

Introduction to GeoMOOSE:
An easy to use and free Web
mapping application

Len Kne

Houston Engineering, Inc.

Goals

- Short overview of GeoMOOSE
- Work with some of the common GeoMOOSE configuration files
- Know how to find helpful resources

About GeoMOOSE

- GeoMOOSE is a Web Client Javascript Framework for displaying distributed cartographic data. GeoMOOSE has a number of strengths including modularity, configurability, and delivers a number of core functionalities in its packages. GeoMOOSE is also very light weight for servers making it easy to handle a large number of users, with a large number of layers, and a large number of services without stressing a server.
- Built on other open source projects
- First started with the support of a FGDC grant in 2006
- Two major versions
- Handful of developers
- Many organization using GeoMOOSE



Klamath Basin Watershed Viewer - Mozilla Firefox
 McLeod County GIS - Mozilla Firefox
 Door County Web Map - Mozilla Firefox
 Pipestone County, MN Online Mapping - Mozilla Firefox
 GeoMOOSE 2.0 - Mozilla Firefox
 ASPCA PRO Viewer - Mozilla Firefox
 Crawford County - Mozilla Firefox
 MMCD Internal - Mozilla Firefox
 CoFIS 2.0 | Coffee Forest Information System - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.cofis.info/atlas/#

GeoMOOSE Apps

SourceForge.net: GeoMOOSE: geo... x CoFIS 2.0 | Coffee Forest Informa... x

CoFIS - Coffee Forest Information System

Reload CoFIS | Help | CoFIS Home | ECFF | Contact

Zoom to area: [dropdown]
 Layer catalog | Information

CoFIS map item query service
 Details on the location you clicked onto:

Ground coordinates
 X: 233123.35958
 Y: 859973.753281

Region information

Name:	Oromia
Population figures (EthioGIS, 2000)	
Total population:	18798039
Rural population:	16848815
Total density:	53
Rural density:	47

Wereda information

Name:	Seka Chekorsa
Region:	Oromia
Zone:	Jimma
Population figures (EthioGIS, 2000)	
Total population:	242302
Rural population:	234150
Total density:	171
Rural density:	166

Data sources

EthioGIS (2000): www.cde.unibe.ch
 CoFIS (2003-2009): www.cofis.info

Coffee Forest Information System

CoFIS 2.0 WebGIS: www.cofis.info/atlas
 CoCE project home: www.coffee.uni-bonn.de

CoFIS v. 2.0 UTM Coordinates: X,Y: 262007.44491, 990485.56430 Scale: 1:708271

Software Packages used for this training (MapServer/GeoMOOSE)

- MapServer (<http://mapserver.org>)
 - Map rendering engine
 - Reads and processes GIS data sources
- GeoMOOSE (<http://www.geomoose.org>)
 - JavaScript Framework
 - Provides an interface and configuration options for a web mapping application

GeoMOOSE Exercise 1: Start

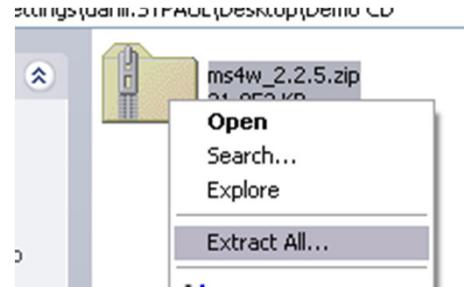
Exercise Title: Installing GeoMOOSE

Goals:

- Download MapServer and GeoMOOSE
- Configure Apache webserver
- Test MapServer and GeoMOOSE

1. Download and extract MS4W (MapServer for Windows)

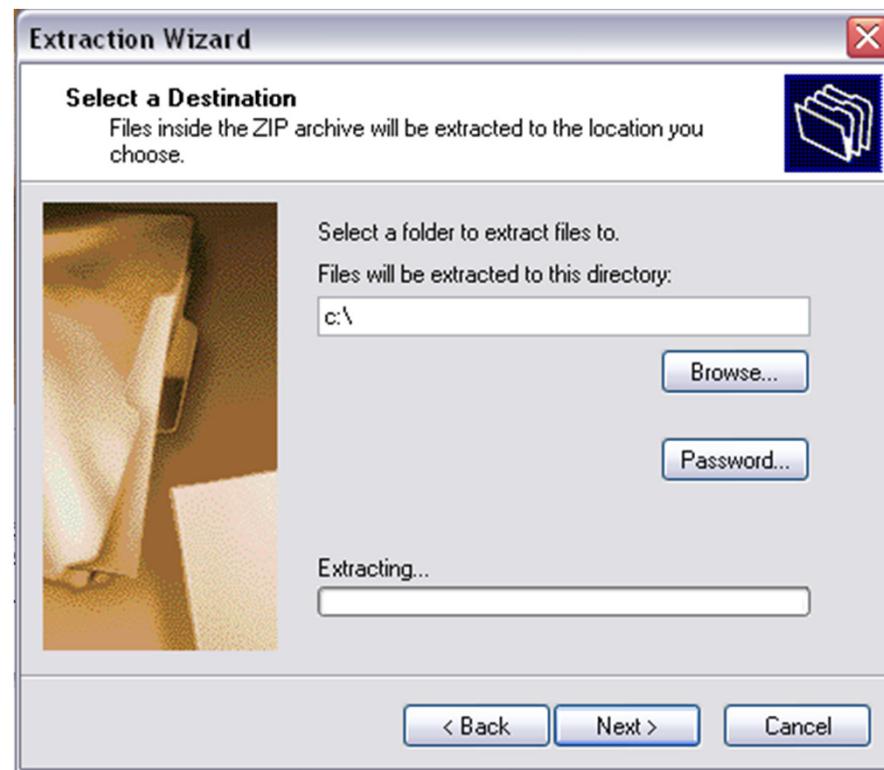
- Download from <http://www.maptools.org/ms4w/index.phtml?age=downloads.html>
- Right click on the ms4w zip file and click “Extract All...”



- **For this workshop, skip steps 1-3**

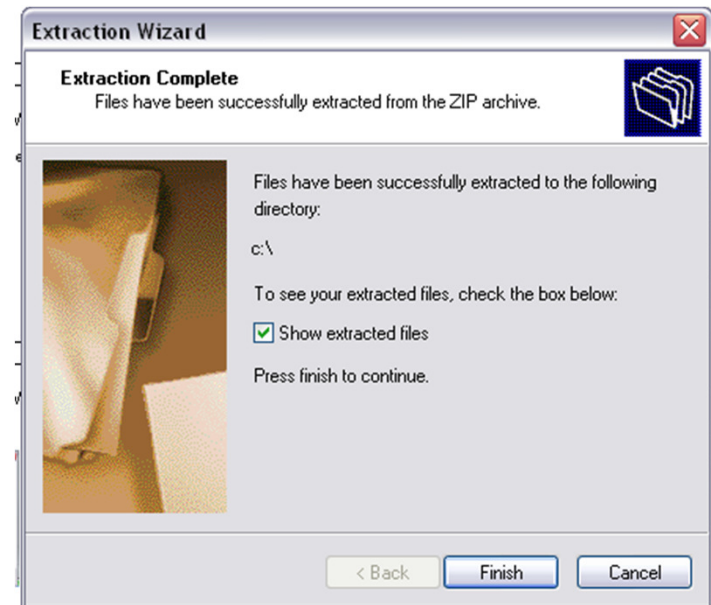
2. Extracting the MS4W Install

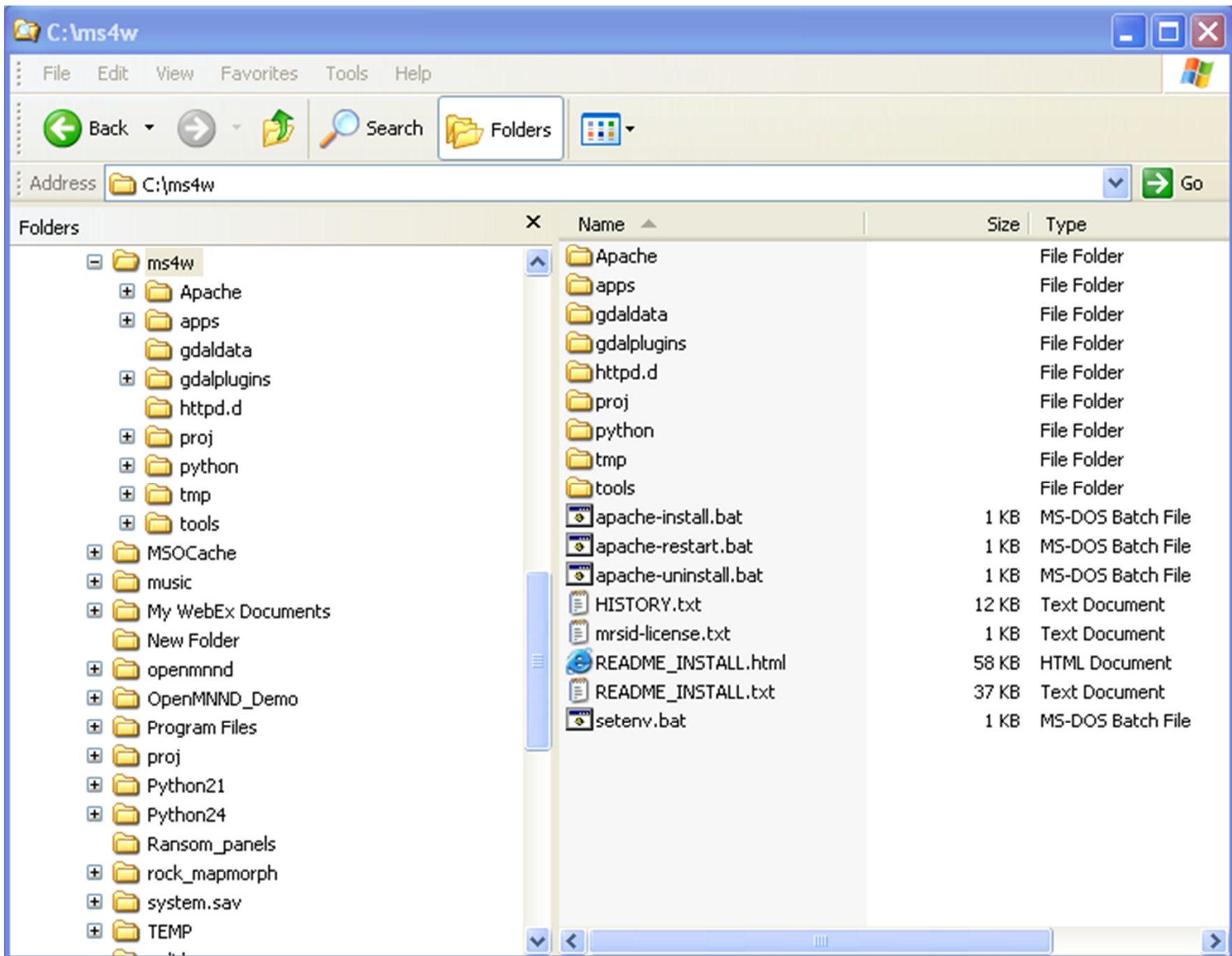
- Change the path to c:\ or d:\, this will create a “ms4w” folder on that drive.



3. Extracting MS4W Package

- Relax! The extraction takes about 3 minutes.
- You should see this when finished.



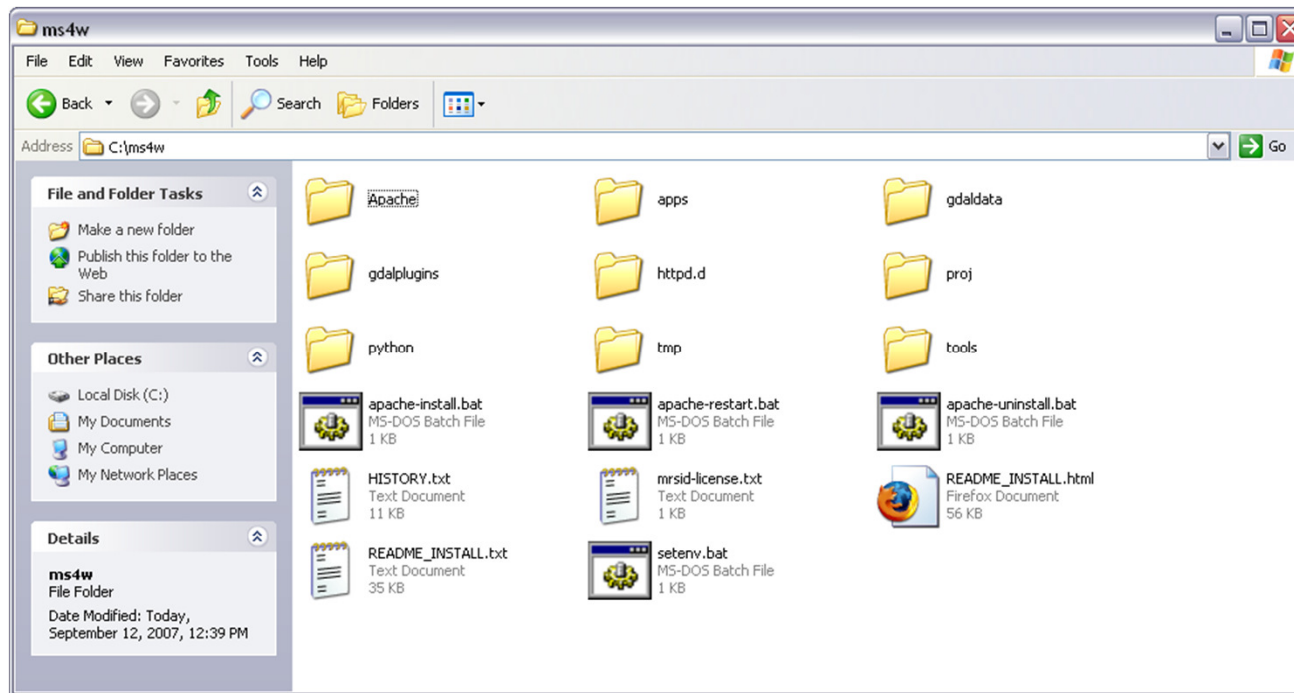


Substitution Warning!

- C:\ will be used for the rest of the presentation. If you used D:\ or any other drive letter for the MS4W install, continue as if C:\ translates to your drive letter of choice!

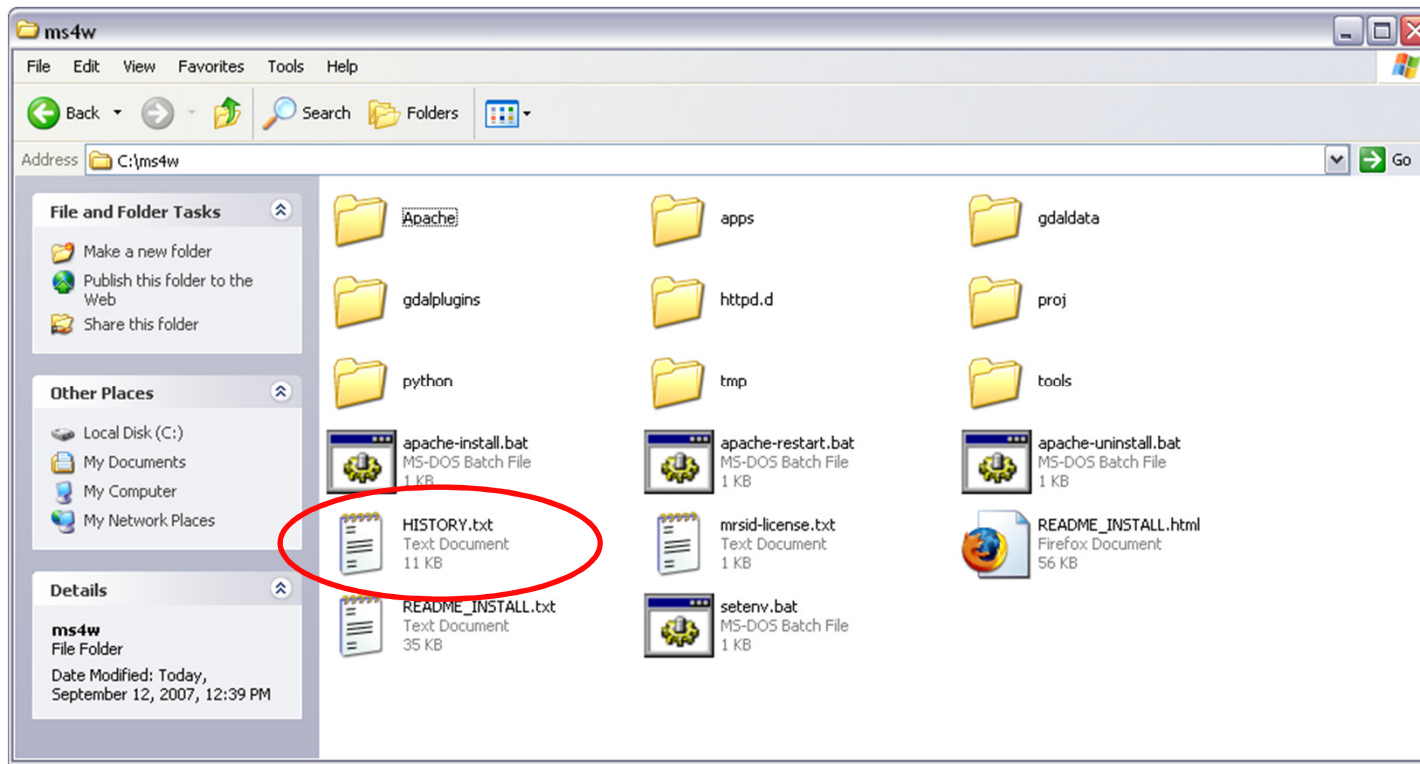
4. Get Apache Running

- Go to the C:\MS4W directory, it should look like this:



5. Get Apache Running

- Double click on the icon “apache-install.bat”
- **IF YOU HAVE IIS RUNNING STOP ALL IIS SERVICES!**

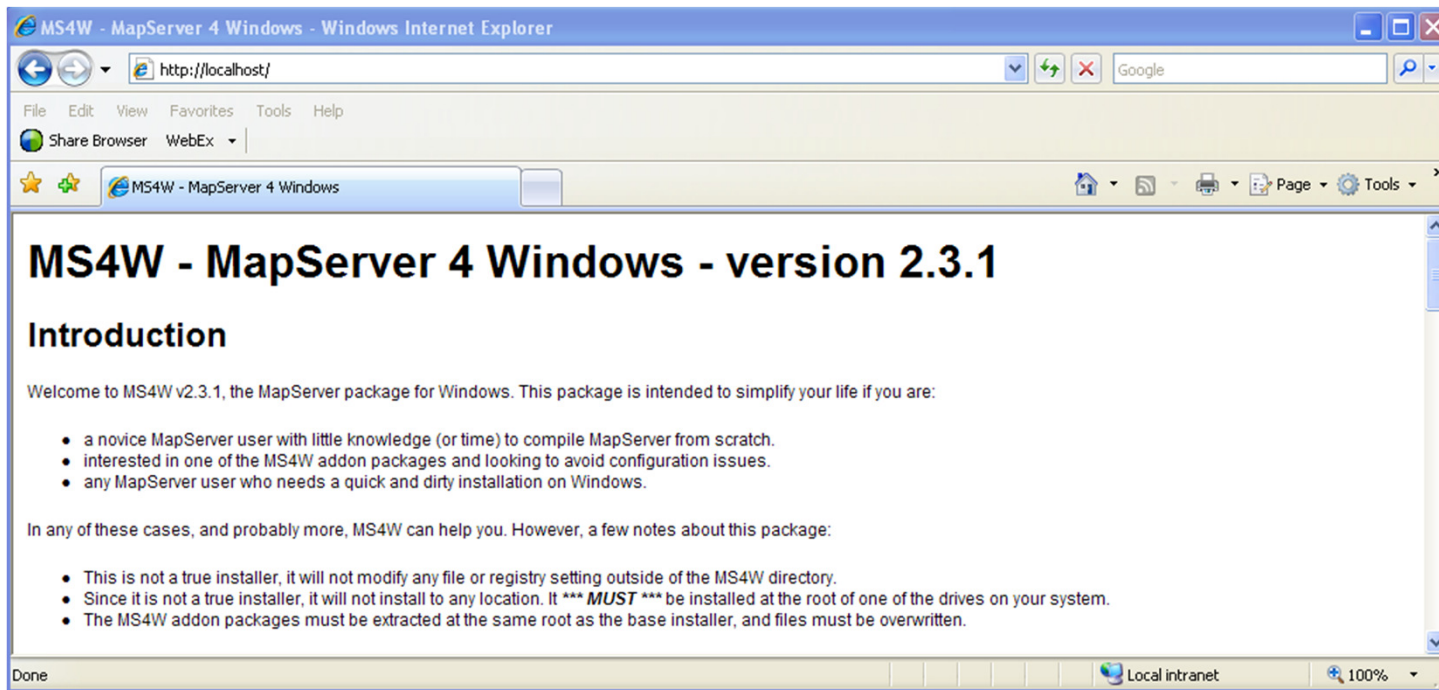


6. Get Apache Running

- After double clicking the icon a terminal window will pop up with some ugly diagnostic messages. This window will ultimately close.
- After this window closes, Apache should be running.
- *The `apache-install.bat` script installed a service. If you would like to remove the service later all you need to do is double-click the `apache-uninstall.bat` file.*

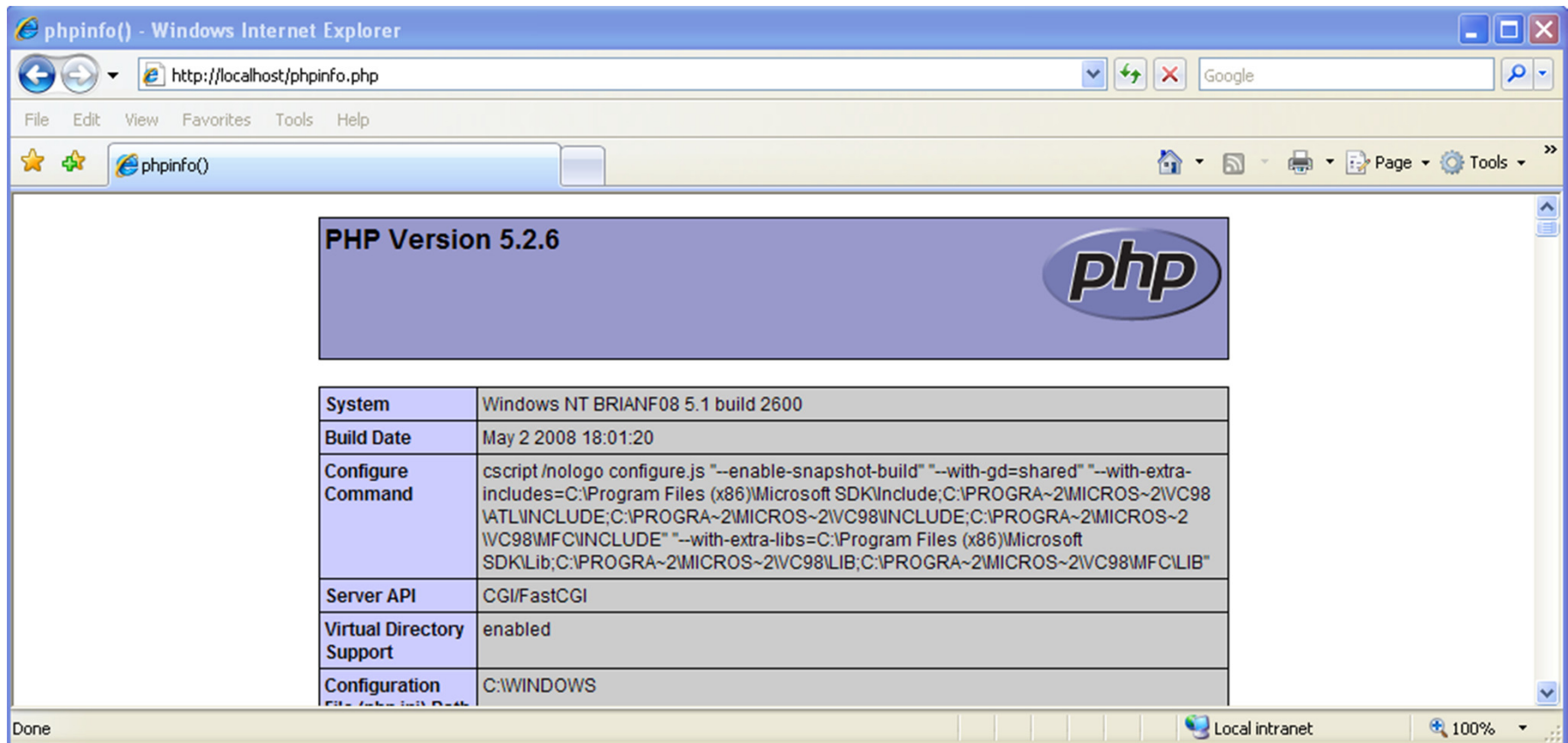
7. Test Apache and PHP

- The first test is to make sure that Apache is, in fact, working. Type <http://localhost> into your favorite Web-browser. It should look something like this:



8. Test Apache and PHP

- Now enter <http://localhost/phpinfo.php> into the Web Browser and you should see:



The screenshot shows a Windows Internet Explorer browser window titled "phpinfo() - Windows Internet Explorer". The address bar contains "http://localhost/phpinfo.php". The page content includes the PHP logo and a table of system information.

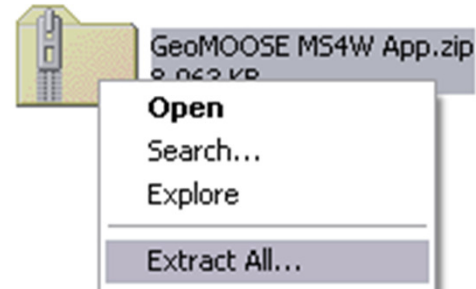
PHP Version 5.2.6	
System	Windows NT BRIANF08 5.1 build 2600
Build Date	May 2 2008 18:01:20
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared" "--with-extra-includes=C:\Program Files (x86)\Microsoft SDK\Include;C:\PROGRA~2\MICROS~2\VC98\ATL\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\MFC\INCLUDE" "--with-extra-libs=C:\Program Files (x86)\Microsoft SDKLib;C:\PROGRA~2\MICROS~2\VC98\LIB;C:\PROGRA~2\MICROS~2\VC98\MFC\LIB"
Server API	CGI/FastCGI
Virtual Directory Support	enabled
Configuration File (path and URL)	C:\WINDOWS

9. MS4W Package Installed

- You now have MS4W package installed which includes a variety of software tools:
 - View `C:\ms4w\README_INSTALL.html` for a complete description of the package
 - Major components
 - Apache (your web server)
 - MapServer (your mapping engine)
 - PHP (a server side programming language)

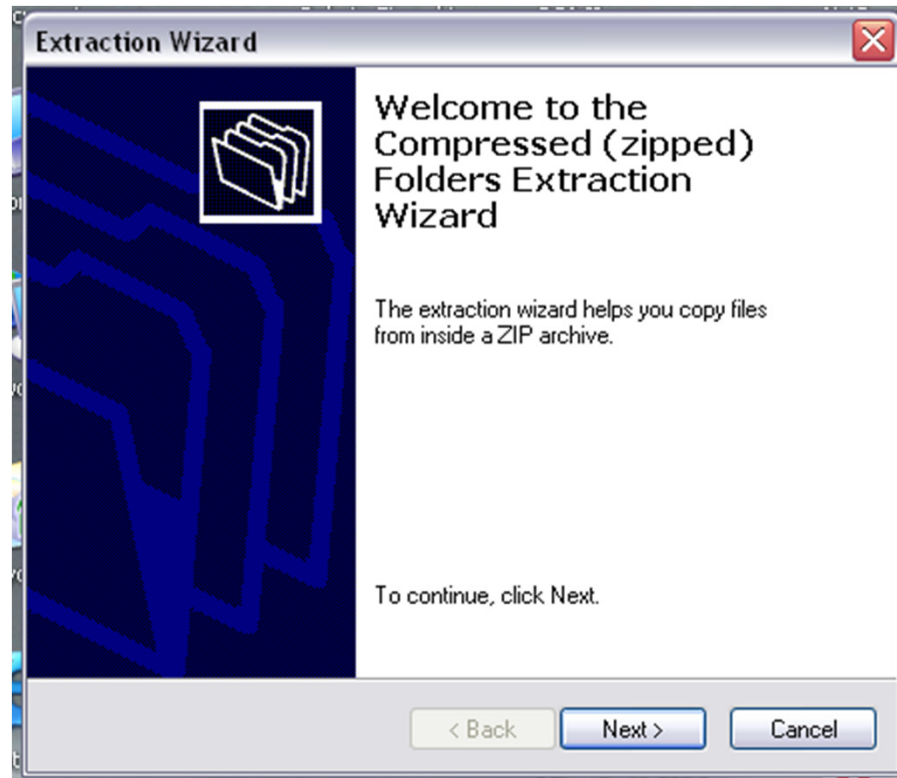
10. Install GeoMOOSE Code

- To Download GeoMOOSE
 - Plugs into MS4W package
 - <http://www.geomoose.org>
- GeoMOOSE-MS4W-2.2
- Click “Extract All...”



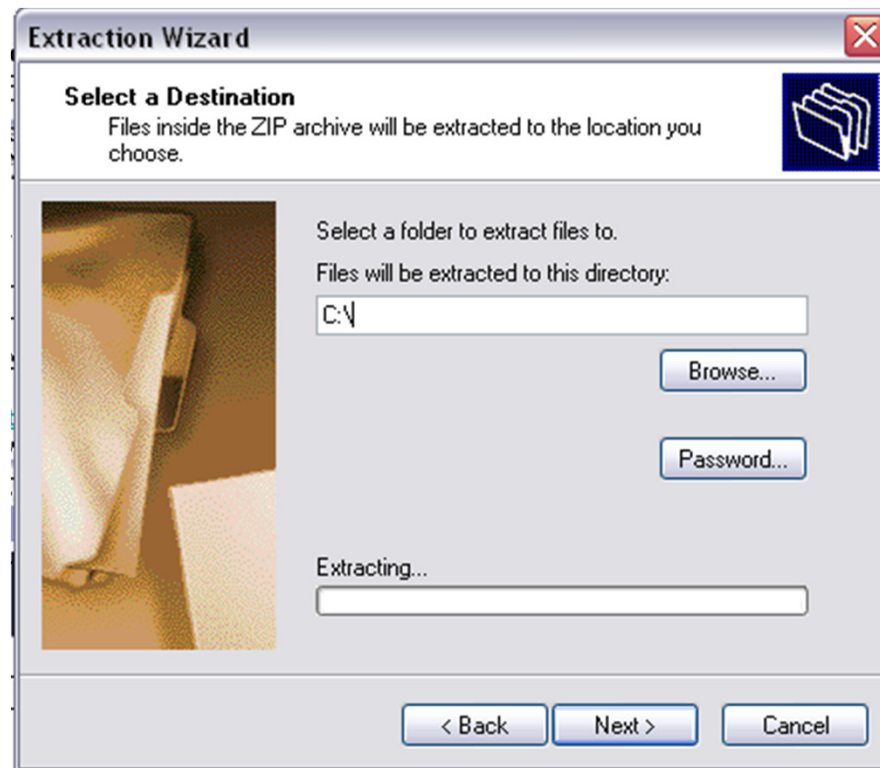
11. Extract GeoMOOSE Application

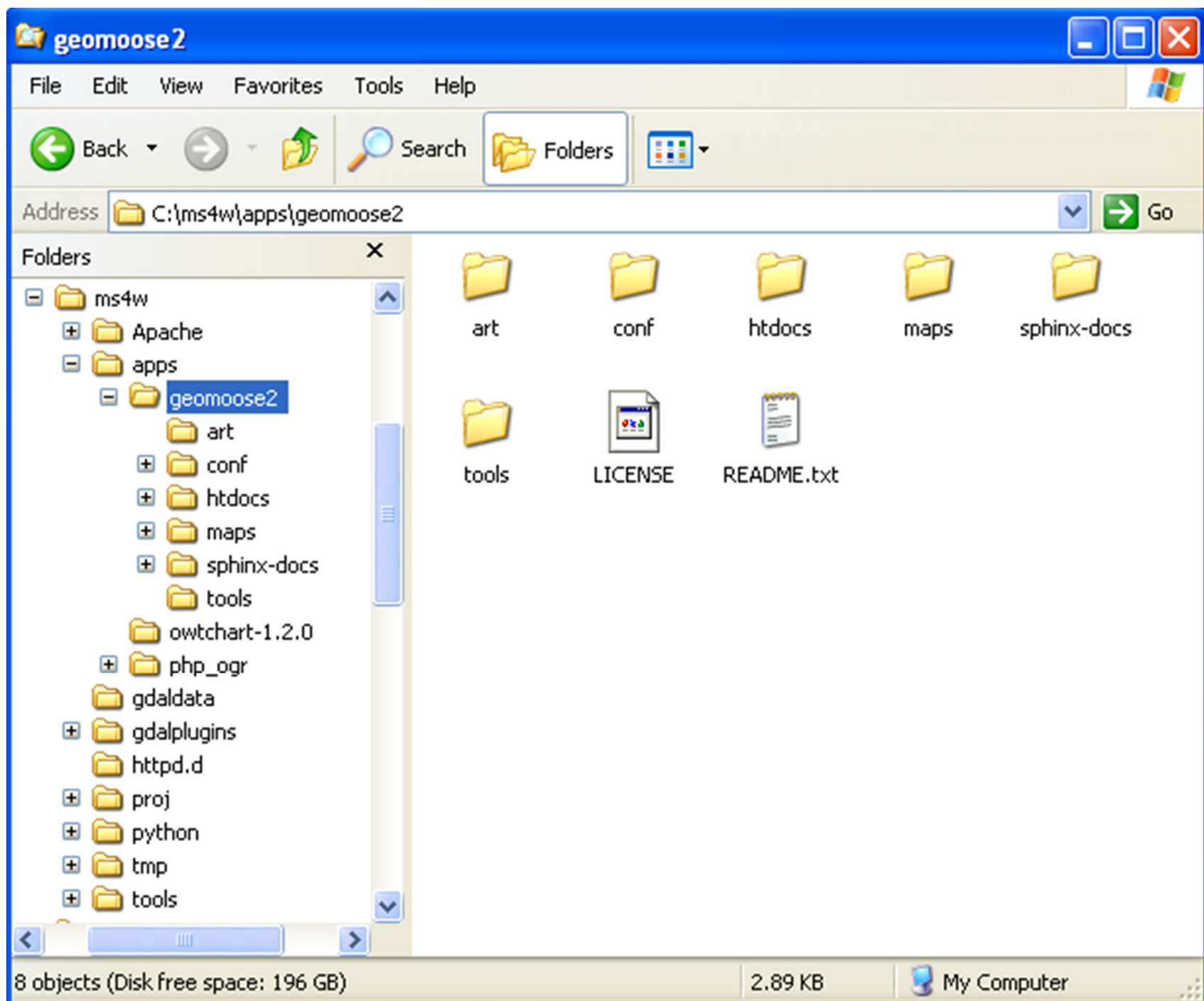
- Click Next



12. Extract GeoMOOSE Application

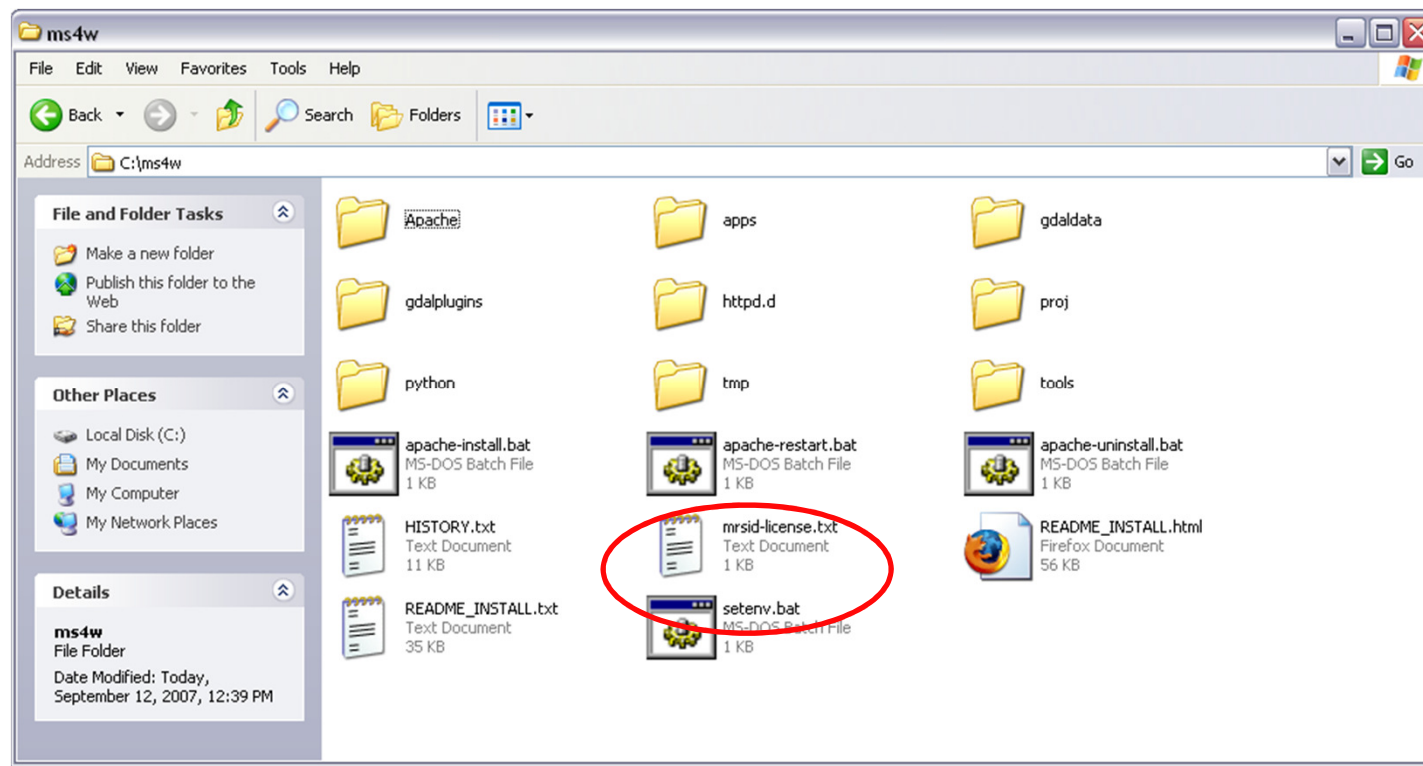
- Use the Same Path as the MS4W Install





13. Restart Web Server

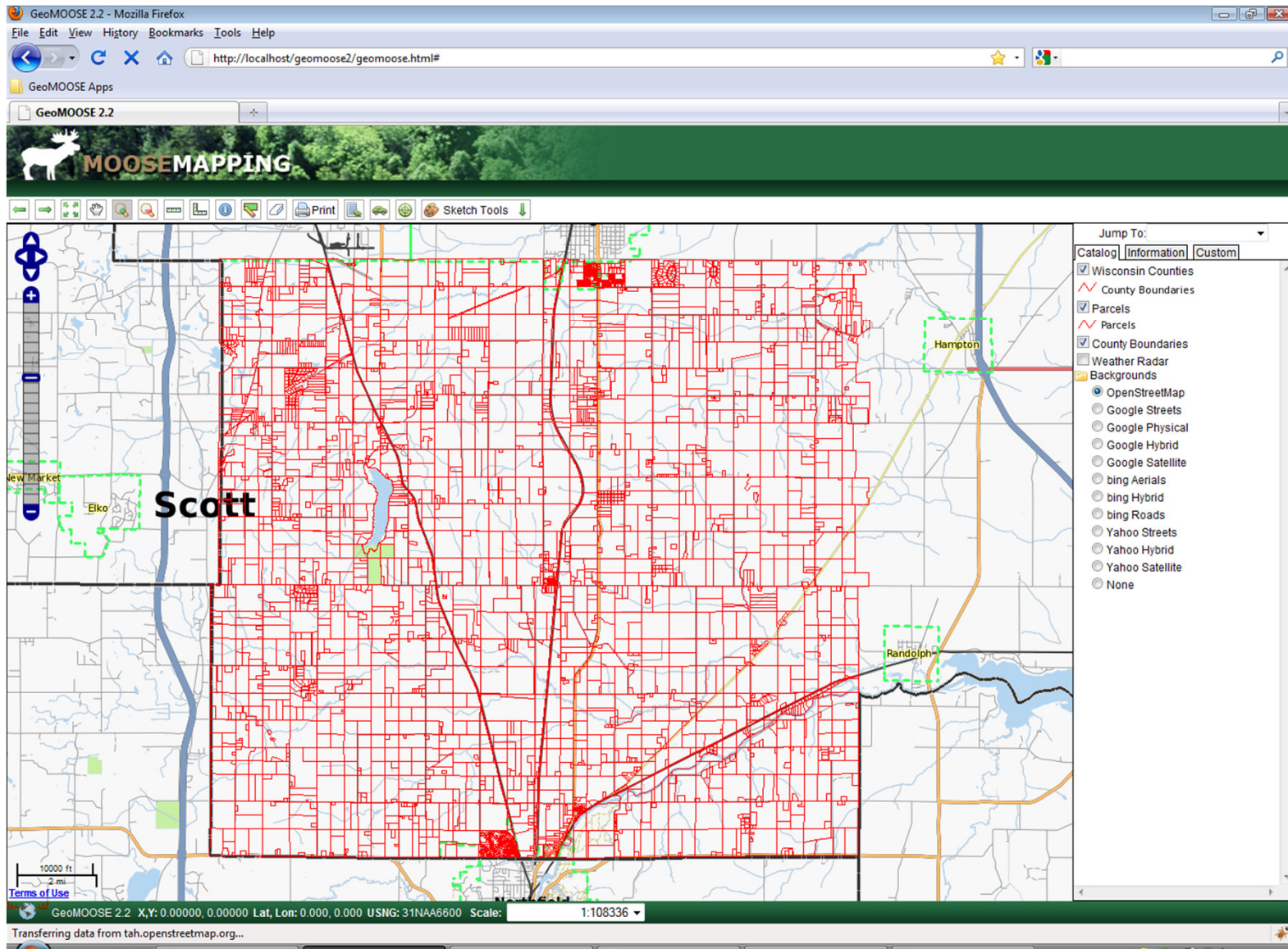
- Go Back to C:\MS4W
- Restart Apache by double clicking “apache-restart.bat”



14. Test the GeoMOOSE Demo App

- Clear the browser cache and go to <http://localhost/geomoose2/geomoose.html> in your web browser

You should now see....



GeoMOOSE Exercise 1: Finish

Exercise Title: Installing GeoMOOSE

Gotcha's

- Don't have administrator rights - run `c:\ms4w\Apache\bin\httpd.exe` from command prompt
- IIS is running on port 80 – change the port to 8080 in `c:\ms4w\Apache\conf\httpd.conf`
- Might need Microsoft Visual C++, get it here <http://www.microsoft.com/downloads/details.aspx?familyid=9B2DA534-3E03-4391-8A4D-074B9F2BC1BF&displaylang=en>

Important GeoMOOSE files

- **geomoose.html** – ms4w\apps\geomoose2\htdocs\geomoose.html – only “web” file for GeoMOOSE
- **Skins** – ms4w\apps\geomoose2\htdocs\skins – sets the color and images used
- **Mapbook** – ms4w\apps\geomoose2\conf\mapbook.xml – the “quarterback” of GeoMOOSE
- **Config** – ms4w\apps\geomoose2\conf\config.ini – check when first installing GeoMOOSE and changes to printing
- **Map files** – ms4w\apps\geomoose2\maps – symbology and service templates

Going Beyond HTML: XML

- XML (Extensible Markup Language) is a new markup language that is gaining popularity
- XML provides rules, guidelines, and conventions for representing data in a text file format
- XML also allows developers to represent relationships among different data values
- In an XML file, custom tags are created to define the meaning of the data (rather than its appearance)

Example of XML-formatted text file

```
<customers>
  <customer>
    <name>John Doe</name>
    <address>12345 Test Drive</address>
    <phone type="home">612-333-5555</phone>
  </customer>
  <customer>
    <name>Mary Jane</name>
    <address>8088 1st St</address>
    <phone type="work">763-468-9090</phone>
  </customer>
</customers>
```

mapbook.xml

- The “quarterback” of GeoMOOSE
- Defines
 - program defaults (<http://geomoose.org/docs/configuration.html>)
 - layers (<http://geomoose.org/howto/layer.html>)
 - services (<http://geomoose.org/docs/services.html>)
 - catalog (<http://geomoose.org/howto/layer.html>)
- The Mapbook is sent to the browser, thus anyone can read its contents. **Do NOT put passwords in the Mapbook!**
- Let’s take a look at:
`/ms4w/apps/geomoose/conf/mapbook.xml`

GeoMOOSE Exercise 3: Start

Exercise Title: GeoMOOSE MapBook Configuration Options

Goals:

- Gain a general understanding of the GeoMOOSE “mapbook.xml” configuration file.
- Discuss each Mapbook configuration parameter
- Learn how to change a few configuration parameters to modify new GeoMOOSE application

GeoMOOSE Exercise 3: Concepts

Understanding the GeoMOOSE “mapbook.xml” configuration parameters

- “C:\ms4w\apps\geomoose2\conf” folder – Stores GeoMOOSE configuration Files
 - “mapbook.xml” – xml file that is the main configuration file for the GeoMOOSE interface
 - Reference guide - <http://www.geomoose.org/moose/docs/mapbook.html>
 - Needs to be a valid xml file
 - Contains 7 major elements (mapbook, **configuration**, map-source, service, layer-controls, catalog, and toolbar (<http://www.geomoose.org/moose/docs/configuration.html>))
 - settings.ini may need to be changed if you install MS4W in a different location or need to change print settings (<http://geomoose.org/docs/settings.html>)

GeoMOOSE Exercise 3: Steps

Modify configuration parameters, such as jump to and links.

- 1) Open the mapbook.xml file for the new application you setup in Exercise 1. It is located in C:\ms4w\apps\geomoose2\conf\mapbook.xml.
- 2) At the top you will see some configuration options.
- 3) Add a link to the links bar in line 4.
- 4) Add a new extent for the State of Wisconsin to the “Jump to Drop down” as shown in line 8.

```
1 <?xml version="1.0"?>
2 <mapbook version="2.0">
3   <configuration>
4     <param name="links_bar_html"><![CDATA[<a href="http://www.geomoose.org" target="_blank">GeoMOOSE Website</a>]]></param>
5     <param name="zoomto['Jump To:']"><![CDATA[
6       {
7         'Dakota County' : [521238.614537864,4924218.86673578,473921.947801381,4974430.36885032],
8         'My New Extent' : [501238.614537864,4984218.86673578,493921.947801381,5074430.36885032],
9         'Parcel Data' : [497205.409367,4923984.423582,477595.805945,4941970.52988],
10        'Full State of MN' : [189783.560000,4816309.330000,761653.524114,5472346.500000]
11      }
12    ]]]></param>
13   <param name="max_extent">189783.560000,4816309.330000,761653.524114,5472346.500000</param>
14   <param name="initial_extent">497205.409367,4923984.423582,477595.805945,4941970.52988</param>
15
```

- 5) Save the mapbook.xml file, open <http://localhost/geomoose2/geomoose.html>, clear the cache and refresh the page. You should now see the link above the toolbar and the new item in the “Jump to” tool.

Note: <![CDATA[]] is a special tag to put HTML code in an XML document

GeoMOOSE Exercise 3: Finish

Exercise Title: GeoMOOSE MapBook Configuration Options

Lessons Learned:

- Gained a general understanding of the GeoMOOSE “mapbook.xml” <configuration> element.
- Learned how to add links to the interface and modify the “jump to” tool.

HTML

- HTML stands for Hypertext Markup Language
- Was developed in 1991 specifically for WWW documents
- Define the structure of a Web page
 - Document parts such as title, headings, paragraphs
 - Presentation such as boldface (``)
 - Hypertext links to other HTML pages (`<a>`)
 - Comments `<!-- -->`
- Has a fixed set of tags, including `<html>`, `<head>`, `<body>`, and many more ...
- HTML is a cross-platform language that works on Windows, Mac, and UNIX platforms
- HTML pages are plain text files that have tags embedded in the text

geomoose.html

- The “web page” for GeoMOOSE
- Defines
 - what CSS file to use
 - tabs
 - title
 - projection file
- Let’s look at geomoose.html in:
`/ms4w/apps/geomoose2/htdocs`

CSS

- Stands for cascading style sheets
- Adds styling to tags
- Can be defined in external file or `<style>` tag inside HTML file

green.css

- This is the main “skin” file
- Controls colors, font sizes, images, etc...
- Let’s look at an example:
`/ms4w/apps/geomoose2/htdocs/skins/green/green.css`

GeoMOOSE Exercise 2: Start

Exercise Title: Creating a Skin and modifying Logo and Colors

Goals:

- Gain a general understanding of the GeoMOOSE user interface layout
- Learn how you to create your own skin for a GeoMOOSE interface
- Learn how to modify the GeoMOOSE CSS files to change colors and fonts

GeoMOOSE Exercise 2: Concepts

Understanding how to control the GeoMOOSE User Interface

- Htdocs – Web accessible files
 - Geomoose.html – default page that loads application
 - Geomoose – contains all javascript source files. Combined into one called geomoose.js
 - PHP – contains all the scripts that control services
 - **Skins – contains the look of your user interface**
 - **CSS – controls colors, fonts, and layout**
 - Images – icons, etc...
 - Proj4js – contains projection library, don't need to worry unless adding a new projection for your app.
 - OpenLayers – contains OpenLayer library, don't need to worry about this folder

GeoMOOSE Exercise 2: Concepts

Understanding how to control the GeoMOOSE User Interface

- GeoMOOSE Skins – GeoMoose uses a concept of skins to modify and apply a new color scheme and graphics to your application.

- Here a GeoMOOSE “How To” document on the topic <http://geomoose.org/howto/skin.html>
- The skin is referenced in the geomoose.html web page or your default page you load

```
37     <!--  
38         This is where your skin is defined.  
39         For an example, comment out the line containing "green.css" and  
40         uncomment "blue.css".  
41     -->  
42     <link type="text/css" rel="stylesheet" href="skins/green/green.css"/>  
43 <!--     <link type="text/css" rel="stylesheet" href="skins/blue/blue.css"/> -->  
44
```

- In the skins folder you will find the .css files and all images referenced in it. This includes the logo or header image.
- Firefox’s Firebug extension is a great way to explore a web pages CSS names and properties
- Let’s take a look at the demo with Firebug.

GeoMOOSE Exercise 2: Steps Creation of your own skin.

- 1) Duplicate the entire blue skin folder by copy and pasting C:\ms4w\apps\geomoose2\htdocs\skins\blue. Call the duplicated folder myskin.
- 2) Inside the myskin folder copy and paste the green.css file and call it myskin.css
- 3) Open "geomoose.html" in a text editor. In the <head> section there should be a tag starting <link... which mentions the skin. Comment out all of the skins you

```
<!--  
    This is where your skin is defined.  
    For an example, comment out the line containing "green.css" and  
    uncomment "blue.css".  
-->  
<!--<link type="text/css" rel="stylesheet" href="skins/green/green.css"/> -->  
<!--<link type="text/css" rel="stylesheet" href="skins/blue/blue.css"/> -->  
<link type="text/css" rel="stylesheet" href="skins/myskin/myskin.css"/>
```

- 4) Next let's change the header logo. The header logo is referenced in the myskin.css file. It is called logo_top.jpg

GeoMOOSE Exercise 2: Steps Creation of your own skin.

- 5) Create a new top_logo.jpg image. It just needs to be 670 pixels wide by 59 pixels high. (optional)
- 6) Now open your duplicated GeoMOOSE application, clear the cache and refresh the page. You should see the new logo in the application.
- 8) Next let's play around with some colors and fonts. Open myskin.css file
- 9) Find the **#toolbar** class and change the background to a tan color (#CCCC99).
Results should look like this:

```
1  body {
2      background-color: #4068AB;
3      font-family: Arial, Helvetica, sans-serif;
4      font-size: 75%;
5      font-style: normal;
6      font-weight: normal;
7      text-decoration: none;
8      color: #000000;
9      margin: 0px 0px 0px 0px;
10 }
11
12
13 #header {
14     background-image: url('images/logo_top.jpg');
15     background-repeat: no-repeat;
16 }
17
18 #toolbar {
19     background: #CCCC99;
20 }
21
22
```

GeoMOOSE Exercise 2: Finish

Exercise Title: Creating a Skin and modifying Logo and Colors

Lessons Learned:

- Gained a general understanding of the GeoMOOSE user interface and how it is put together
- Created a new skin for your new GeoMOOSE application
- Learned how to modify GeoMOOSE user interface CSS files
- There are other GeoMOOSE CSS files that control other parts of the layouts. They are located in C:\ms4w\apps\geomoose2\htdocs\css. Feel free to experiment, but be careful it is valid CSS syntax!

GeoMOOSE Exercise 4: Start

Exercise Title: Removing and adding a Layer to GeoMOOSE

Goals:

- Gain a general understanding of how layers are displayed and added to GeoMOOSE.
- Discuss the **<map-source>** and **<catalog>** elements in the mapbook.xml file.
- Learn how to remove and add a layer from a GeoMOOSE application.

GeoMOOSE Exercise 4: Concepts

Understanding the GeoMOOSE “mapbook.xml” <map-source> and <catalog> elements

- “C:\ms4w\apps\geomoose2\conf” folder – Stores GeoMOOSE configuration Files
 - “mapbook.xml” – xml file that is the main configuration file for the GeoMOOSE interface
 - Reference guide - <http://www.geomoose.org/moose/docs/mapbook.html>
 - Needs to be a valid xml file
 - The layers that are displayed in the application are in the order they appear in the <catalog>.
 - Relevant “How To” guides
 - <http://www.geomoose.org/moose/howto/layer.html>
 - <http://www.geomoose.org/moose/docs/projections.html>
 - <http://mapserver.org/mapfile/index.html#mapfile>

GeoMOOSE Exercise 4: Steps

Remove the a layer and add another.

- 1) There are two places in the Mapbook that need to be deleted to remove a layer. You could also comment out a layer if you might need it in the future.
Comment by: `<!-- -->`
- 2) Let's delete the "County Boundaries layer. Start in the `<catalog>` and find:

`<layer title="County Boundaries" src="borders/county_borders" status="on" show-legend="false" legend="false"/>`
- 3) We are going to delete this line, but first take note of the source, this will tell you what `<map-source>` is associated with this layer
- 4) Go to the `<map-layers>` and look for "borders" If "county_borders" is the only layer, then the entire `<map-source>` can be deleted. If there are other layers, then only delete the "layer" entry.
- 5) Save the Mapbook and try running GeoMOOSE in the browser

GeoMOOSE Exercise 4: Steps

Remove the a layer and add another.

- 1) Let's add a layer showing county forests. Scroll down and find the `<map-source>` called "wi". Copy the layer and name the copy "county_forests". It should look like this:

```
<map-source name="wi" type="mapserver">  
  <file>./wi/county/wi.map</file>  
  <layer name="county_bnds"/>  
  <layer name="county_forests"/>  
</map-source>
```

- 2) Now add the line below to the `<catalog>` block:

```
<layer title="Wisconsin County Forest" src="wi/county_forests" status="off"  
show legend="true" popups="true" minscale="0" maxscale="150000"  
legend-show="true" />
```

- 3) Save the mapbook and re-load GeoMOOSE to see if it works

GeoMOOSE Exercise 4: Finish

Exercise Title: Removing and Adding a Layer

Lessons Learned:

- Gained a general understanding of the GeoMOOSE “mapbook.xml” **<map-source>** and **<catalog>** elements.
- Learned how to remove and add layers to a application.

MapServer Map Files

- Text based, hierarchical
- Control all aspects of a layer
 - symbology, colors, labels
 - connection to the data source (map files, database, etc...)
 - connection to templates for Identify, Select, Buffer services
- Have a .map extension

MAP

NAME 'mn_county'
SIZE 800 650
STATUS ON
EXTENT 189783.560000 4816309.330000 761653.524114 5472346.500000
UNITS METERS
FONTSET './fonts/fontset.list'
SYMBOLSET './symbols/symbol.sym'
IMAGETYPE PNG

LEGEND
STATUS ON
END

LAYER # County Boundary Polygon Outline Layer

NAME 'county_borders'
DATA './county.shp'
STATUS DEFAULT
TYPE POLYGON
LABELITEM 'COUNTYNAME'
CLASS
NAME "County Boundary"
STYLE
SYMBOL 'countydashed'
SIZE 2
COLOR -1 -1 -1
OUTLINECOLOR 50 50 50
END

LABEL
TYPE TRUETYPE
FONT vera_sans-bold
FORCE TRUE
SIZE 9
COLOR 0 0 0
OUTLINECOLOR 232 232 232

END

END

END # County Boundary Polygon Layer

END ## end Map

Map File Caveats

- Case insensitive (with few exceptions like shape file name and attributes)
- 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, - or _. All other strings must be quoted
- Order of layers is significant - FIFO

Map File Help

- Map File Reference Guide on MS Website
 - <http://mapserver.org/mapfile/index.html#mapfile>
- Symbols Help
 - <http://mapserver.org/mapfile/symbology/construction.html>

Geo Exercise MOOSE 5: Start

Exercise Title: Learning MapServer Configuration Files

Goals:

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

GeoMOOSE Exercise 5: Concepts

Understanding MapServer Mapfiles

- 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, - or _. All other strings must be quoted.
- Order of layers is significant - FIFO.

GeoMOOSE Exercise 5: Steps Creating a MapServer mapfile

- 1) Take a look at the map file for Wisconsin Counties at:
`/ms4w/apps/geomoose2/maps/wi/county/wi.map`
- 2) We'll walk through the map file together
- 3) Change the outline color for counties to blue
- 4) Change the label color for counties to blue

GeoMOOSE Exercise 5: Finish

Exercise Title: Creating a MapServer Mapfile

Lessons Learned:

- Gain a general understanding of MapServer Configuration Files (.map)

GeoMOOSE Exercise 6: Start

Exercise Title: Debugging MapServer Layer Type

Goals:

- Gain a general understanding of different tools for debugging mapfiles (.map)

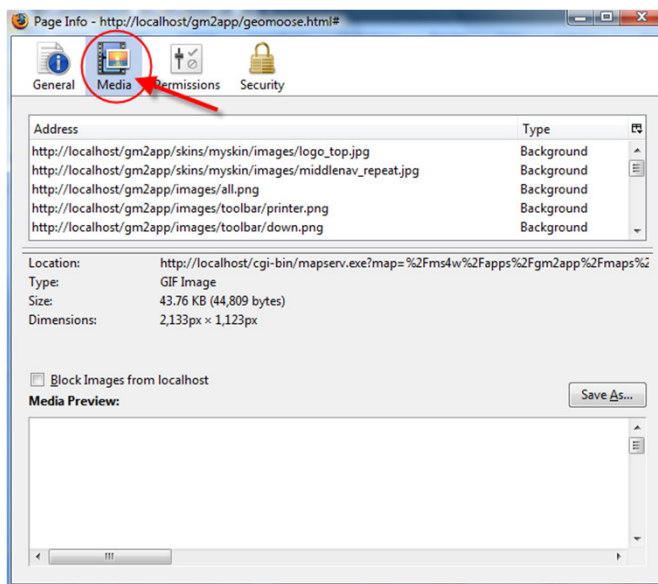
GeoMOOSE Exercise 6: Concepts

Debugging MapServer Layers

- Errors are usually caused by syntax problems
- GeoMOOSE will display a pink screen if a layer is not returning a valid image
- Three good tools for debugging problems with MapFiles
 - Page Info inside Firefox
 - Firebug Extension of Firefox
 - Fiddler for Internet Explorer

GeoMOOSE Exercise 6: Steps Debugging a MapServer Layer Type

- 1) Open /ms4w/apps/geomoose2/maps/wi/county/wi.map you created in Exercise 5.
- 2) Purposely mis-spell the data source (shapefile name) for the “county_bnd” layer
- 3) Save the wi.map file
- 4) Open GeoMOOSE (<http://localhost/geomoose2/geomoose.html>) and turn on the County Boundary layer. You should see a pink screen because the image source is stack on the top of the other layers.
- 5) Now go to Tools->Page Info menu in Firefox
- 6) Next click on the “Media” tab



GeoMOOSE Exercise 6: Steps

Debugging a MapServer Layer Type

- 7) Find the URL (image source) that has the wi.map reference in it (it will have “mapserv” in it)
- 8) Next copy this URL and paste it into a new browser window or tab. You should see this error.
*“msDrawMap(): Image handling error. Failed to draw layer named 'county_bnds'.
msShapefileOpen(): Unable to access file.
(/ms4w/apps/geomoose2/maps//wi/county/county_bnxds.shp) msShapefileOpen():
Unable to access file. (/ms4w/apps/geomoose2/maps//wi/county/county_bnxds.shp) ”*
- 9) *This is telling you it can not access the “county_bnxds.shp” file. Now we know this is because it is misspelled, but it could be a path issue as well.*
- 10) *Now open the Firebug extension in Firefox. It is under the tools menu.*
- 11) *Make sure you have these options active. Then toggle the broken layer on and off. You will then see the call made from the browser to the server and the response back from the server show up in Firebug. Once again we see a very similar response as above.*

GeoMOOSE Exercise 6: Finish

Exercise Title: Debugging a MapServer Mapfile Layer

Lessons Learned:

- Learned 2 methods for debugging a MapServer mapfile error

How interactive web mapping applications work...

- All solutions work off the same basic information (pan/zoom/query):
 - geographic extent of an image
 - size of that image in pixels
 - image coordinate via mouse click
 - layers of interest (optional)
 - zoom in/out and size (optional)
- How the information is packaged and sent to the server, and then returned to the client varies

The chain of files

- All of the files we have been working with connect together to display the data
- Splitting the functions among several tables gives a lot of power and flexibility... but makes it somewhat difficult to remember how they all fit together

The chain of files

1. Start with the <catalog> in the Mapbook – this points to a <map-source>
2. Still in the Mapbook, look at the <map-source> - it gives you the name of the map file
3. Follow the path to the map file – the “data” tag will give you the name of the shape file (other data source)

Important GeoMOOSE files

- **geomoose.html** – ms4w\apps\geomoose2\htdocs\geomoose.html – only “web” file for GeoMOOSE
- **Skins** – ms4w\apps\geomoose2\htdocs\skins – sets the color and images used
- **Mapbook** – ms4w\apps\geomoose2\conf\mapbook.xml – the “quarterback” of GeoMOOSE
- **Map files** – ms4w\apps\geomoose2\maps – symbology and service templates
- **Config** – ms4w\apps\geomoose2\conf\config.ini – check when first installing GeoMOOSE and changes to printing

GeoMOOSE Tools and Services

- GeoMOOSE uses a concept of Services for some of the tools
- Most tools are either call a server-side service (PHP) in most cases or a javascript function
- Services are defined in the mapbook
- This gives developers the flexibility of creating their own tools and services
 - <http://www.geomoose.org/moose/docs/services.html>

GeoMOOSE Tools and Services

- GeoMOOSE processes templates by the reference in the metadata object of a mapfile
- Lots of moving parts
- MapServer Reference Guides used by GeoMOOSE Services:
 - Some services make use of MapServer PHP MapScript to perform attribute searches and spatial queries (<http://mapserver.org/mapscript/php/index.html>)
 - Some services have their results sent back to GeoMOOSE by using a MapServer Query Template (<http://mapserver.org/mapfile/template.html>)
 - Some services take advantage of MapServer variable substitutions (http://mapserver.org/mapfile/variable_sub.html)

GeoMOOSE Exercise 7: Start

Exercise Title: Configuring the Identify Service/Tool

Goals:

- Gain a general understanding of defining services
- Learn about the GeoMOOSE “Identify Service”

GeoMOOSE Exercise 7: Concepts

Identify Service

Purpose: Drill down through all visible layers that are configured to be identified and report attributes back to the user in the table of contents.

User Interaction: User activates the tool and then clicks on the map.

Service Script Source: identify.php

Notes: Only layers that are configured to be identified in the .map file metadata object will get queried by mapserver. This service makes use of PHP mapscript and the “QuerybyShape” function. The results are returned using a MapServer Query Template.

GeoMOOSE Exercise 7: Steps Configuring the Identify Service

- 1) Open C:\ms4w\apps\geomoose2\conf\mapbook.xml and find the “Identify Service” definition. Should look like this:

```
<service name="identify" title="Identify" display="false">  
  <url>php/identify.php</url>  
  <step type="spatial" name="shape" line="false" polygon="false" jump-start="true" default="point">  
    <input type="visiblelayers" name="layers"/>  
  </step>  
</service>
```

- 2) Next find the tool definition for the Identify Tool. Should look like this:

```
<tool name="identify" title="Identify" type="service" service="identify"/>
```

- 3) Notice how the service attribute in the tool definition references the service name. Since the Identify service is already setup in this application you will not need to modify anything in the mapbook. For a description of what some of the attribute options are in the service definition, please read the mapbook reference guide.
- 4) Next we are going to make the “county_bnd” layer identifiable.

GeoMOOSE Exercise 7: Steps Configuring the Identify Service

- 5) Open the wi.map file. Add a metadata object to the “county_bnd” layer object. This will tell GeoMOOSE when it is processing the visible layers whether or not this layer is configured to be identified and which query template to use. Your result should look like this:

```
METADATA
    'identify_record'    'identify_county_forest.html'
END
```

- 5) Remember to fix your “county_bnd” shapefile name if you are using the .map file from the previous exercise.

GeoMOOSE Exercise 7: Steps Configuring the Identify Service

- 6) Next create the query template file that MapServer will use to process the query results. You can read more about MapServer templating at: <http://mapserver.org/mapfile/template.html>
- 7) To do this create a .html file and call name it “identify_county_forest.html”. This is the name you referenced in the metadata object of the mapfile (step 5).
- 8) This is the HTML code that will get returned by MapServer and displayed in the table of contents. Whenever you use square brackets [] mapserver will substitute that for the actual data source attribute value. These are **case sensitive**. See example below:

```
<!-- MapServer Template -->
<tr bgcolor="#DEE5EB"><td colspan="2"><b><u>County Forests</u></b></td>
<tr>
<td align="right"><b>DNR Region:</b></td>
<td>[DNR_REGN]</td>
</tr>
<tr>
<td align="right"><b>County:</b></td>
<td>[CTY_NAME]</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
```

- 9) Save the template and Map file and give it a try in GeoMOOSE.

GeoMOOSE Exercise 7: Finish

Exercise Title: Configuring the Identify Service

Lessons Learned:

- General Understanding of GeoMOOSE services
- Configuring new layers to be identified and return attribute results through MapServer query templating

Bing Bird's Eye and Google StreetView Services

Purpose: Provides a way for users to open an external window for the location they click in the GeoMOOSE map to Bing Maps and Google Maps.

User Interaction: User activates the tool and then clicks on the map.

Service Script Source: `birdseye.php` and `streetview.php`

Notes: This service takes the map x,y coordinate the user clicks converts it to Lat and Long. Then constructs a URL with those coordinates to open up the external map service. The way these services configured do not violate the Google or Bing API license terms.

Google Geocoder Service

Purpose: Provides a way for users to find and map an address.

User Interaction: User activates the tool, then types in address.

Service Script Source: geocode.php

Notes: This service takes address string and sends it to the Google Geocoder api. It then displays a popup for the location the address string geocodes to. The service can also geocode intersections and landmarks. Future enhancements to this service are planned and include: limiting the geocode extent to the max extent defined in the mapbook, giving the user back a friendly error message when the address string is not found by the Google geocoder. As of now it returns the location of the top candidate.

Cool stuff

- Firebug – add-on for Firefox great for troubleshooting (getfirebug.com)
- Fiddler – add-on for IE good for troubleshooting (www.fiddler2.com)
- IE Developer Toolbar – Microsoft CSS viewer (Google it)
- GeoMOOSE website – sign up on listserv (www.geomoose.org)
- MapServer website (www.mapserver.org)
- Colorbrewer – selecting symbol colors (colorbrewer2.org)
- ColorZilla – Color picker for FireFox (www.colorzilla.com/firefox/)
- Spatial Reference – get MapServer projections (www.spatialreference.org)
- www3schools – reference for HTML, XML, JavaScript (www.w3schools.com)
- AT File Mole – freeware to edit cursor files (www.screen thiefsoft.com/quicktourmole.html)
- Quantum GIS – open source desktop GIS (www.qgis.org)
- TextPad – Text editor (www.textpad.com)
- Amein! – ArcGIS plug-in to create MAP file (<http://chris.narx.net/2005/09/22/how-to-convert-mxd-to-map-using-amein>)