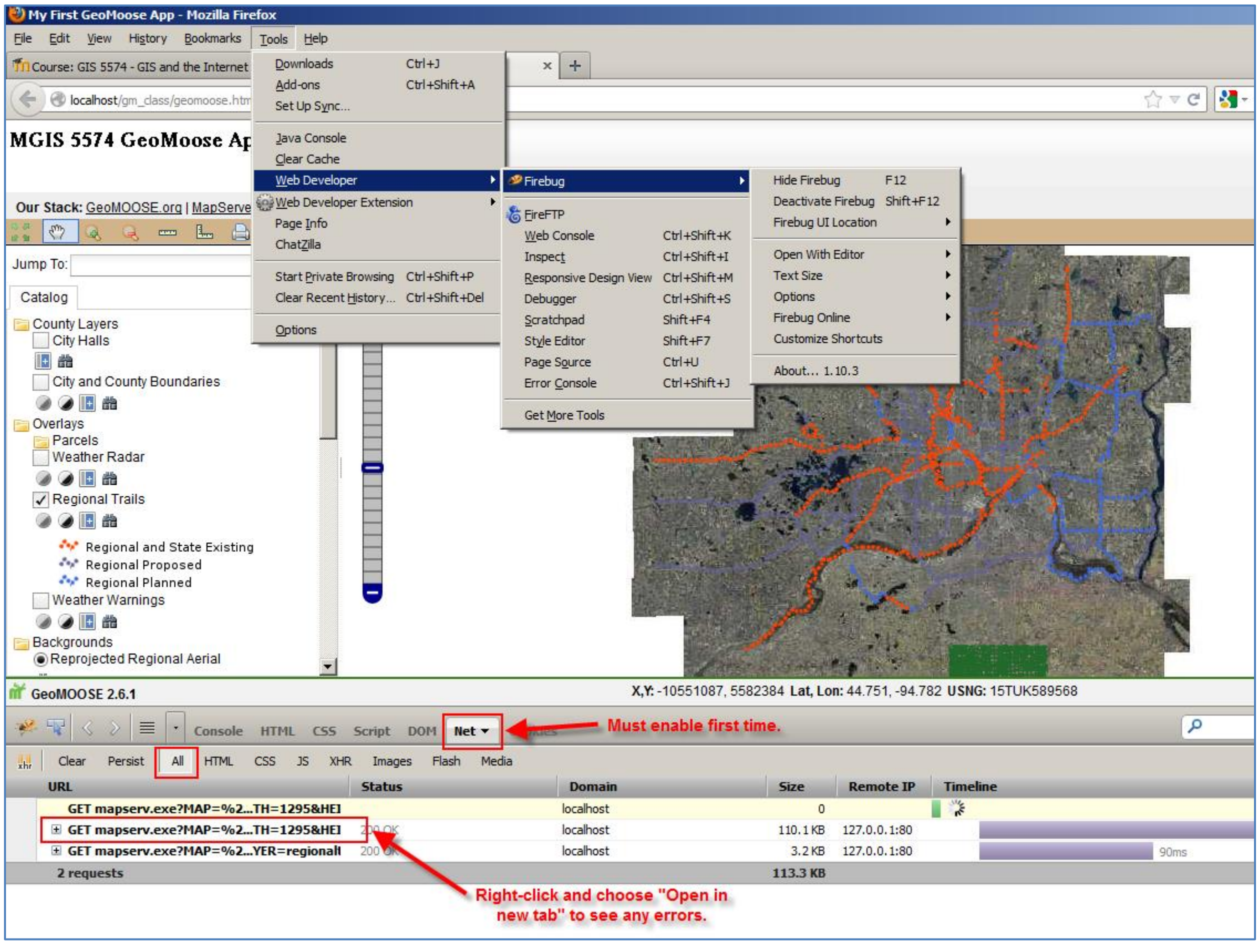

```
        <map-source name="cityhalls" type="mapserver">
                <file>./demo/countydata/cityhalls.map</file>
                <layer name="cityhalls" status="off"/>
        </map-source>

        <map-source name="metrogis" type="mapserver">
                <file>./demo/metrogis_shapefiles/metrogis.map</file>
                <layer name="regionaltrails" status="off"/>
        </map-source>

        <map-source name="parcels" type="mapserver">
                <file>./demo/parcels/parcels.map</file>
                <layer name="parcels" status="on"/>
                <param name="FORMAT" value="image/png;bits=8"/>
        </map-source>
```

8) Next add the catalog definition for the new Regional Trails layer to your map book. Notice we added a new group to our catalog called MetroGIS.


```
                <layer title="Weather Radar" src="iastate/nexrad-n0r" />
                <layer title="Regional Trails" src="metrogis/regionaltrails" />
                <layer title="Weather Warnings" src="nws_weather/1">
                        <metadata>http://rmgsc.cr.usgs.gov/ArcGIS/rest/services/nhss_weat
                </layer>
```

9) Now save the mapbook and mapfile. Then go to http://localhost/gm_class/geomoose.html and refresh your browser. You should see the new Regional Trails layer added to the application. If you happen to get a pink screen that likely means you have a syntax error in your mapfile. A great way to debug the mapfile syntax errors is to use firebug extension for Firefox. It will let you see every call made to the server. To use Firebug, go to Firefox and click Tools->Firebug->Open Firebug. Make sure you have the Net tab and All tab's selected. If this is the first time you have used firebug you may have to enable the Net tab to use it. If you scroll down to the bottom of the calls you should see some calls that start with "GET mapserv.exe". These tell you these are the calls for the MapServer type layers. Then if you look further to the right in the URL you will see the mapfile (.map) it is calling and the layer. You can right click on this URL and say open in new tab. This will show you any errors being sent back to the browser. If you see an image you know it is working correctly.

