Getting Started with MapServer, Part 1

Jeff McKenna
Tyler Mitchell
Perry Nacionales

Outline

MapServer Background

- Brief History
- Features/Capabilities

MapServer Demo Application

- Compilation/Downloading
- Installation
- Demo Application

Third Party Client/Management Tools

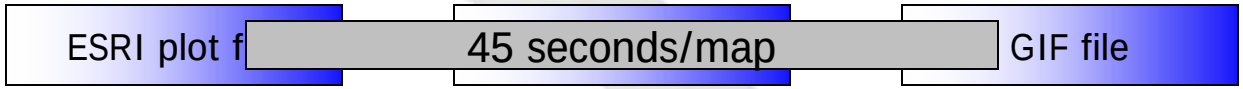
- Overview of Third Party Client/Management Tools

Background

A Brief History of MapServer
(or MapServer's Brief History

MapServer: The Past

- 1994 - C Program Creates ArcPlot AMLs



- 1994 - NASA-sponsored ForNet Project
- 1997 - S*hapelib L*ibrary (LGPL or “MIT-type”)
- 1997 - NASA-sponsored TerraSIP Project
- 1998 - MapServer 3.2 Released as Open Source
- 2001 - MapServer 3.4
- 2003 - MapServer 4.0
- 2007 - MapServer 5.0

MapServer: The Present

- Version 5.0 Released September, 2007
- 10+ Active Developers Around The World
- 2,100+ Subscribers to MapServer-Users List
- **40,000**+ Global Applications Deployed
- FOSS4G 2007

MapServer: The Future

- Open Source Community
- Open Source Geospatial Foundation
  - https://www.osgeo.org/
- OGC Compliant

Background

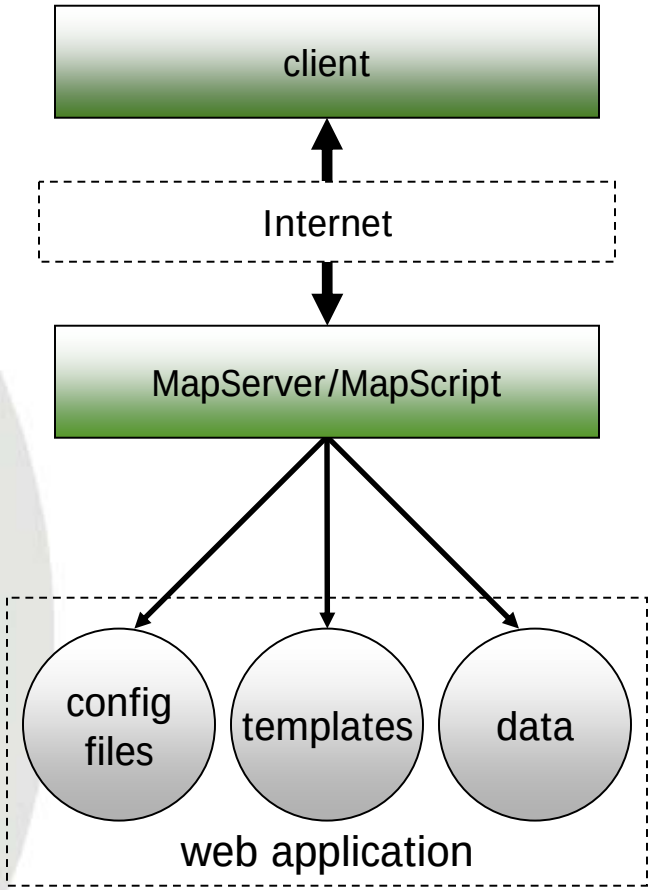
MapServer Features and Capabilities

What isn't MapServer

- A Desktop GIS!
- A Ready-To-Use Application (e.g. parcel mapping)
- Client-Based (e.g. isn't installed in your computer)
- Easy-To-Use, "Wizard" Configurable
- Expensive!

What is MapServer?

- Application Development System for Web-Based Mapping
- Server-Based
- Open Source
- Supports OGC Web Services Specifications
- *Fast*
- Extremely Configurable
- Easy-To-Use



Creating A MapServer Application

The Way to Web Mapping Nirvana

Required and Optional Software

Any HTTP server that supports the CGI protocol

- Apache, IIS, Tomcat, etc.

MapServer CGI program and utilities

- shp2img, shptree, sortshp, etc.

Other geospatial programs/utilities

- GDAL/OGR utilities (gdalinfo, ogrinfo, ogr2ogr, etc.)
- Proj.4 utilities (proj, cs2cs)
- GRASS, QGIS, etc.

Downloading and Compiling

- Download the Source Code
  - http://mapserver.gis.umn.edu/download
- Compile the Source Code
  - See http://mapserver.gis.umn.edu/docs/howto/compiling_on_unix
  - See http://mapserver.gis.umn.edu/docs/howto/win32_compiling
- Download Pre-compiled Binary Packages
  - If compiling isn't an option
  - MS4W for Windows, FGS for Linux, FWTools
  - Mac OS X
  - Debian and other Linux Distributions

Installating MapServer (Packages)

- For MS4W (windows):
  - Make sure you have no web server installed/running
  - Unzip the MS4W archive to the C: drive (or any drive)
  - Using Windows Explorer (not Internet Explorer), navigate to C:\ms4w and double-click on README_INSTALL.html
  - If you are averse to reading READMEs (bad!), double-click on apache_install.bat
- For stand-alone binaries on Windows:
  - Unzip archive in a temporary directory
  - Copy mapserv.exe to web server's CGI-BIN directory
  - Copy DLL's to system directory or to CGI-BIN directory
  - Copy Proj.4 directory to C:\

Installating MapServer (Packages)

- For FGS (Linux):
  - See http://www.maptools.org/fgs/index.phtml?page=install.html and http://www.maptools.org/fgs/index.phtml?page=readme.html
- For FWTools:
  - Follow directions at http://fwtools.maptools.org/windows-main.html or at http://fwtools.maptools.org/linux-main.html
- For more platform specific instructions:
  - Go to http://mapserver.gis.umn.edu/docs/howto and look for documents under the “Compiling” section

Testing Your MapServer Installation

- Open a web browser window and type: http://localhost/cgi-bin/mapserv (Linux/Unix/Mac) or http://localhost/cgi-bin/mapserv.exe (Windows)
- This should return the following message:

```
No query information to decode. QUERY_STRING is set, but empty.
```

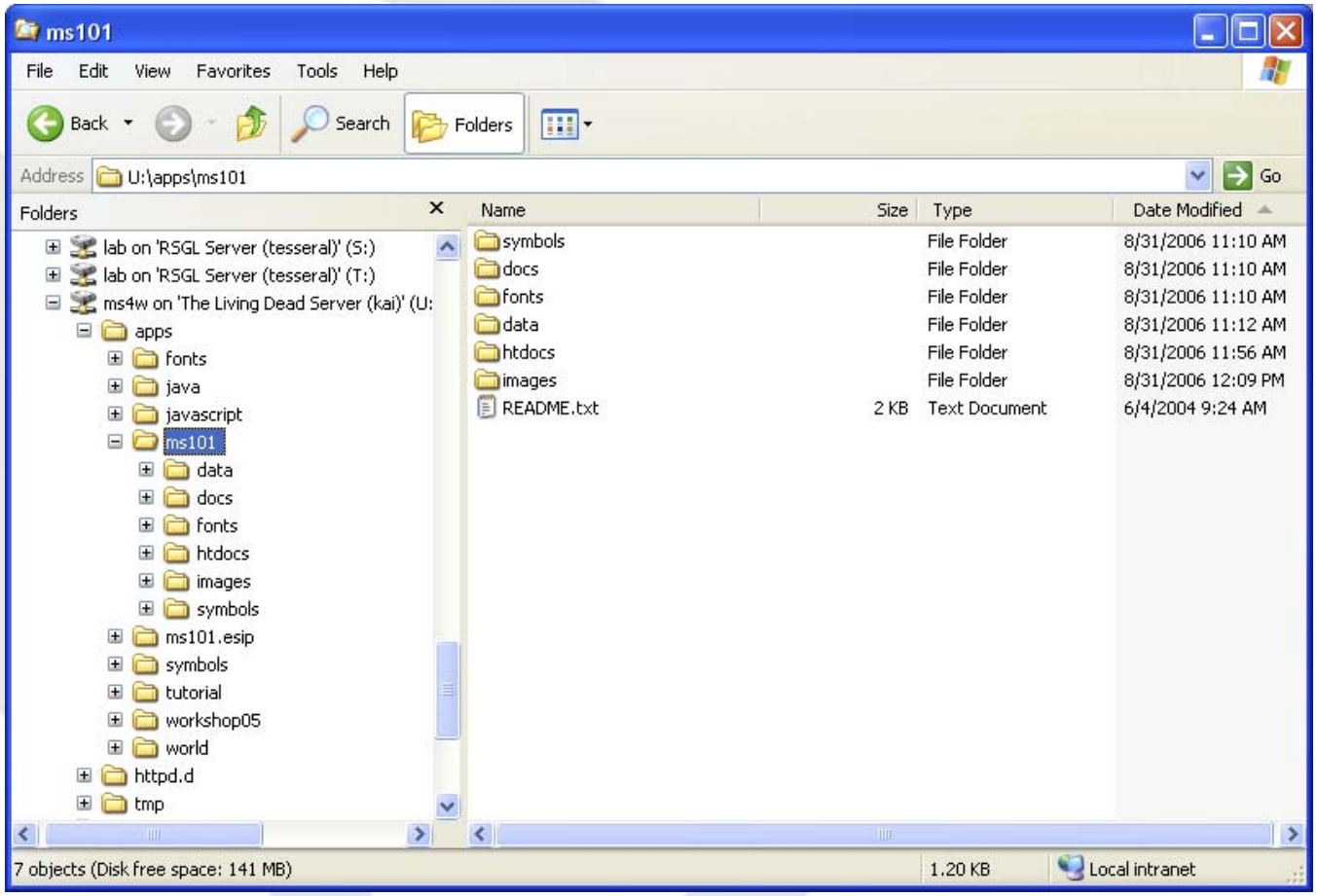
- If you didn't get the message above, check your installation and make sure that:
  - The web server has been restarted
  - All the required libraries are installed either in the system directory (e.g. C:\Windows\system32) or in the CGI-BIN directory (e.g. C:\Apache\cgi-bin)
  - The MapServer CGI program (mapserv or mapserv.exe) is in the CGI-BIN directory

Application Files and Directories

- Application Directory: `/data/www/ms101`
  - Map and HTML template file are located here
  - Demo Application template file: `ms101final.html`
  - Demo Application map file: `ms101.map`
- Web Root Directory: `/data/www/ms101/htdocs`
  - HTML files (index.html) and web-readable images
  - Web Root Alias (Virtual Directory): /ms101
  - Application URL: http://localhost/ms101/index.html
- Temporary Files Directory: `/ms4w/tmp/ms_tmp`
  - Files created by MapServer goes here
  - Alias: /ms_tmp
- MapServer CGI Program: `/cgi-bin/mapserv`(`.exe`)

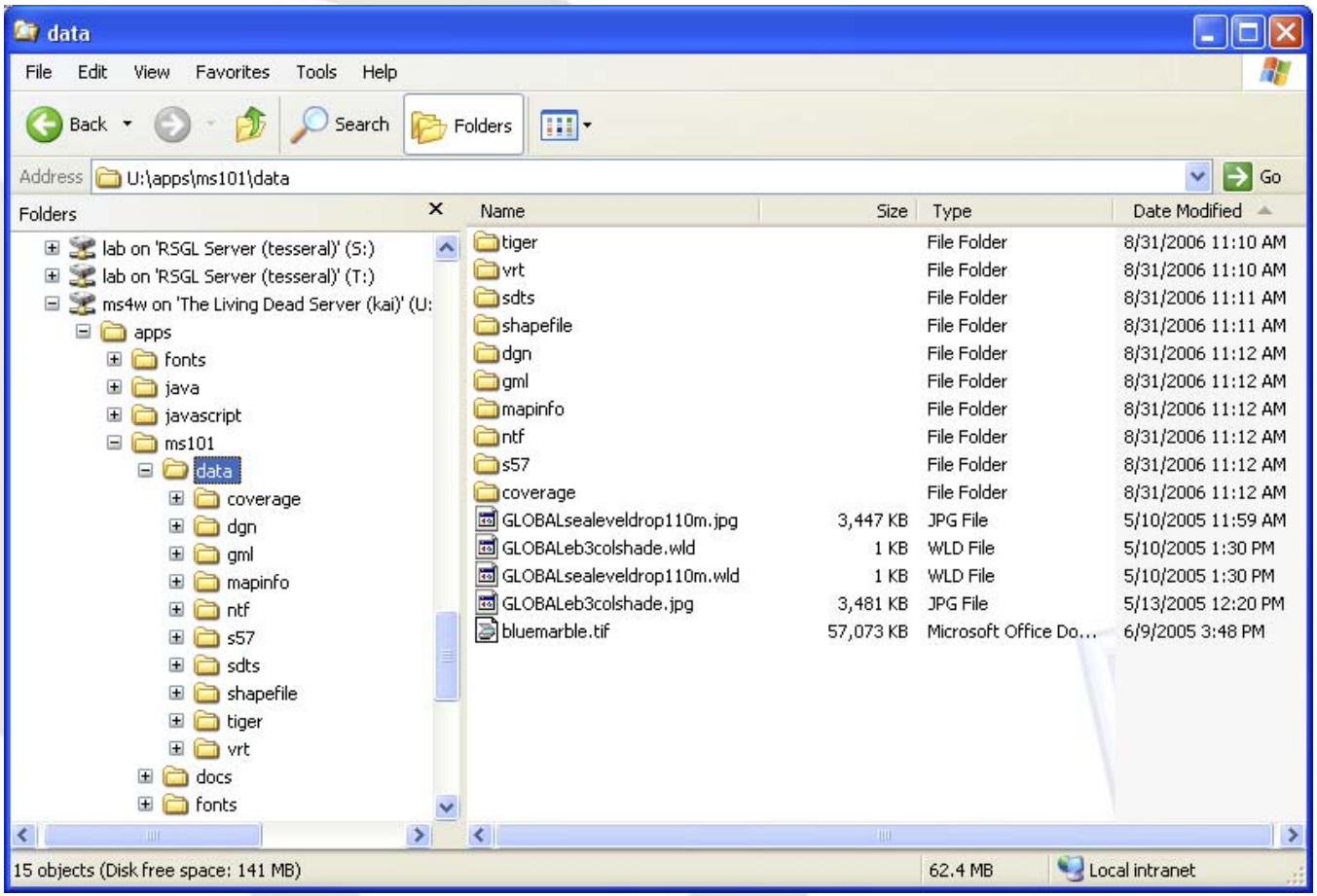
Creating a Demo Application

- Demo application directory structure



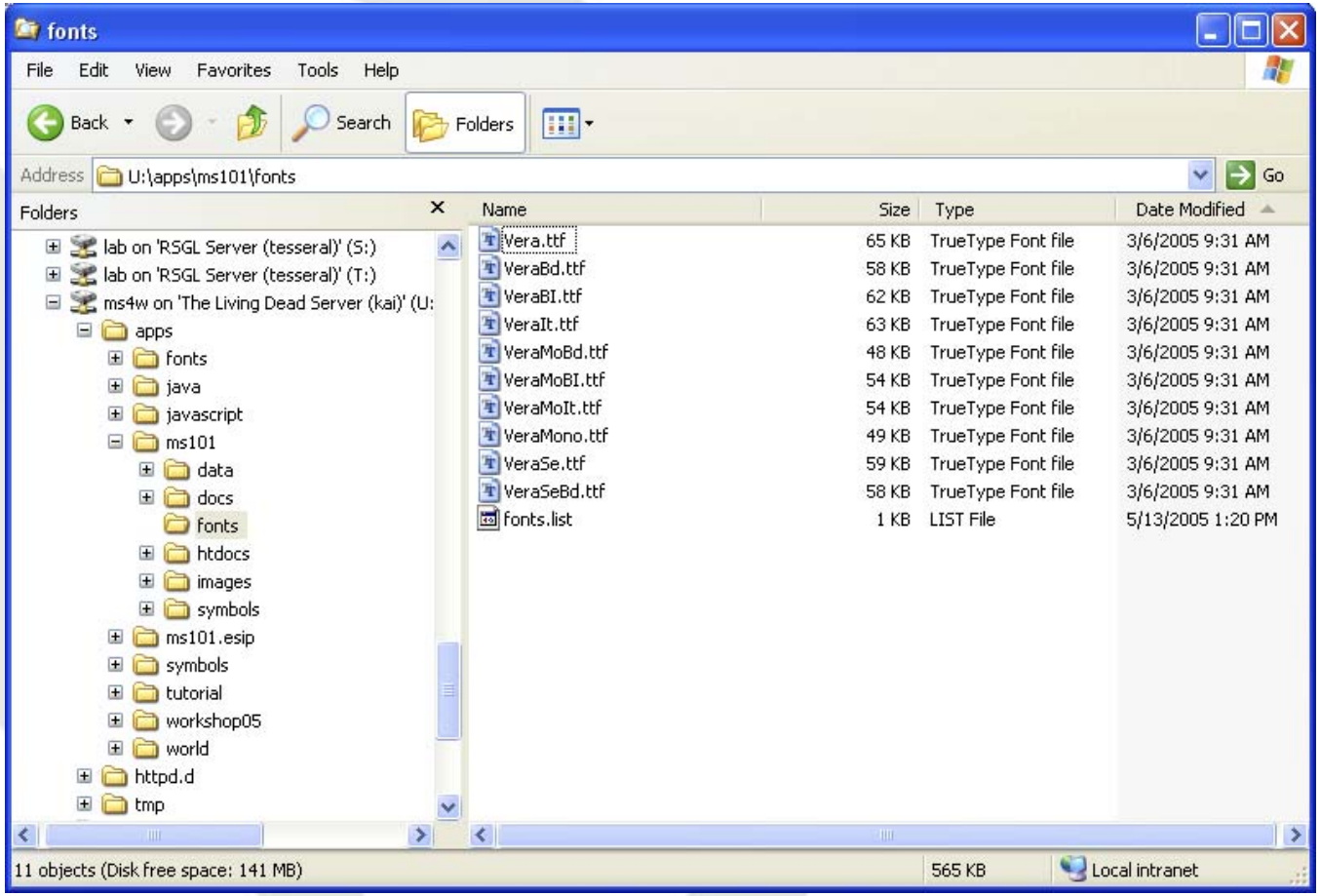
Creating a Demo Application

- All data are in “data” subdirectory



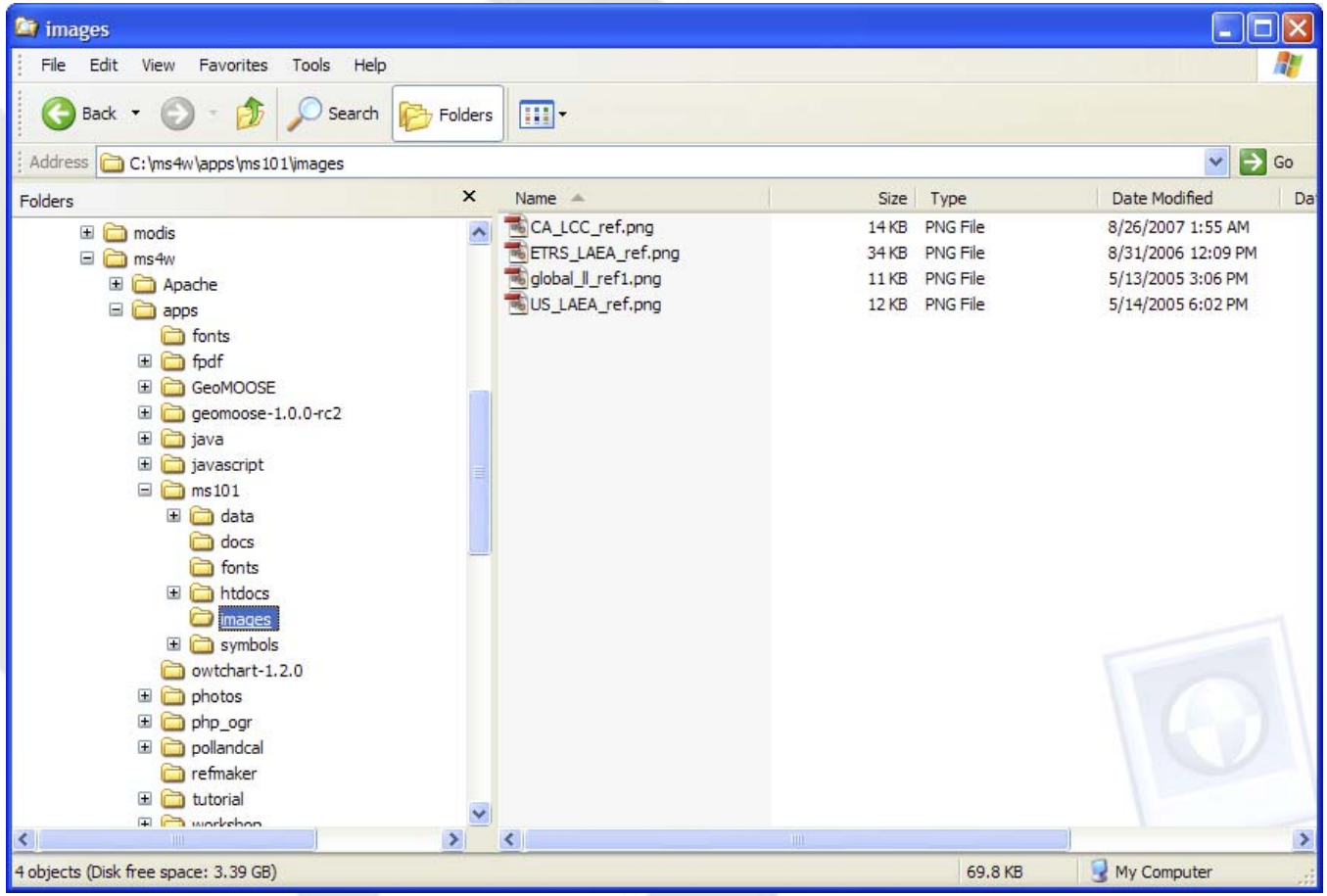
Creating a Demo Application

- TrueType fonts are in “fonts” subdirectory



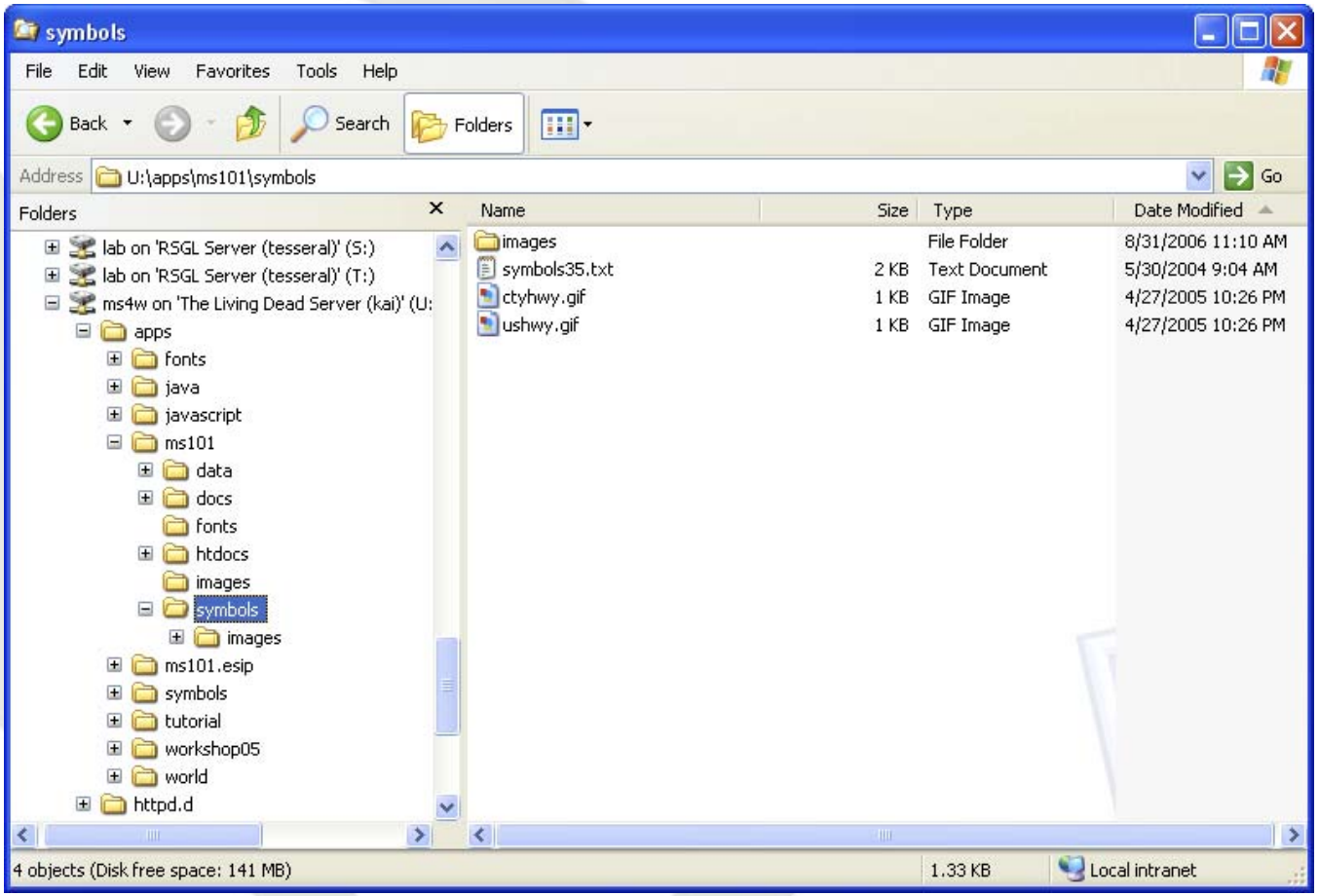
Creating a Demo Application

- Template images are under the “images” subdirectory



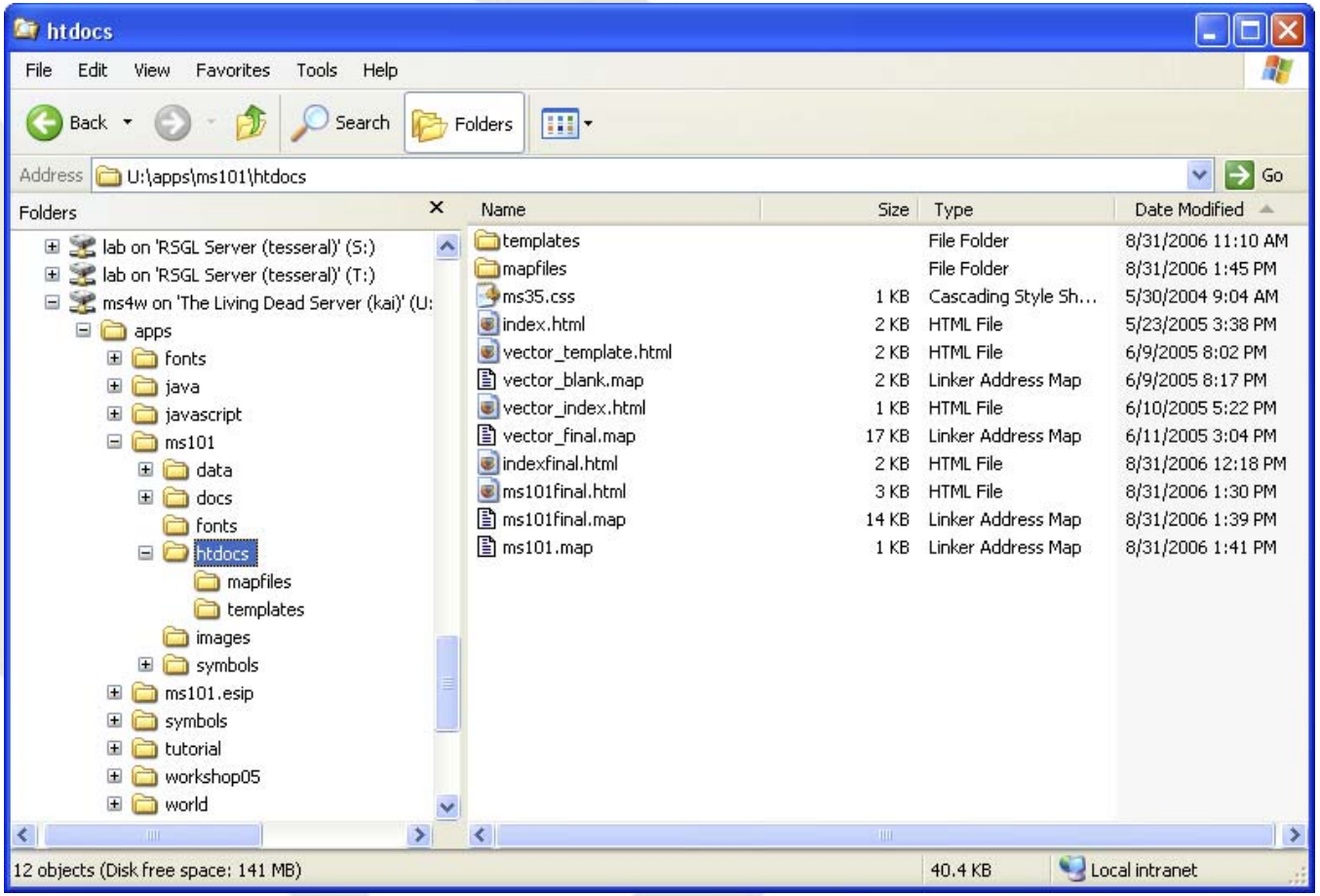
Creating a Demo Application

- Symbol files are under the “symbols” subdirectory



Creating a Demo Application

- Map files, template files, and other web-readable files are in the “htdocs” subdirectory



Creating a Demo Application

- Define a web “alias” (virtual directory) in your web server
- The demo's alias looks like this in Apache

```
Alias /ms101 "/ms4w/apps/ms101/htdocs"
<Directory "/ms4w/apps/ms101/htdocs">
    AllowOverride None
    Options Indexes MultiViews
    Order allow,deny
    Allow from all
</Directory>
```

- This definition needs to go into Apache's httpd.conf
  - You can also save it as a file and “Include” it in httpd.conf – see /ms4w/Apache/conf/httpd.conf
- This Alias tells the web server where the url http://localhost/ms101 is located.

Example 1: The Map Object

- Open "/ms4w/apps/ms101/htdocs/ms101.map" in your favorite text editor

```
MAP
  NAME            MS101_
  EXTENT         -180 -90 180 90 # Geographic
  SIZE            800 400
  IMAGECOLOR      128 128 255
END
```

- "EXTENT" is the output extent in output units
- SIZE is the width and height of the web map in pixels
- IMAGECOLOR is the default image background color
- The entire MAP FILE REFERENCE is available at:
  http://mapserver.gis.umn.edu/docs/mapfile_reference
- To see the example map, open the URL:
  http://localhost/cgi-bin/mapserv?map=/path/to/mapfile.map&mode=map

Example 1: The MapServer Request

- The request protocol: http://localhost/cgi-bin/mapserv.exe?map=/path/to/ms101.map&mode=map
- So, what's up with the output?
  - Remember the map file?
    - SIZE 800 400
    - IMAGECOLOR 128 128 255

Example 2: Adding a Polygon Layer

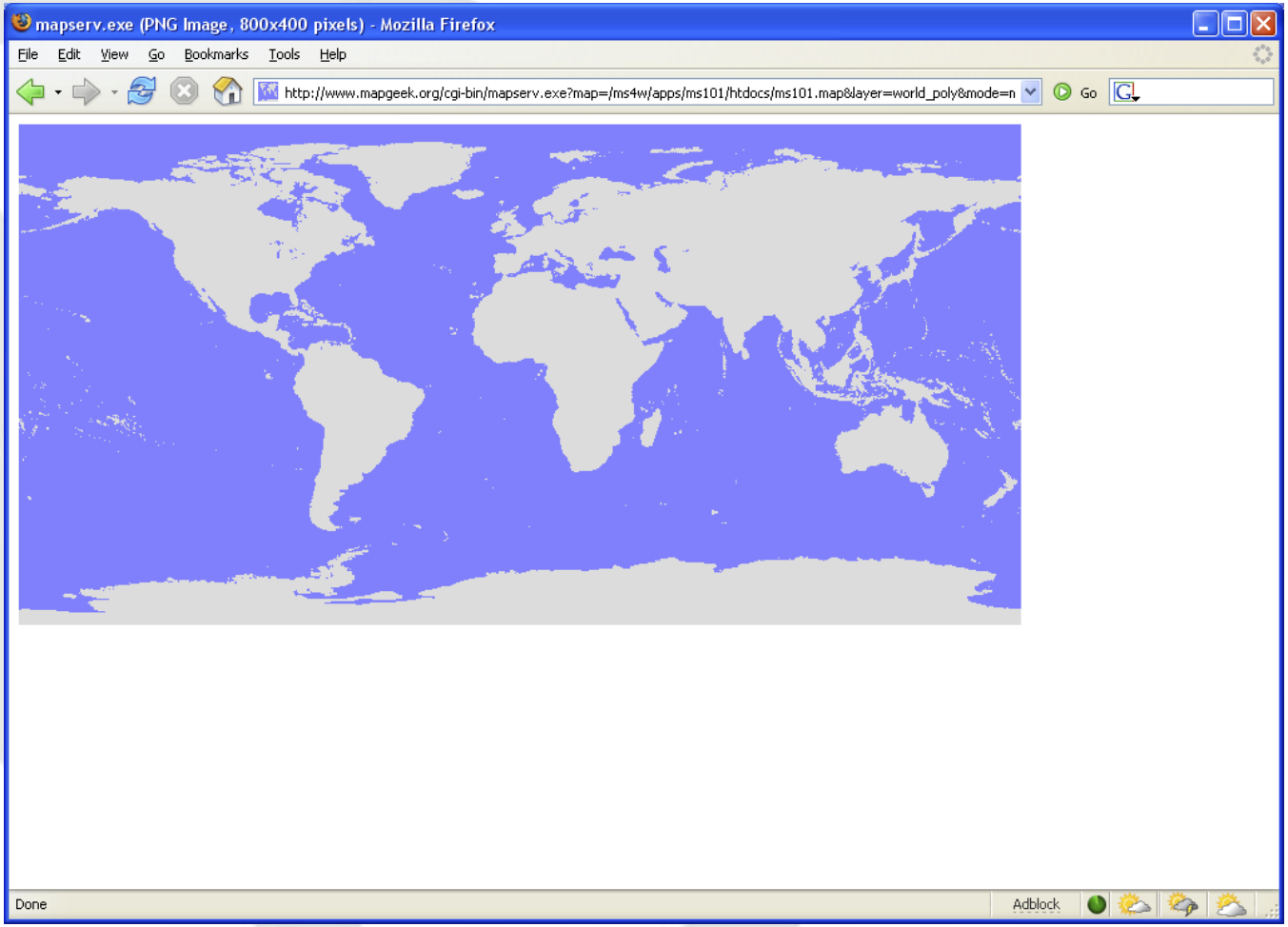
```
LAYER # World polygon layer begins here
  NAME          world_poly
  GROUP         world
  DATA          'shapefile/Countries_area.shp'
  STATUS        ON
  TYPE          POLYGON

  CLASS
    NAME        'The World'
    STYLE
      COLOR     220 220 220
    END
  END
END # World polygon layer ends here
```

- Open “mapfiles/world_poly_layer.map” and append its contents to “ms101.map”, just before the “END” keyword
- View the resulting map with the following request:

http://.../cgi-bin/mapserv.exe?map=/.../ms101.map&layer=world_poly&mode=map

Example 2 Output



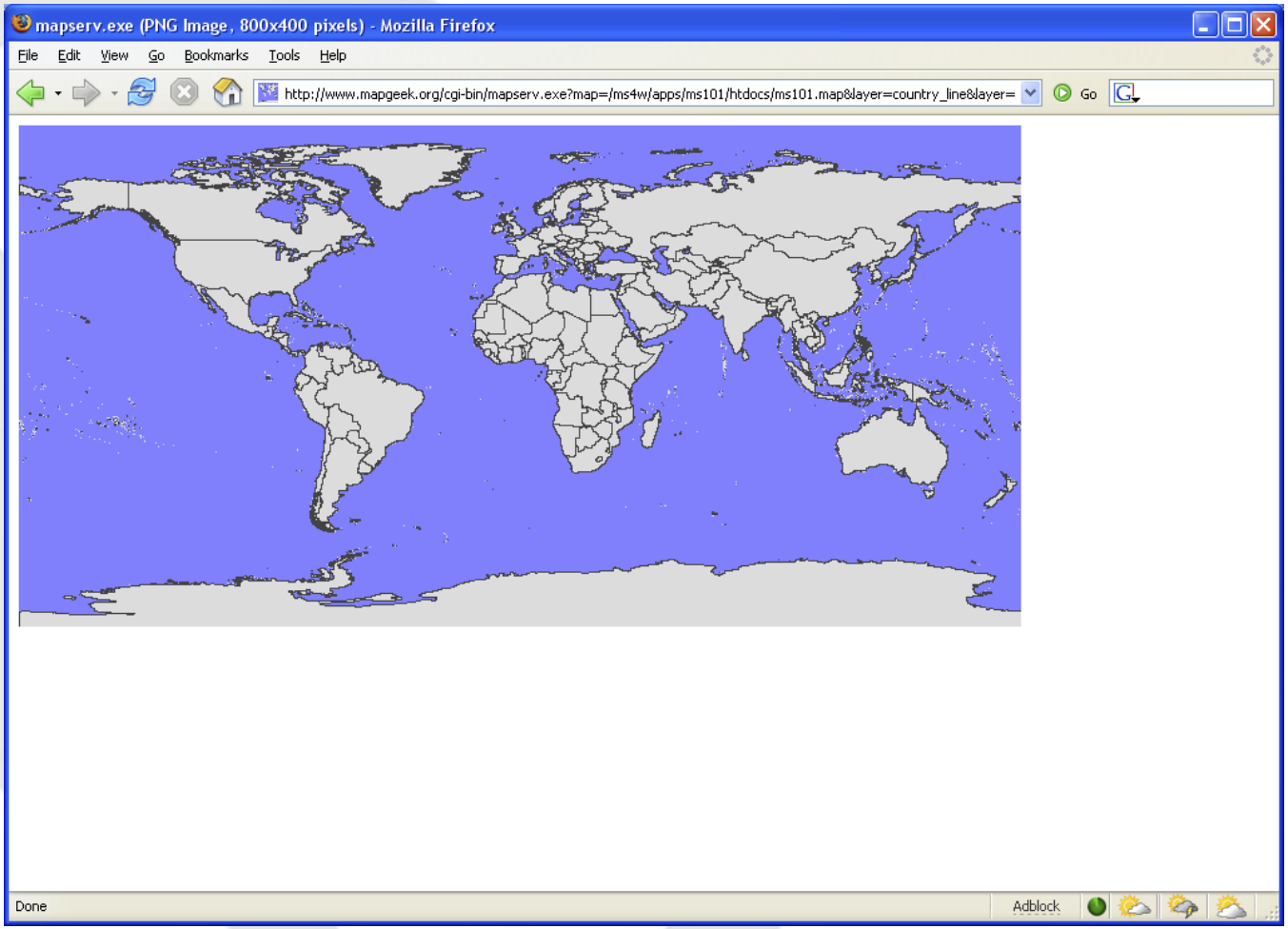
Example 3: Adding a Line Layer

```
NAME          country_line
GROUP         world
DATA          'shapefile/Countries_area'
STATUS        ON
TYPE          LINE

CLASS
  NAME        'Country Boundary'
  STYLE
    COLOR     64 64 64
  END
END
```

- Open “mapfiles/world_line_layer.map” and append its content to “ms101.map” just below the first layer and before the “END” keyword
- Again, view the map by typing in the MapServer request

Example 3 Output



Example 4: Defining Layer Classes

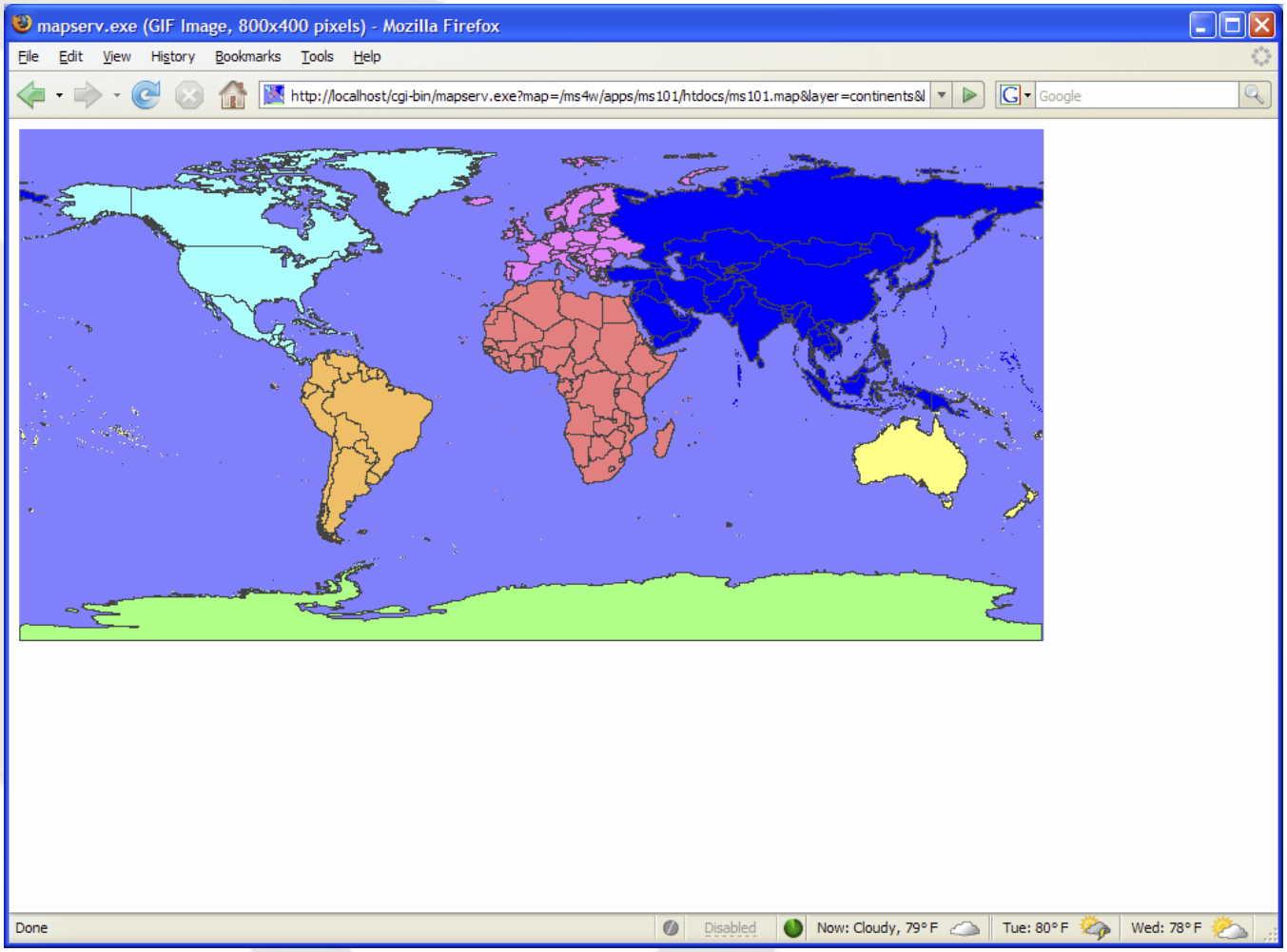
- Open the file “mapfiles/continents_layer.map”
- Here's a snippet of the file:

```
LAYER # World polygon classified by continents begins here
  NAME          continents
  DATA          'shapefile/Countries_area'
  STATUS        ON
  TYPE          POLYGON

  CLASSITEM     'NA3DESC'
  CLASS
    NAME        'Africa'
    EXPRESSION 'Africa'
    STYLE
      COLOR     255 128 128
    END
  END
  ...
```

- Add it to “ms101.map”, below the first layer.
- Make another request to view the map.

Example 4 Output



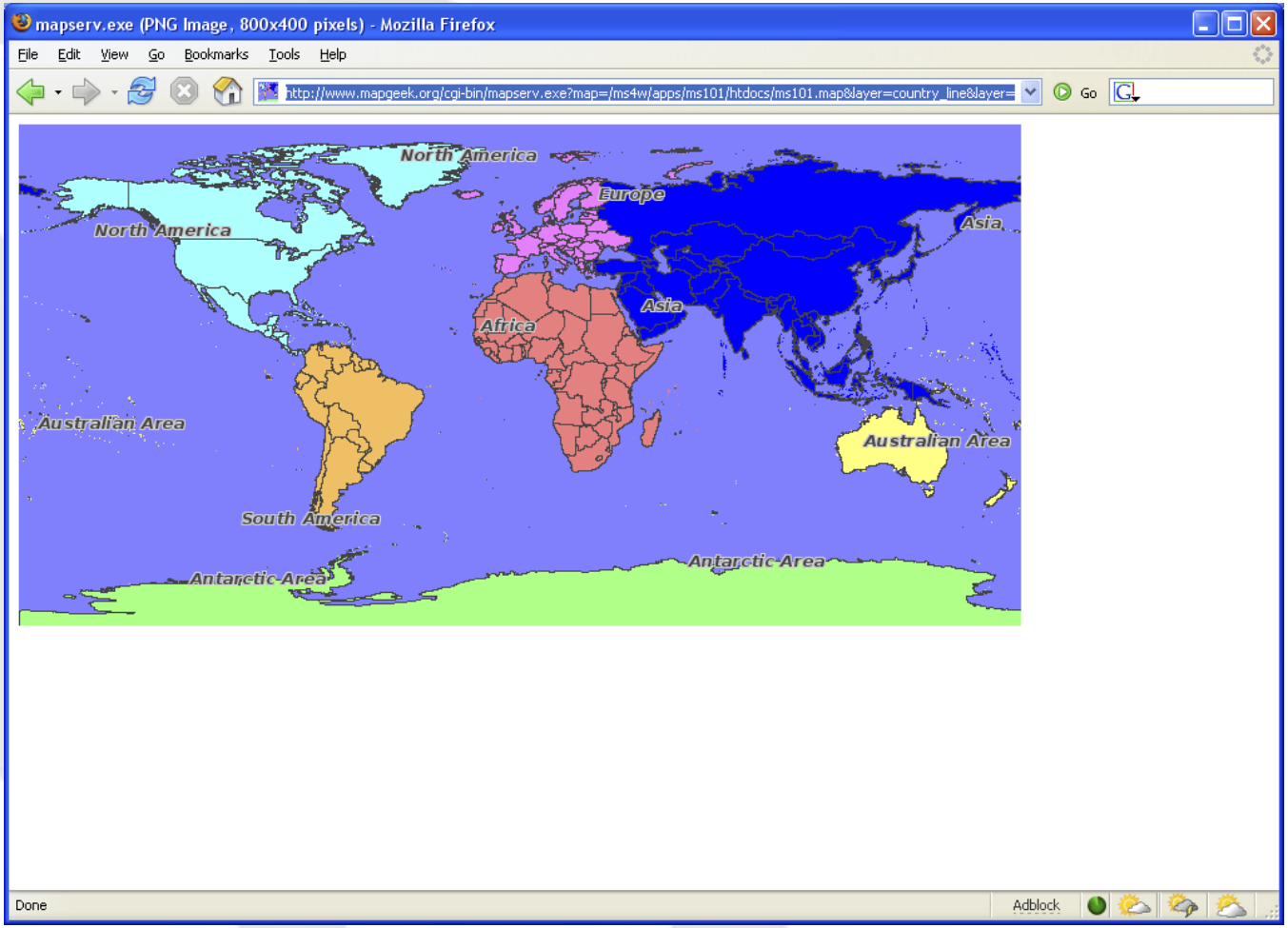
Example 5: Labeling

- Open the file “mapfiles/continents_label_layer.map”
- Here's a part of the file:

```
LAYER # World polygon classified by continents begins here
  ...
    LABEL
      COLOR    64 64 64
      OUTLINECOLOR 212 212 212
      TYPE     TRUETYPE
      FONT     vera_sans-bold-italic
      SIZE     10
      ANTIALIAS TRUE
      POSITION CC
      PARTIALS FALSE
      MINDISTANCE 250
      BUFFER   4
    END
  ...
```

- Replace the continents layer in “ms101.map” with this file.
- Make another request to view the map.

Example 5 Output



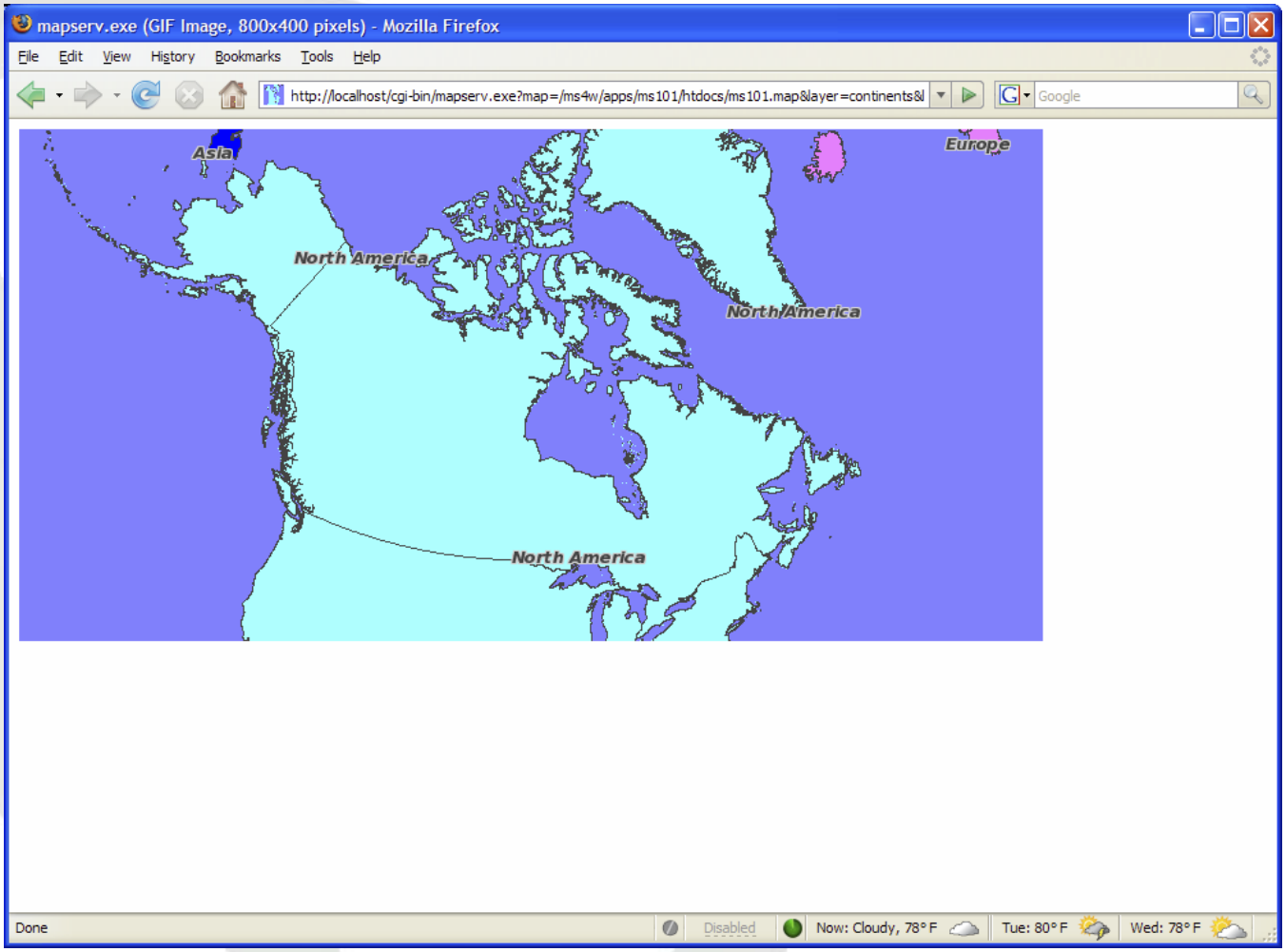
Example 6: Reprojection

- Open the file “mapfiles/output_projection_block.map”
- Here's a part of the file:

```
...
  PROJECTION
    ...
    "init=epsg:42304"
    ...
  END
...
```

- Append the contents of this file to “ms101.map”, just before the first layer definition.
- Open the file “mapfiles/layer_projection_block.map” and copy its content to each of the layers defined in ms101.
- Replace the map EXTENT with the content of “mapfile/ca_lcc_extent.map”
- Make another request to view the map.

Example 6 Output



Example 7: Adding Raster Data

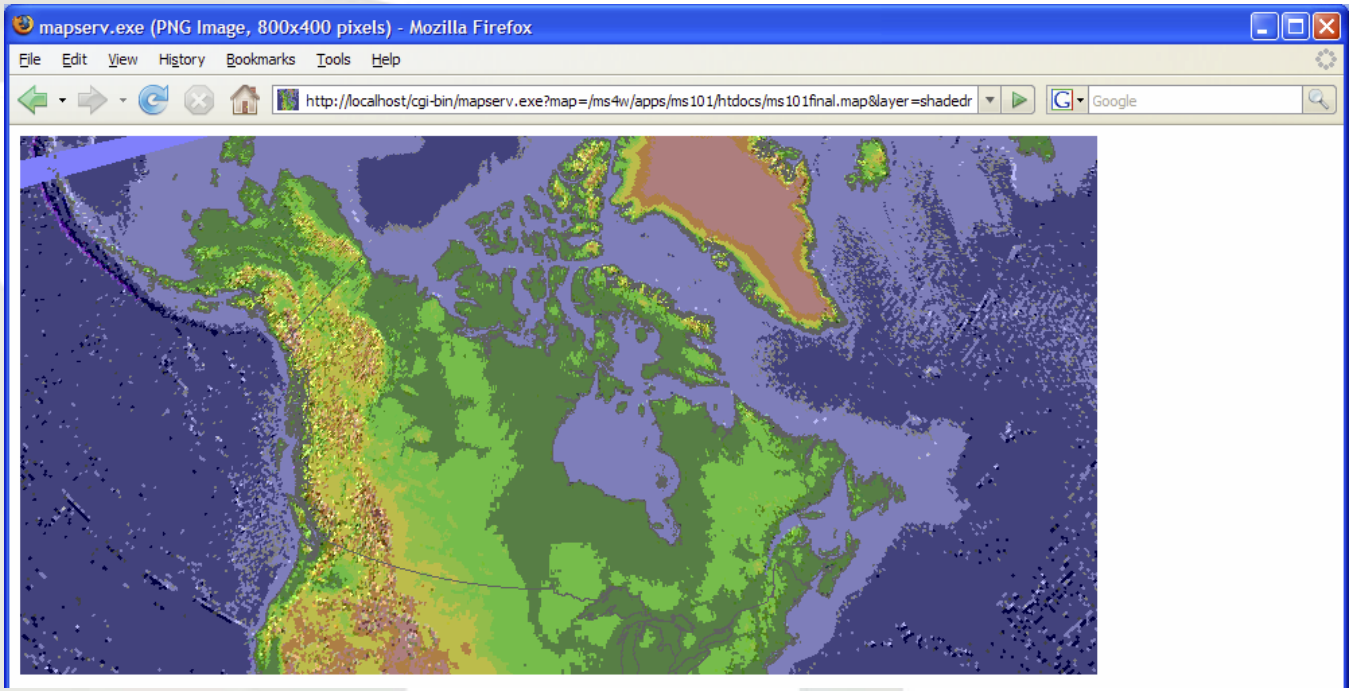
- Open “mapfiles/shadedrelief_raster_layer.map”

```
LAYER # Shaded relief raster layer ends here
  NAME          'shadedrelief'
  STATUS        ON
  TYPE          RASTER
  DATA          'GLOBALeb3colshade.jpg'

  PROJECTION
    "init=epsg:4326"
  END
END # Shaded relief raster layer ends here
```

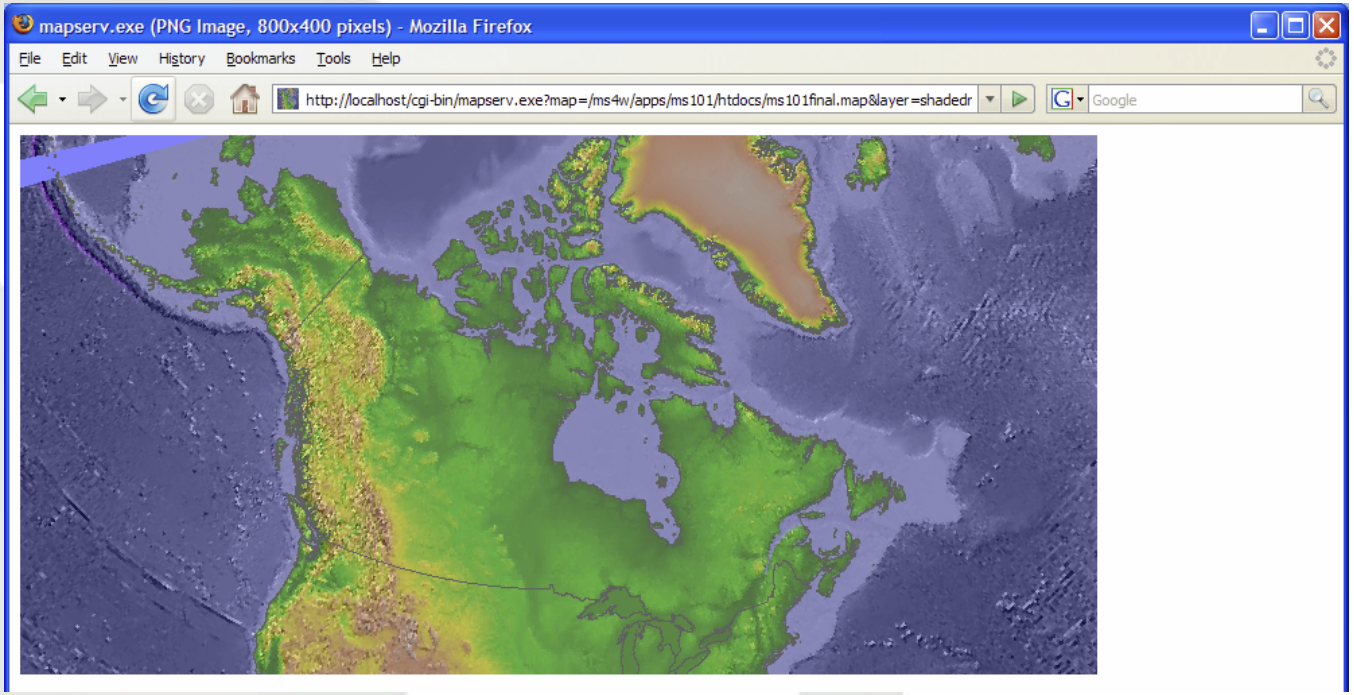
- Append the contents of this file to “ms101.map”, after the two polygon layers—world_poly and continents
- View The Map
- Raster Data Access for more information

Example 7 Output



- Ugly?
- APPEND the following line just before the projection object: `IMAGETYPE PNG24`
- Refresh the browser

Example 7 Output, Reloaded



- You just changed MapServer's default output image from 8-bit PNG to truecolor PNG (24-bit, RGB).
- By doing this, you also increased the file size of the output image—this will result in slightly longer rendering times and higher CPU usage.

Example 8: Adding OGC WMS Layer

- Open “mapfiles/wms_client_modis_layer.map”

```
LAYER # MODIS WMS map from JPL
  ...
  CONNECTIONTYPE WMS
  CONNECTION "http://wms.jpl.nasa.gov/wms.cgi?"
  METADATA
    "wms_srs" "EPSG:4326"
    "wms_name" "BMNG"
    "wms_server_version" "1.1.1"
    "wms_format" "image/jpeg"
    "wms_style" "Aug"
  END
  ...
END # Modis WMS image ends here
```

- Append the contents of this file to “ms101.map”, after the “shadedrelief” layer

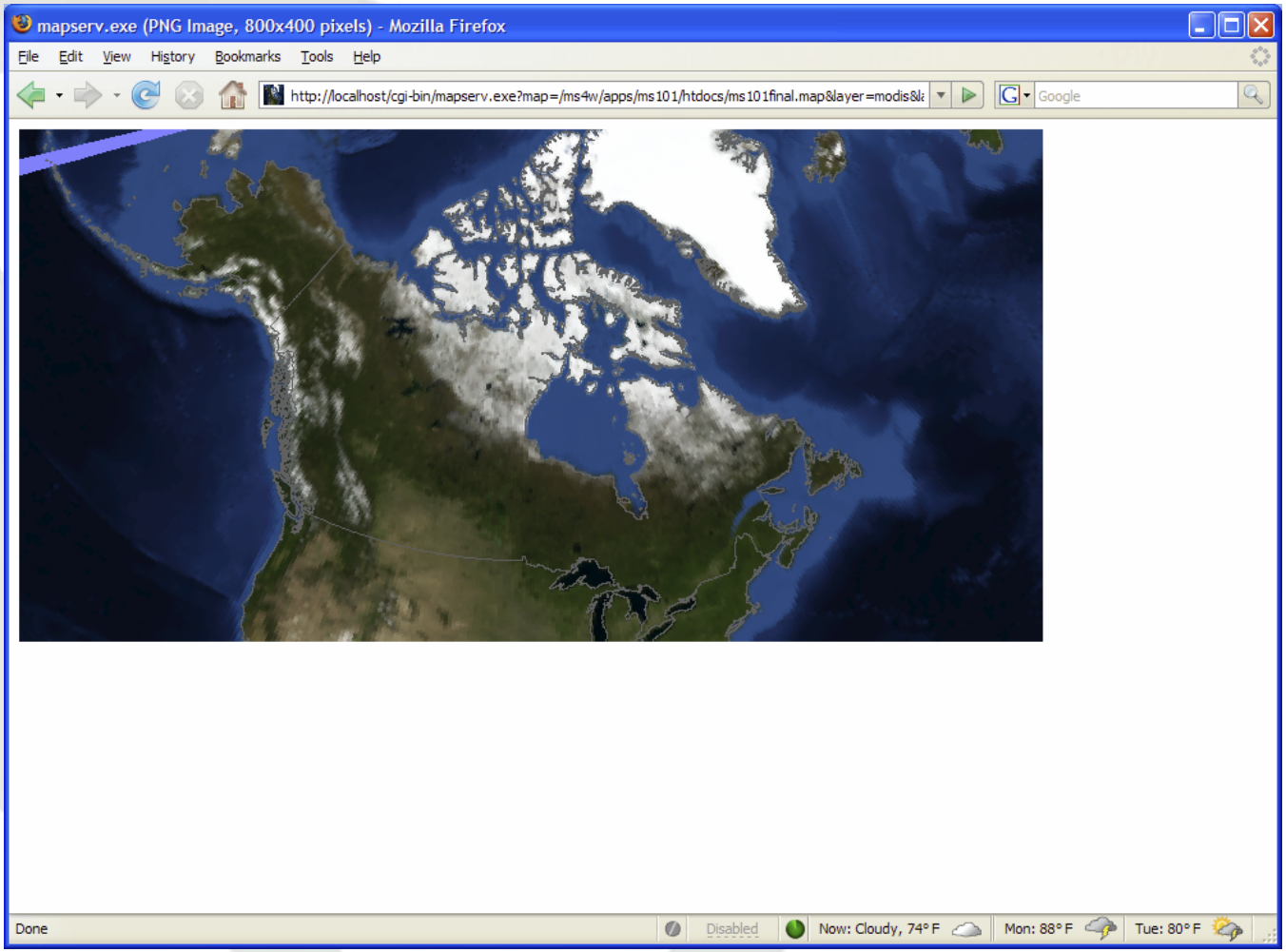
Example 8: Adding OGC WMS Layer

- Open “mapfiles/web_object_block.map”

```
WEB
  ...
  IMAGEPATH     '/ms4w/tmp/ms_tmp/'
  IMAGEURL      '/ms_tmp/'
  ...
END
```

- Append the contents of this file to “ms101.map”, before the output projection block
- View The Map

Example 8 Output



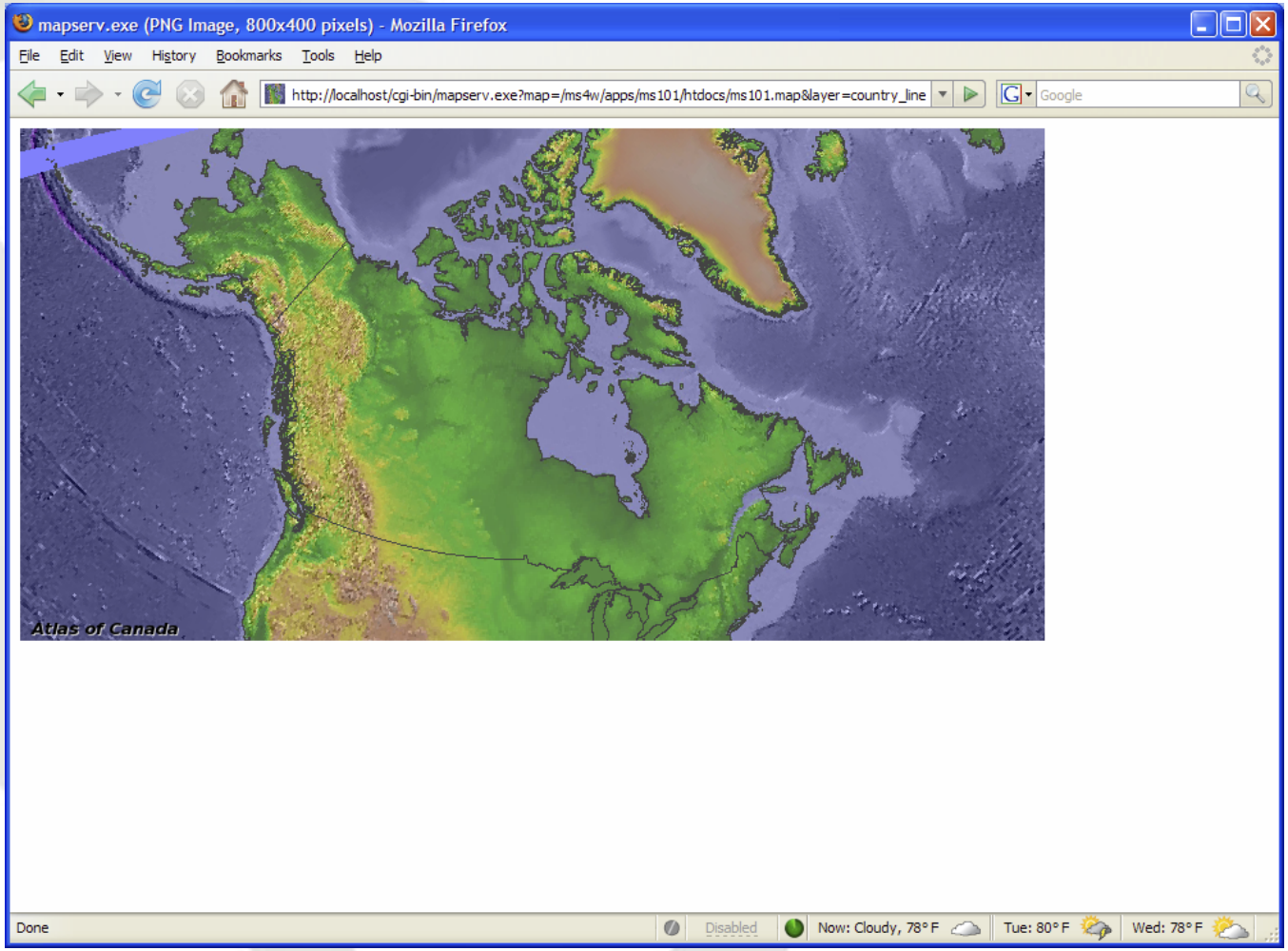
Example 9: Adding Inline Feature

- Open “mapfiles/inline_feature_layer.map”

```
...
FEATURE
  POINTS
    5 390
  END
  TEXT 'Atlas of Canada'
END
...
```

- Append the contents of this file to “ms101.map”, after the last layer and before the last “`END`” keyword
- View The Map

Example 9 Output



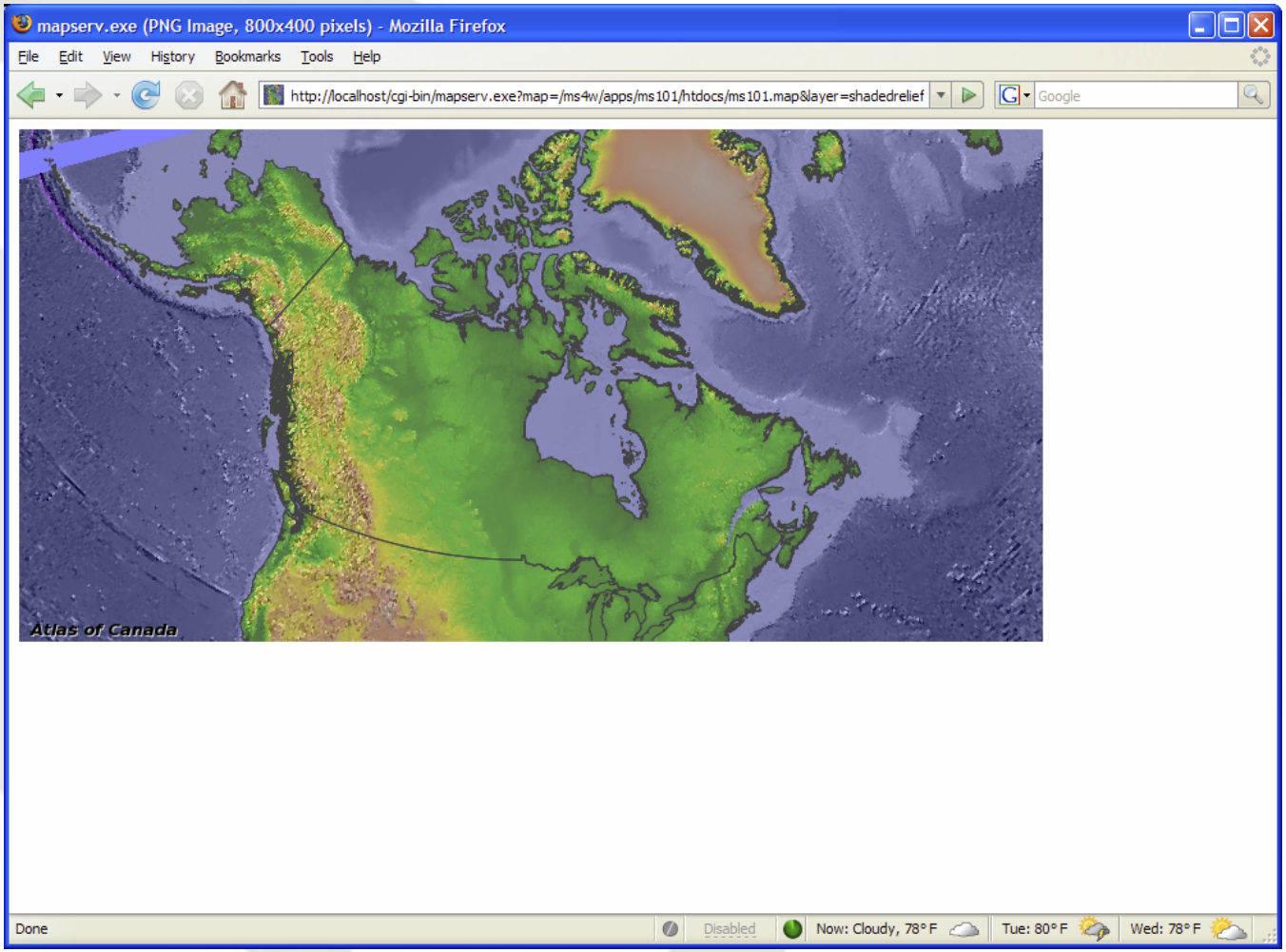
Example 10: Antialiasing

- Open “mapfiles/agg_outputformat.map”

```
...
OUTPUTFORMAT
  NAME 'AGG_PNG‘
  DRIVER AGG/PNG
  IMAGEMODE RGB
END
...
```

- Append the contents of this file to “ms101.map”, above the projection object
- Change the value of the `IMAGETYPE` keyword to `‘AGG_PNG’`
- View The Map
- The effects of this change will become more apparent once you are able to zoom-in on your map.

Example 10 Output



Example 11: SCALEBAR

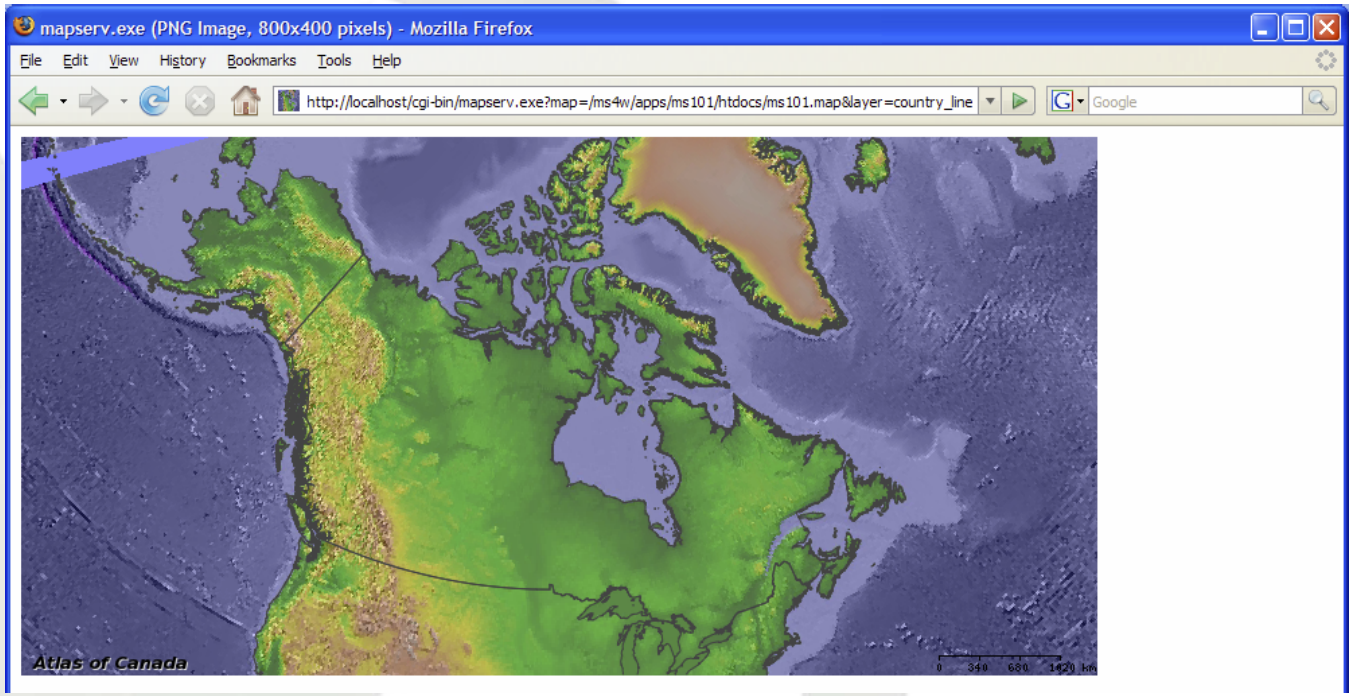
- Open “mapfiles/scalebar_object_block.map”

```
SCALEBAR
  IMAGECOLOR   255 255 255
  LABEL
    COLOR      0 0 0
    SIZE       TINY
  END
  STYLE        1
  SIZE         167 3
  COLOR        0 0 0
  UNITS        KILOMETERS
  INTERVALS    3
  TRANSPARENT  TRUE
  STATUS       EMBED
  POSITION     LR
END
```

- Append the contents of this file to “ms101.map”, after the output projection block
- View The Map or just the scalebar itself

Example 11 Output

- Embedded Scalebar



- The Scalebar image via “mode=scalebar”

Example 12: LEGEND

- Open “mapfiles/legend_object_block.map”

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

- Append the contents of this file to “ms101.map”, after the scalebar block
- View The Legend

Example 12 Output

- The Legend graphic:

Example 13: REFERENCE

- Open “mapfiles/reference_object_block.map”

```
REFERENCE
  STATUS       ON
  IMAGE        '../images/CA_LCC_ref.png'
  SIZE         180 90
  EXTENT               -4508654 -712571.045113 4622390.090226
                            3847236.954887 # CANADA LCC
  COLOR        -1 -1 -1
  OUTLINECOLOR 255 0 0
END
```

- Append the contents of this file to “ms101.map”, after the legend block
- View The Legend:

Example 14: HTML Template

- The previous examples dealt exclusively with mapfile objects and how the affect the output map (or legend, reference, scalebar)
- The mapfile you have done is practically complete and can be used in many client interfaces (like OpenLayers or ka-Map!)
- The remaining examples deal with creating a custom HTML template that uses the mapfile you just created
- HTML templates provide an interface that allows MapServer maps to be viewed (and interacted) on your web browser.

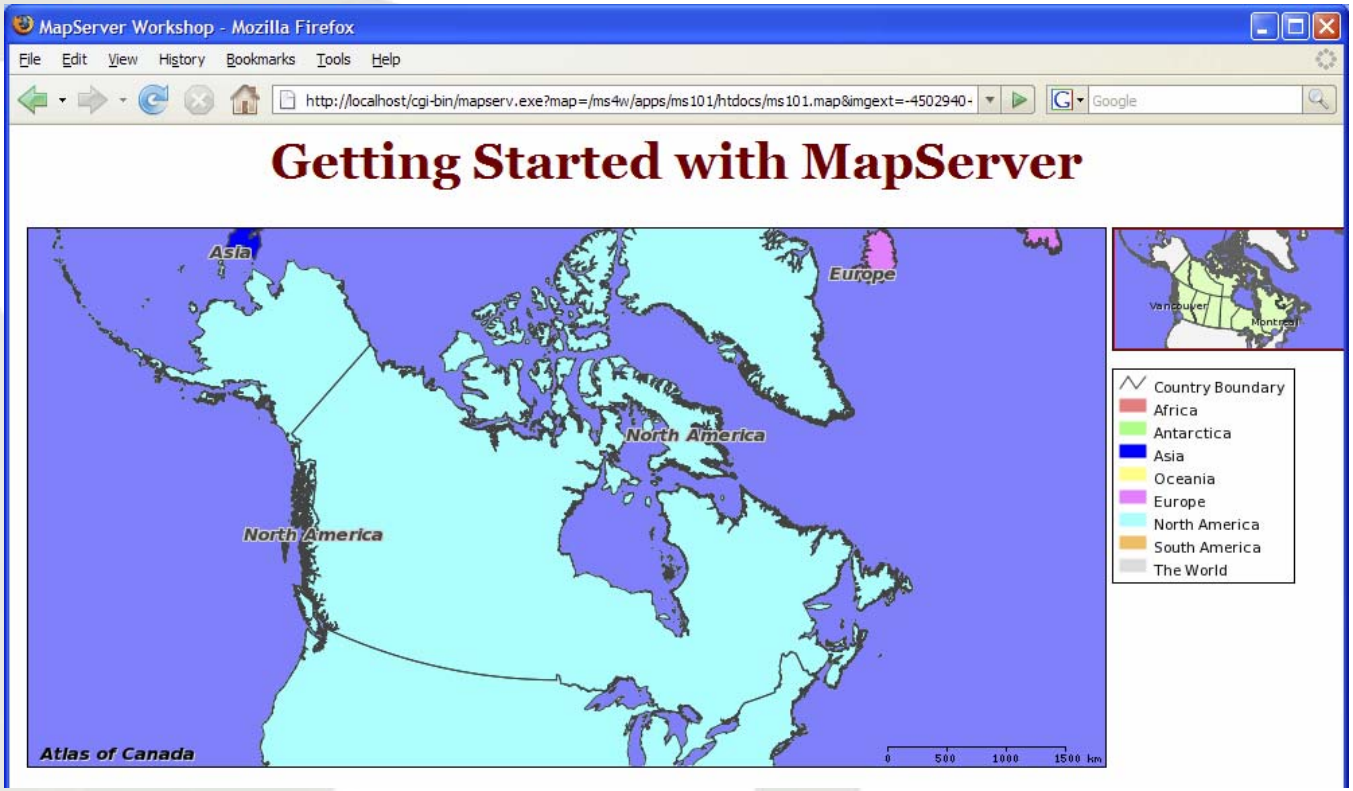
Example 14: HTML Template

- Open “templates/ms101.html”

```
<html>
  ...
  <body bgcolor="#FFFFFF" text="#000000">
    <h1 align="center">Getting Started with MapServer</h1>
    <div align="center">
      <img src="[img]" align="center" border="1" />
    </div>
  </body>
</html>
```

- Create a table within this file and add image tags for legend and reference:
  - Reference tag is [ref]
  - Legend tag is [legend]
- View The Map

Example 14 Output



- While not interactive, this example shows you simple way to create an HTML template and make use of MapServer CGI variables
- To see an interactive HTML template example, have a look at "ms101final.html"

Example 15: QUERY

- Open “mapfiles/world_query_layer.map”

```
LAYER # World query layer begins here
  ...
  STATUS       ON
  TYPE         QUERY
  ...
  HEADER       'templates/countries_header.html'
  FOOTER       'templates/countries_footer.html'
  TOLERANCE    2
  CLASS
    TEMPLATE   'templates/countries_query.html'
    ...
  END
END # World query layer ends here
```

- Append this file to “ms101.map”
- In “ms101.map”, replace web object template keyword to:
  `TEMPLATE 'ms101final.html'`
- View the map, select a “query” mode, click on a country

Example 15: QUERY, Continued

- To see the HTML template query files used in this example…

- Open “templates/header.html”

```
<html>
  <head>
    <title>MapServer Demo Interface</title>
    <link type="text/css" rel="stylesheet"
  href="/ms101/ms35.css" />
  </head>
  <body bgcolor=#FFFFFF>
    <h1 align="center">MapServer Query Interface</h1>
```

- Open “templates/footer.html”

```
  </body>
</html>
```

Example 15: QUERY, Continued

- Open “templates/countries_header.html”

```
<h4>
  <b>Layer: Countries</b>
</h4>
<table cellpadding=5 cellspacing=2 border=0>
  <tr bgcolor=#CCCCCC>
    <th>COUNTRY NAME</th>
    <th>CODE</th>
    <th>CONTINENT</th>
  </tr>
```

- Open “templates/countries_footer.html

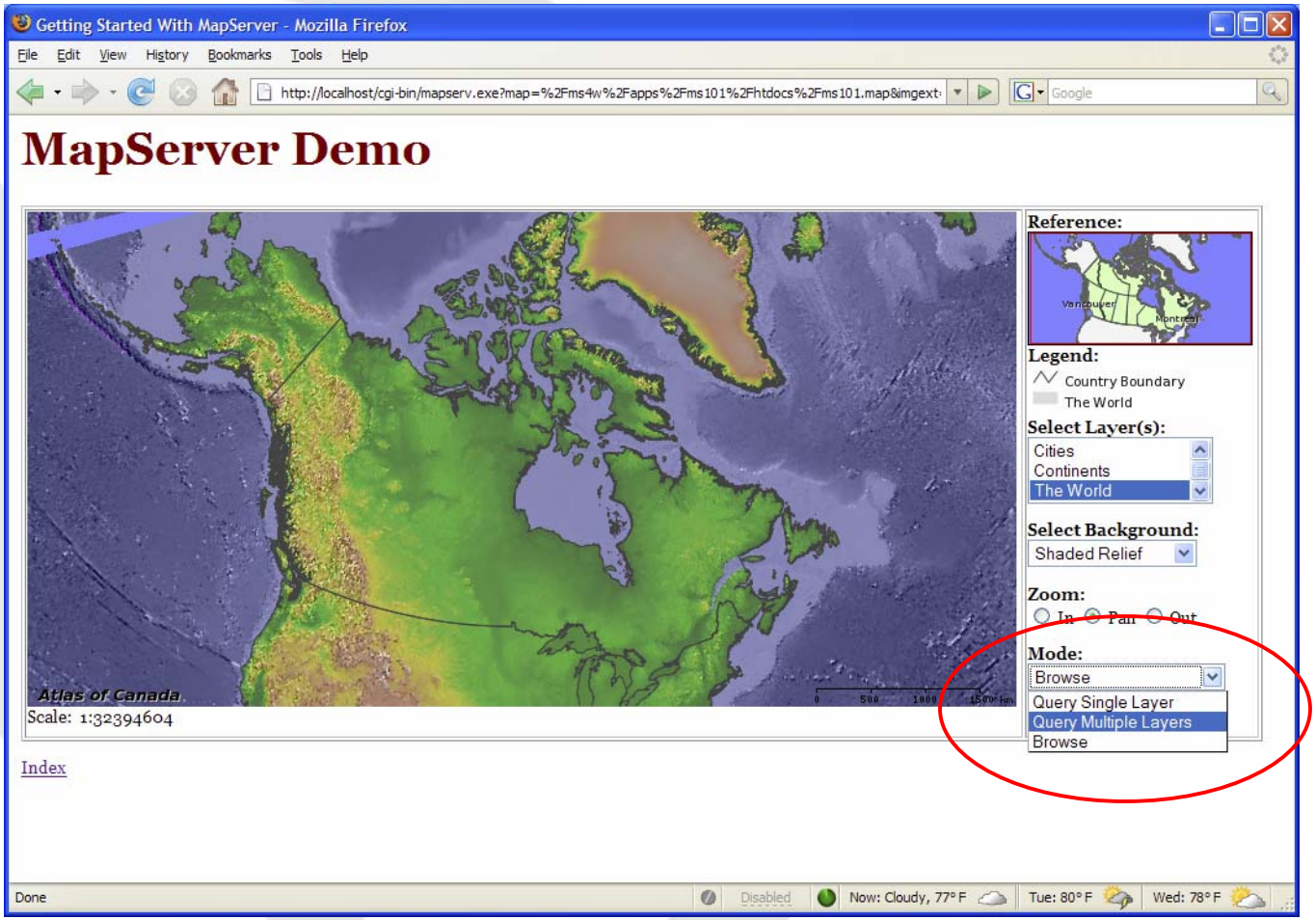
```
</table>
<p> </p>
```

- Finally, open “templates/countries_query.html

```
<tr>
  <td>[NA2DESC]</td>
  <td>[NA2]</td>
  <td>[NA3DESC]</td>
</tr>
```

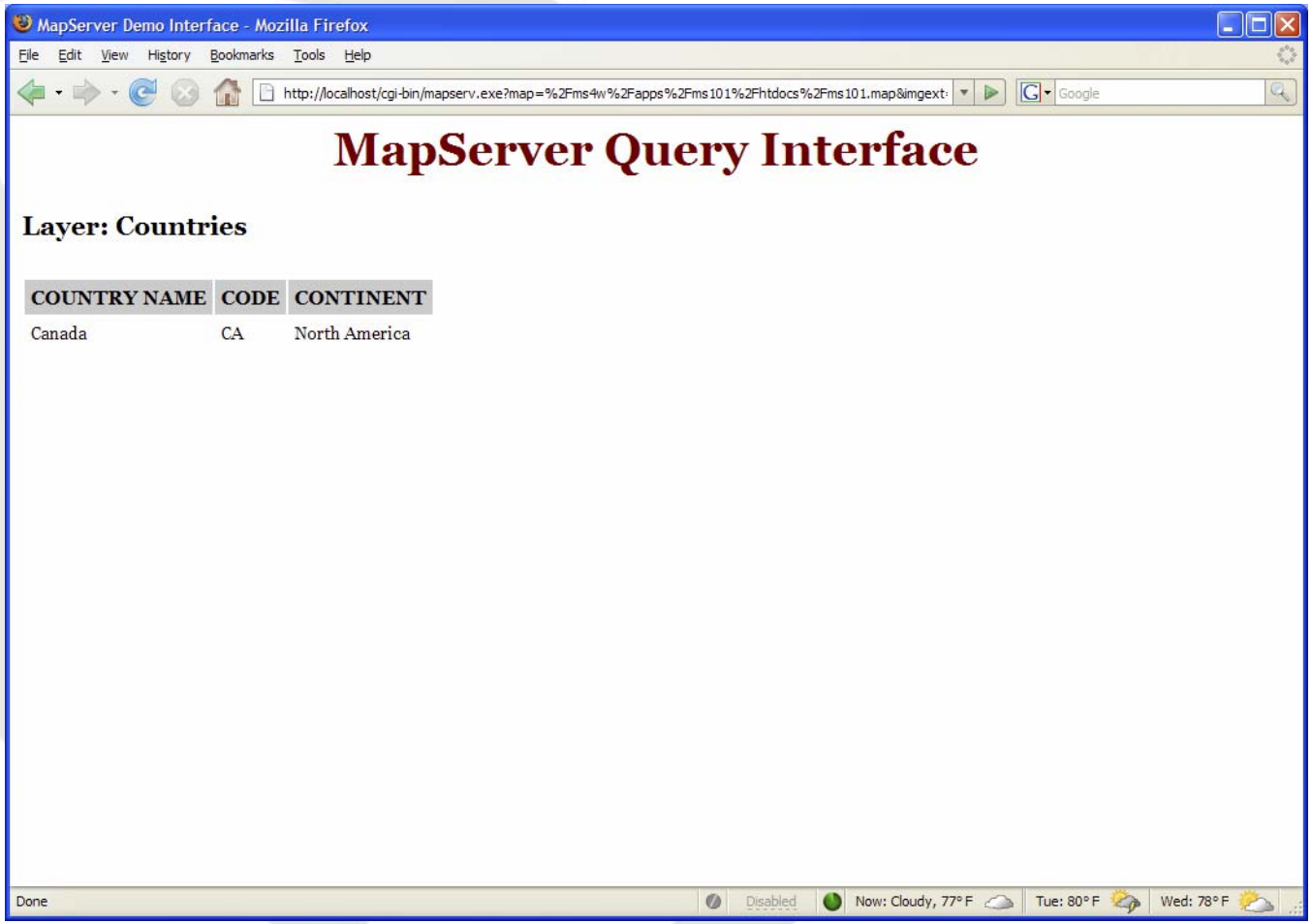
Example 15 Output

- Query Mode Selection:



Example 15 Output

- Query Result:



The Finished Demo Application

- The final mapfile is at /ms4w/apps/ms101/htdocs/ms101final.map
- The final HTML template is at /ms4w/apps/ms101/htdocs/ms101final.html
- Open a web browser and enter the following url: http://localhost/ms101/indexfinal.html

MapScript

Exposing the MapServer API to Scripting Languages

MapScript Overview

- Extends MapServer capabilities
- Seamless Integration with Other Applications
- Uses SWIG (except PHP)
- Support for Several Languages:
  - PHP
  - Python
  - Perl
  - Ruby
  - Java
  - C#
  - TCL/Tk
- See the MapScript API Reference for more information
- Demo: Tcl/Tk MapScript
- Demo: PHP/MapScript

Third Party Tools and Support

Client and Management Tools for MapServer Applications

Third Party Tools/Support

- Open Source Client Tools
  - MapLab
  - Chameleon
  - ka-Map! – Demo
  - CartoWeb – Demo
  - MapServer Workbench
  - MapBender
  - MapBuilder
  - OpenLayers
  - PMapper – Demo
  - PrimaGIS – Demo
  - Others
- Proprietary Client Tools
  - Neapoljs

Support and Other Resources

- Support Available via Mailing Lists
  - MapServer-Users
  - MapServer-Dev
- Additional Support via Consulting/Services Companies
- MapServer Website: http://mapserver.gis.umn.edu
- MapTools: http://www.maptools.org
- GDAL/OGR: http://www.gdal.org
- PROJ.4: http://proj.maptools.org
- OSGeo: https:www.osgeo.org,
- OSGeo Wiki: http://wiki.osgeo.org

Questions?