Firefox is going to make a call to the server for every image (whether it's a map layer or an image that's part of the interface) and all the other files required by the application such as CSS and JavaScript. See below for an example of Firebug. We will learn more about debugging tools in Exercise 10.

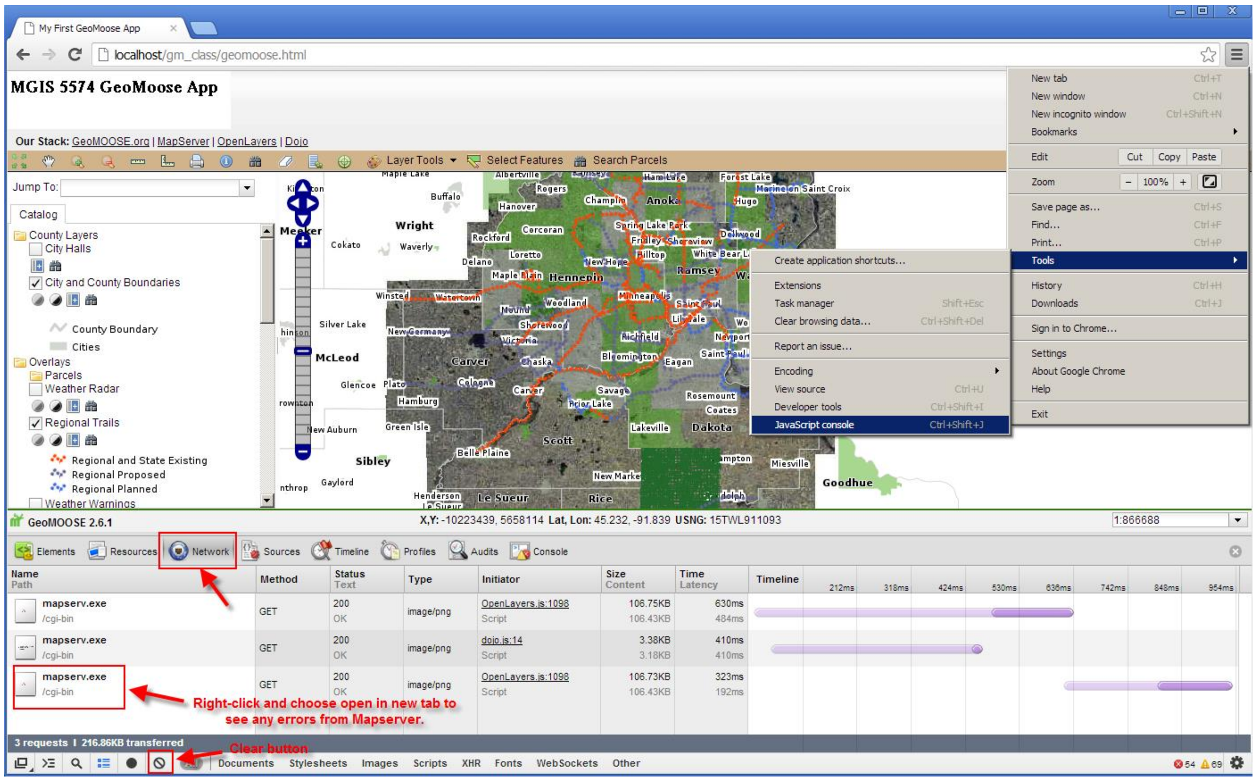


Must enable first time.

URL	Status	Domain	Size	Remote IP	Timeline
GET mapserv.exe?MAP=%2...TH=1295&HEI		localhost	0		
GET mapserv.exe?MAP=%2...TH=1295&HEI	200 OK	localhost	110.1 KB	127.0.0.1:80	
GET mapserv.exe?MAP=%2...YER=regionalt	200 OK	localhost	3.2 KB	127.0.0.1:80	90ms
2 requests			113.3 KB		

Right-click and choose "Open in new tab" to see any errors.

Firebug extension within Firebug. Must install as an add-on.



Name / Path	Method	Status / Text	Type	Initiator	Size / Content	Time / Latency
mapserv.exe /cgi-bin	GET	200 OK	image/png	OpenLayers.js:1098 Script	106.75KB 106.43KB	630ms 484ms
mapserv.exe /cgi-bin	GET	200 OK	image/png	dojo.js:14 Script	3.38KB 3.18KB	410ms 410ms
mapserv.exe /cgi-bin	GET	200 OK	image/png	OpenLayers.js:1098 Script	106.73KB 106.43KB	323ms 192ms

Right-click and choose open in new tab to see any errors from Mapserver.

3 requests | 216.86KB transferred

Web Developer Tools in Google Chrome.

10) An alternative way to debug you mapfiles is to use the command line utility called shp2img.exe. http://mapserver.org/utilities/shp2img.html

11) Now try on your own to add 2 more layers. Shopping Centers which is a point layer and Growth Management which is a polygon layer. The answers to adding these layers to the mapfile and mapbook can be found in the Week 9 exercise folder.

12) You should be aware of a few things. There is a include file that is being referenced in the mapfile called geomoose_globals. In this map file it has reference to a symbol file. The symbol file is just a place where the symbol definitions can be stored. You can open this file (C:\ms4w\apps\gm_class\maps\symbols\symbol.sym) in notepad to see the symbol definitions and names. Then in the class definition of each layer it references a symbol name from that file. These symbol definitions could also be added inline with STYLE parameters. The font's work in the same way. Notice a fontset file in C:\ms4w\apps\gm_class\maps\fonts\fontset.list) You can learn about more MapServer parameters for STYLE's and LABEL's by looking at the MapServer mapfile reference guide. http://www.mapserver.org/mapfile/label.html and http://www.mapserver.org/mapfile/style.html.

Notes: