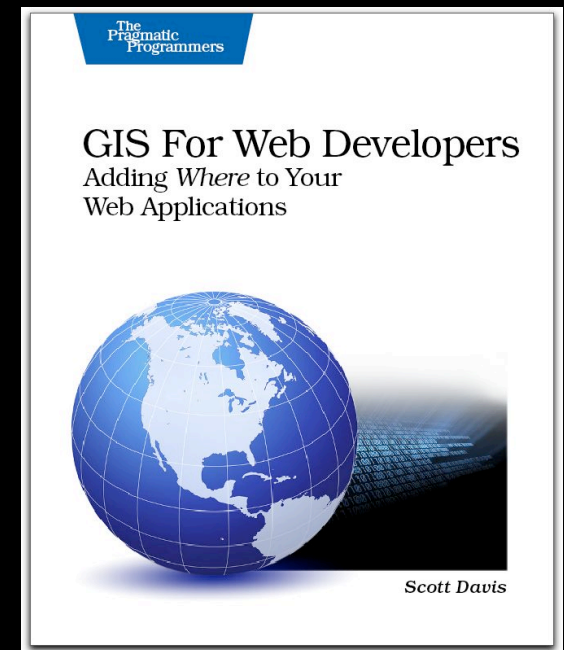


GIS for Web Developers

Adding *Where* to Your Application

Introduction

- My name is Scott Davis
 - ◆ Editor in Chief of <http://aboutGroovy.com>
 - ◆ Author
 - ★ Groovy Recipes:
Greasing the Wheels of Java
(Pragmatic Bookshelf)
 - ★ GIS for Web Developers
(Pragmatic Bookshelf)
 - ★ Google Maps API
(Pragmatic Bookshelf)
 - ★ JBoss At Work
(O'Reilly)



Demystifying GIS

- Geographic Information Systems sounds like something you need a degree in
 - ◆ (It might help, but it's not a requirement...)

Neo-Geography

- New wave of non-GIS professionals creating maps using free data and open source applications
 - ◆ Google Maps, et al

View / Source

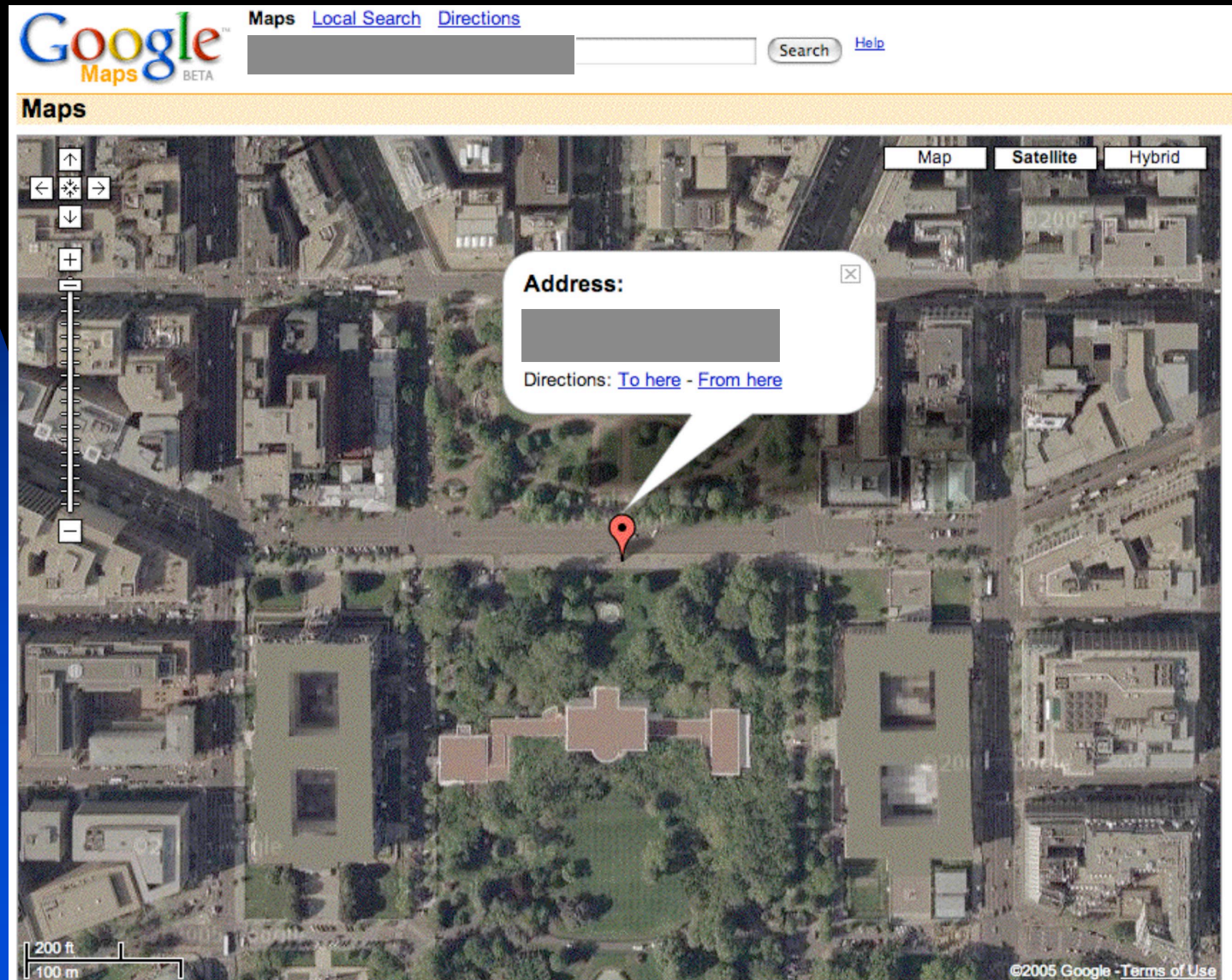
- This talk is the equivalent of doing a View / Source on web maps
 - ◆ You will learn the jargon
 - ◆ You will find free sources of data
 - ◆ You will find free applications to use

Act 1: Free Data

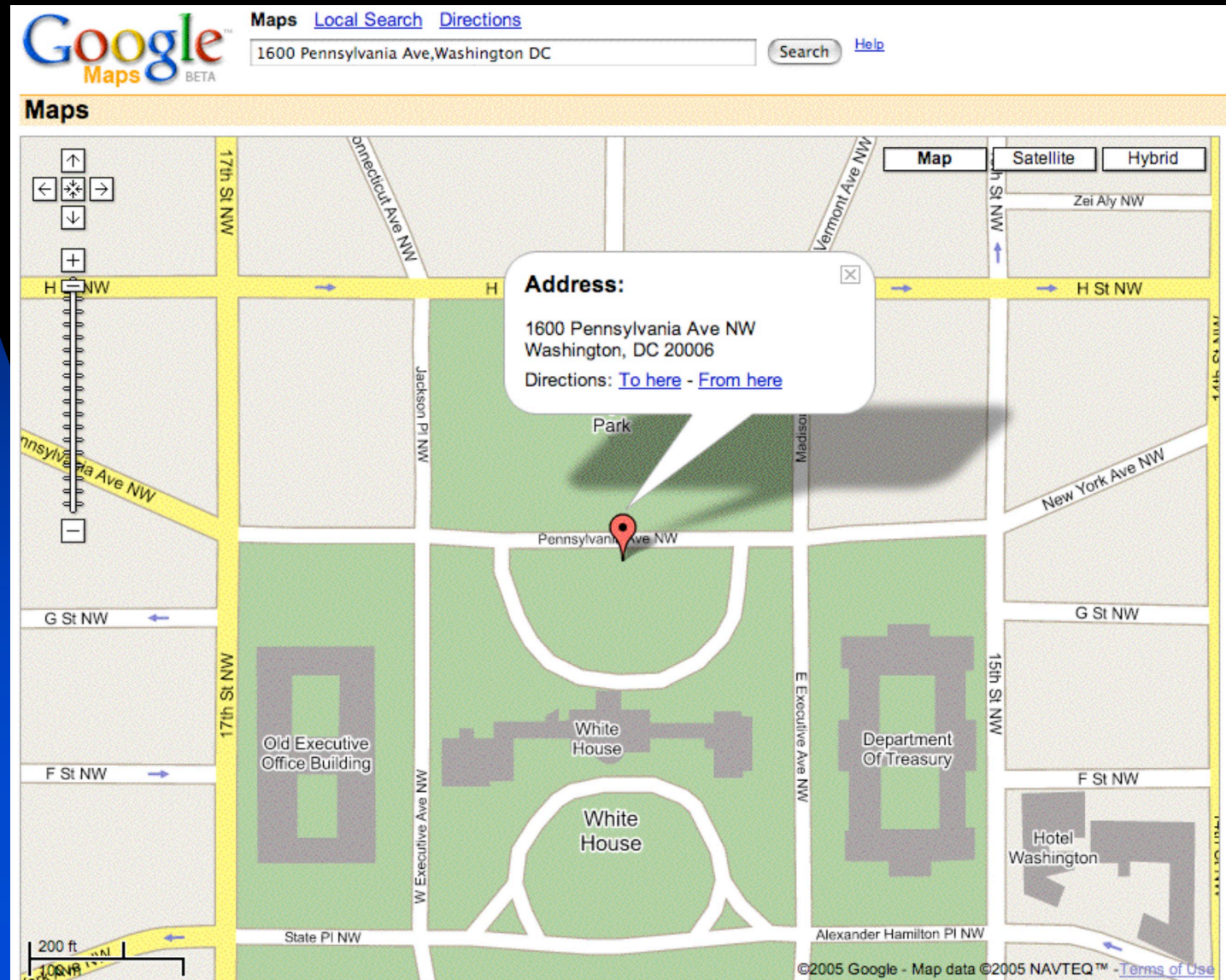
Vectors vs. Rasters

- There are two types of data out there:

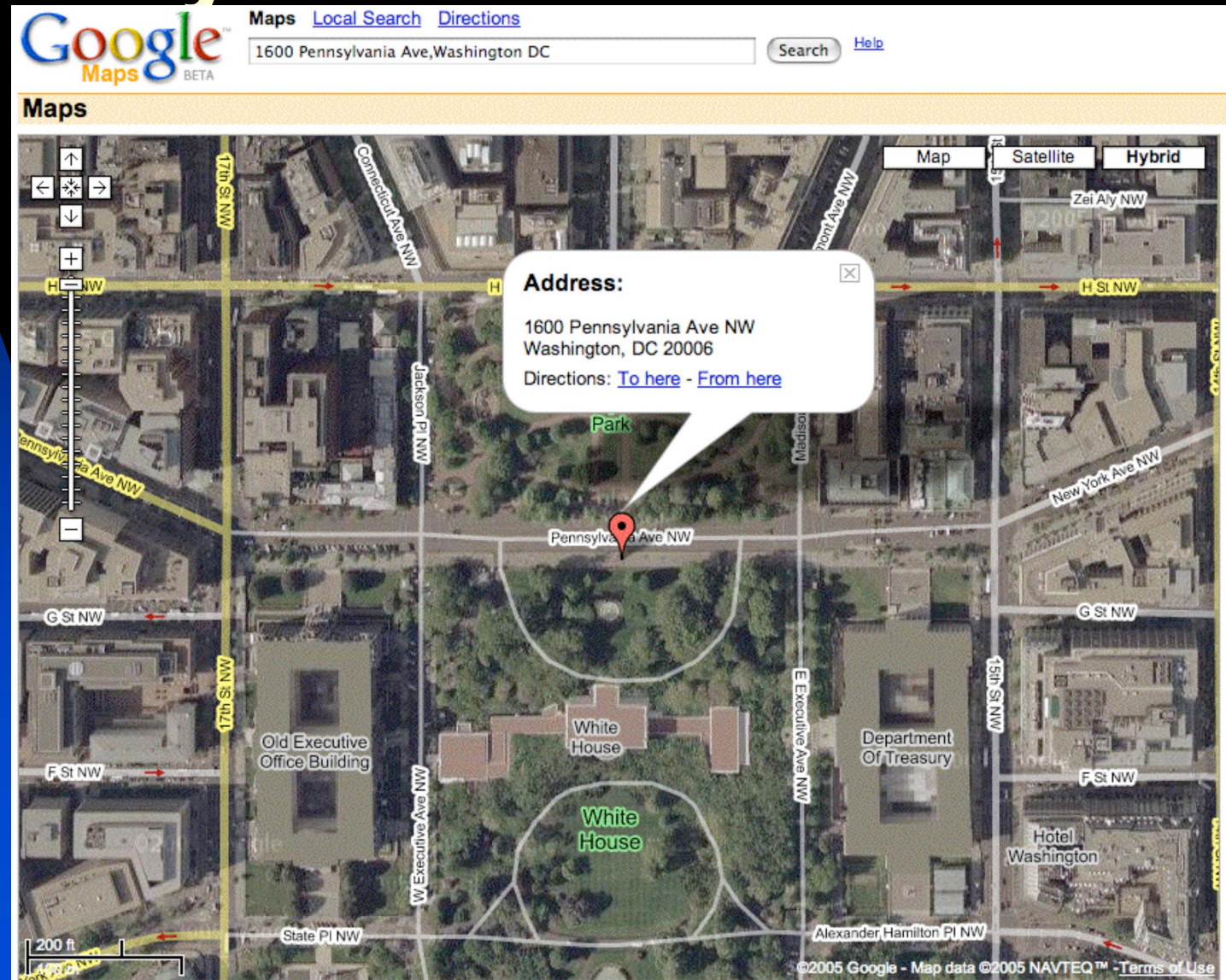
Raster Data



Vector Data



Hybrid

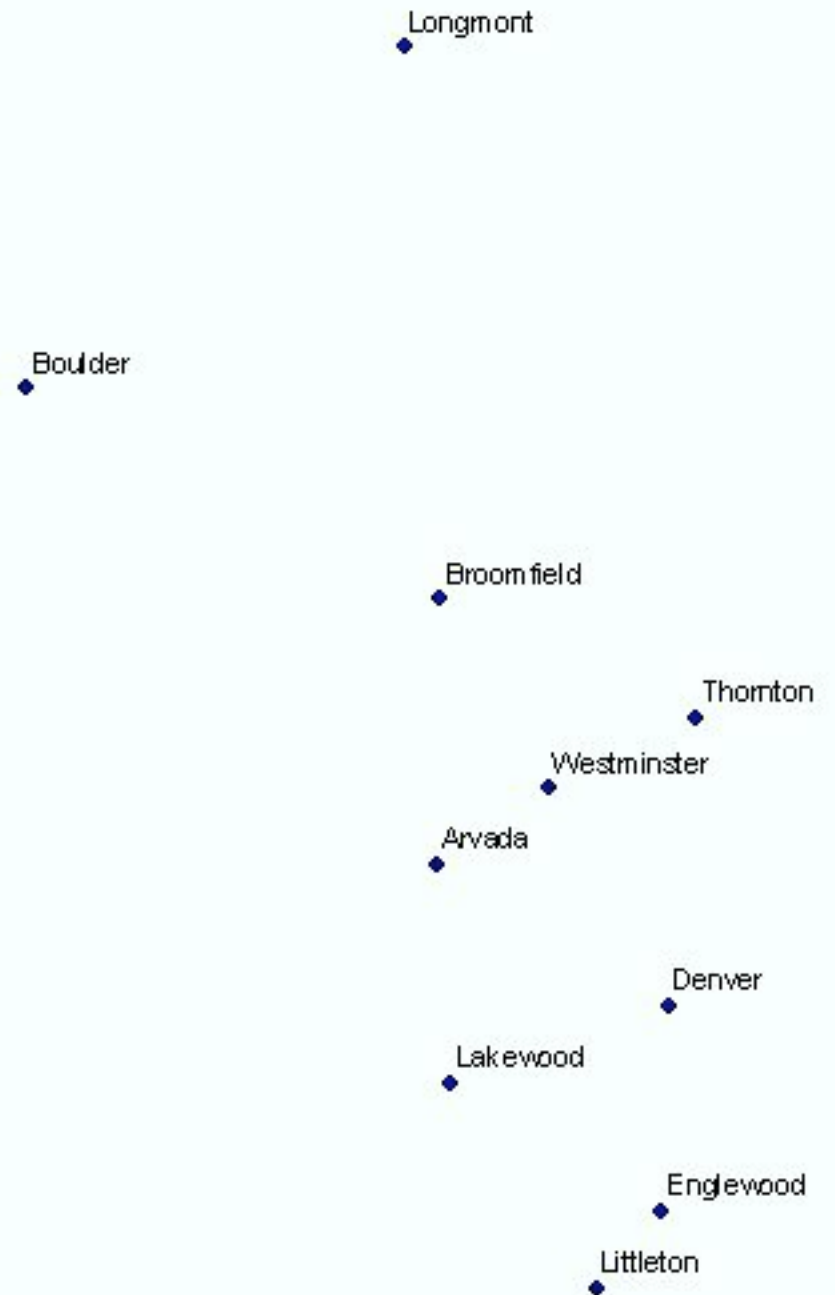


Type of Vectors

- Vectors come in three primitive datatypes:

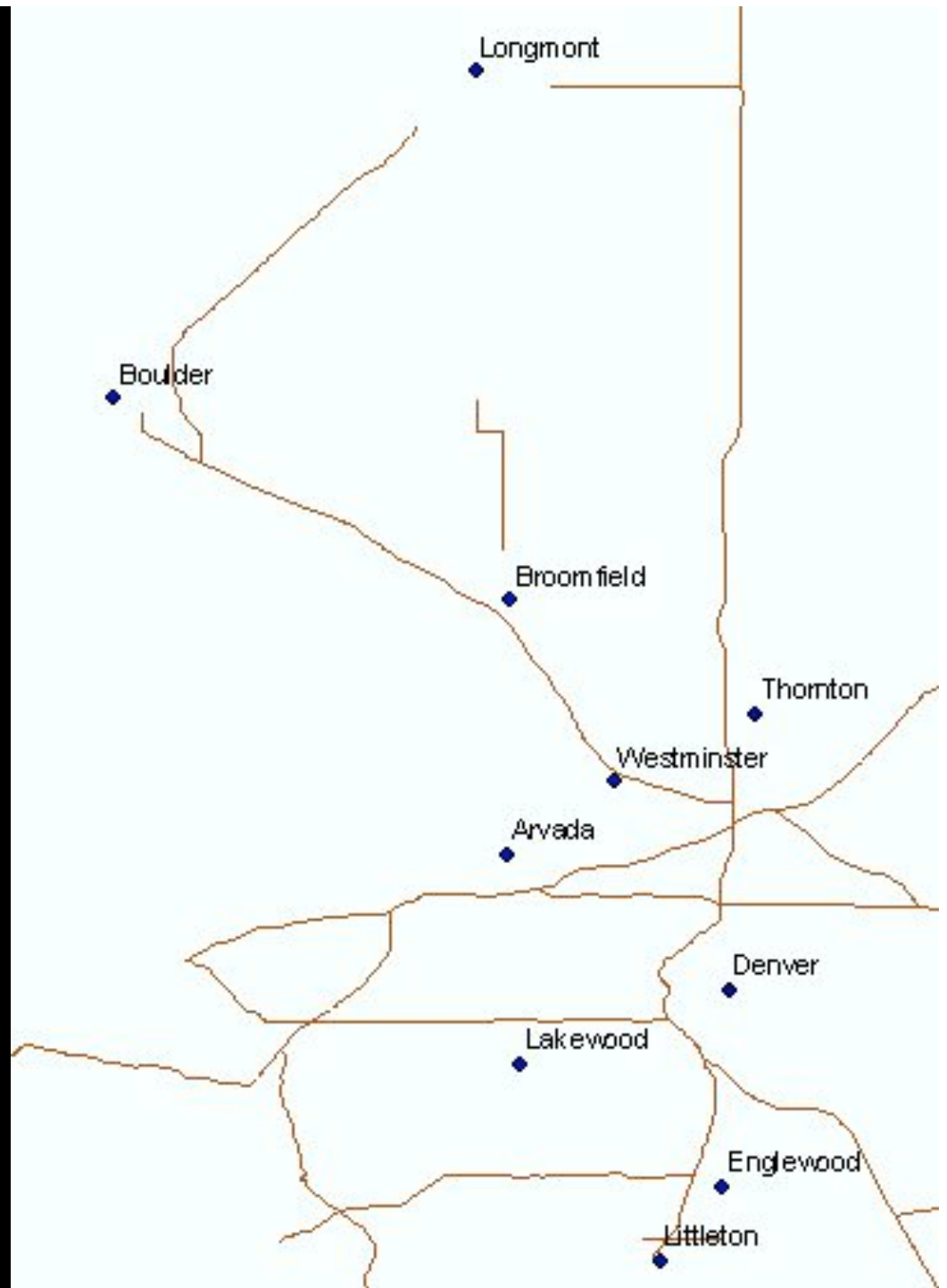
Point

© 2007, Davisworld.org

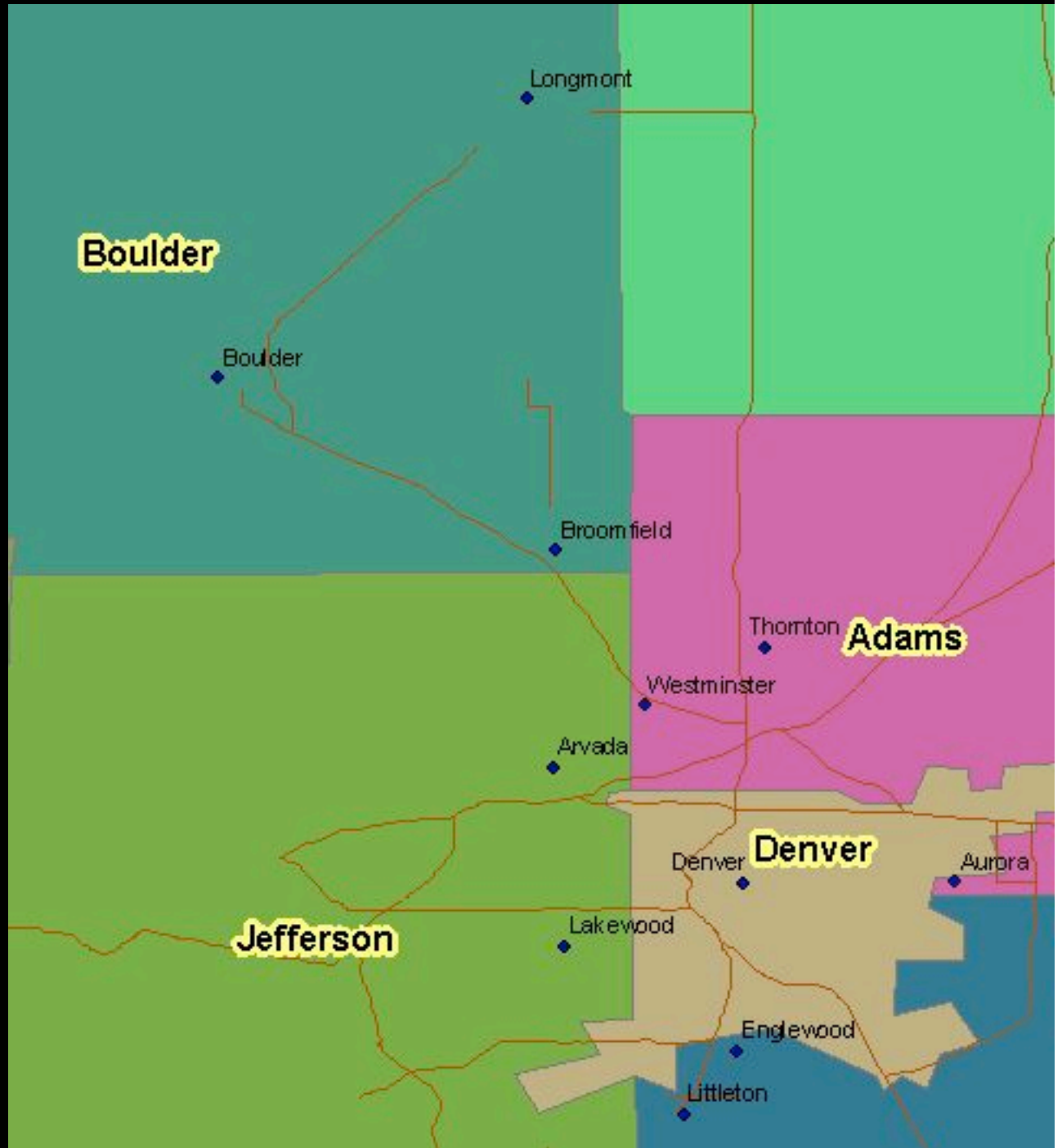


Line

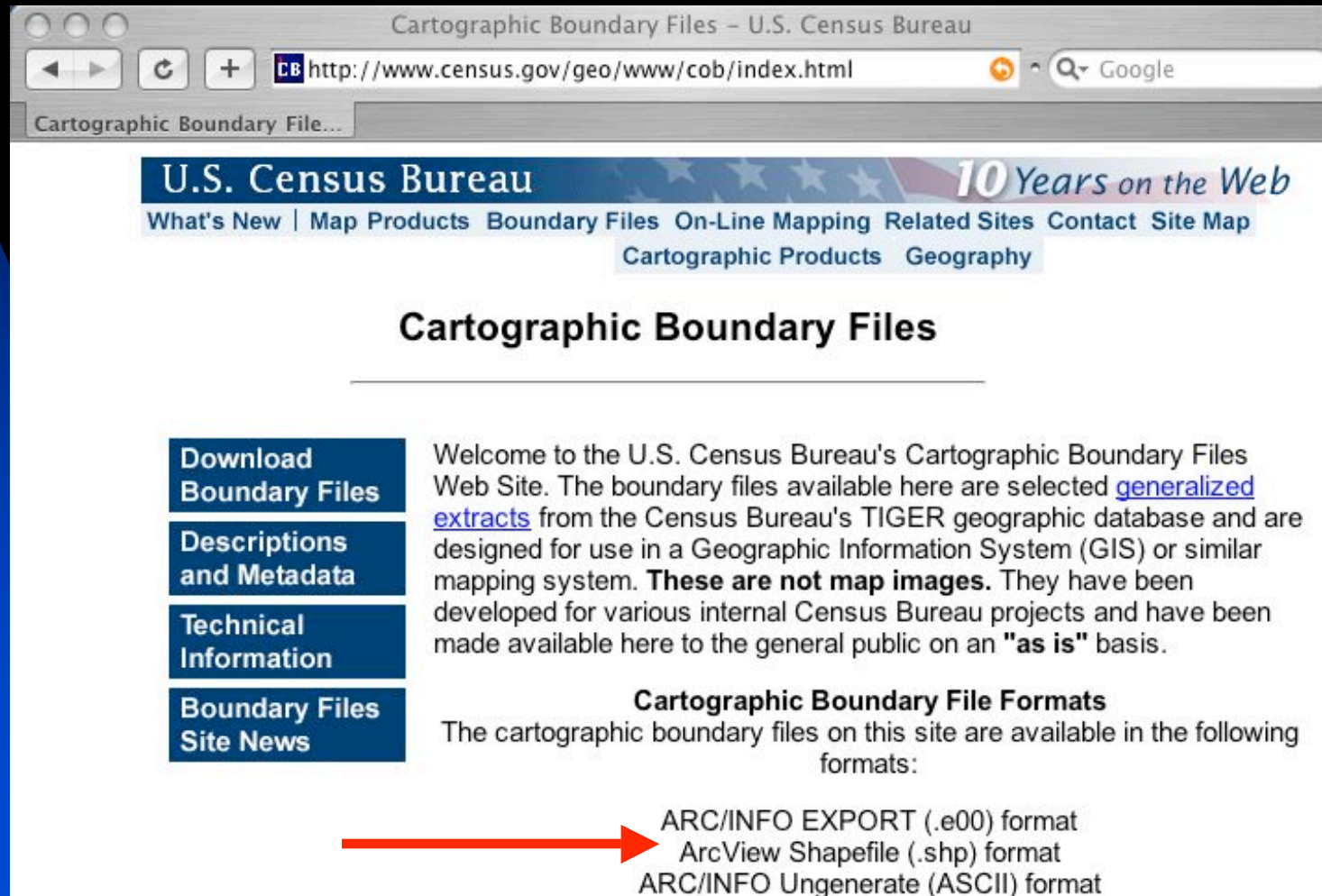
© 2007, Davisworld.org



Polygon



Free Vectors:



Cartographic Boundary Files – U.S. Census Bureau

U.S. Census Bureau *10 Years on the Web*

What's New | Map Products | Boundary Files | On-Line Mapping | Related Sites | Contact | Site Map

Cartographic Products | Geography

Cartographic Boundary Files

Download Boundary Files

Descriptions and Metadata

Technical Information

Boundary Files Site News

Welcome to the U.S. Census Bureau's Cartographic Boundary Files Web Site. The boundary files available here are selected [generalized extracts](#) from the Census Bureau's TIGER geographic database and are designed for use in a Geographic Information System (GIS) or similar mapping system. **These are not map images.** They have been developed for various internal Census Bureau projects and have been made available here to the general public on an "as is" basis.

Cartographic Boundary File Formats

The cartographic boundary files on this site are available in the following formats:

- ARC/INFO EXPORT (.e00) format
- ArcView Shapefile (.shp) format
- ARC/INFO Ungenerate (ASCII) format

Shapefile

- Shapefiles are a proprietary ESRI standard, but the format is well documented:
 - ◆ <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>
- Each shapefile is made up of several files:
 - ◆ .shp, .shx, .dbf

The Downloadable States of America

- http://www.census.gov/geo/cob/bdy/st/st00shp/st99_d00_shp.zip



Uh, OK.

Now what?

Downloading Viewers

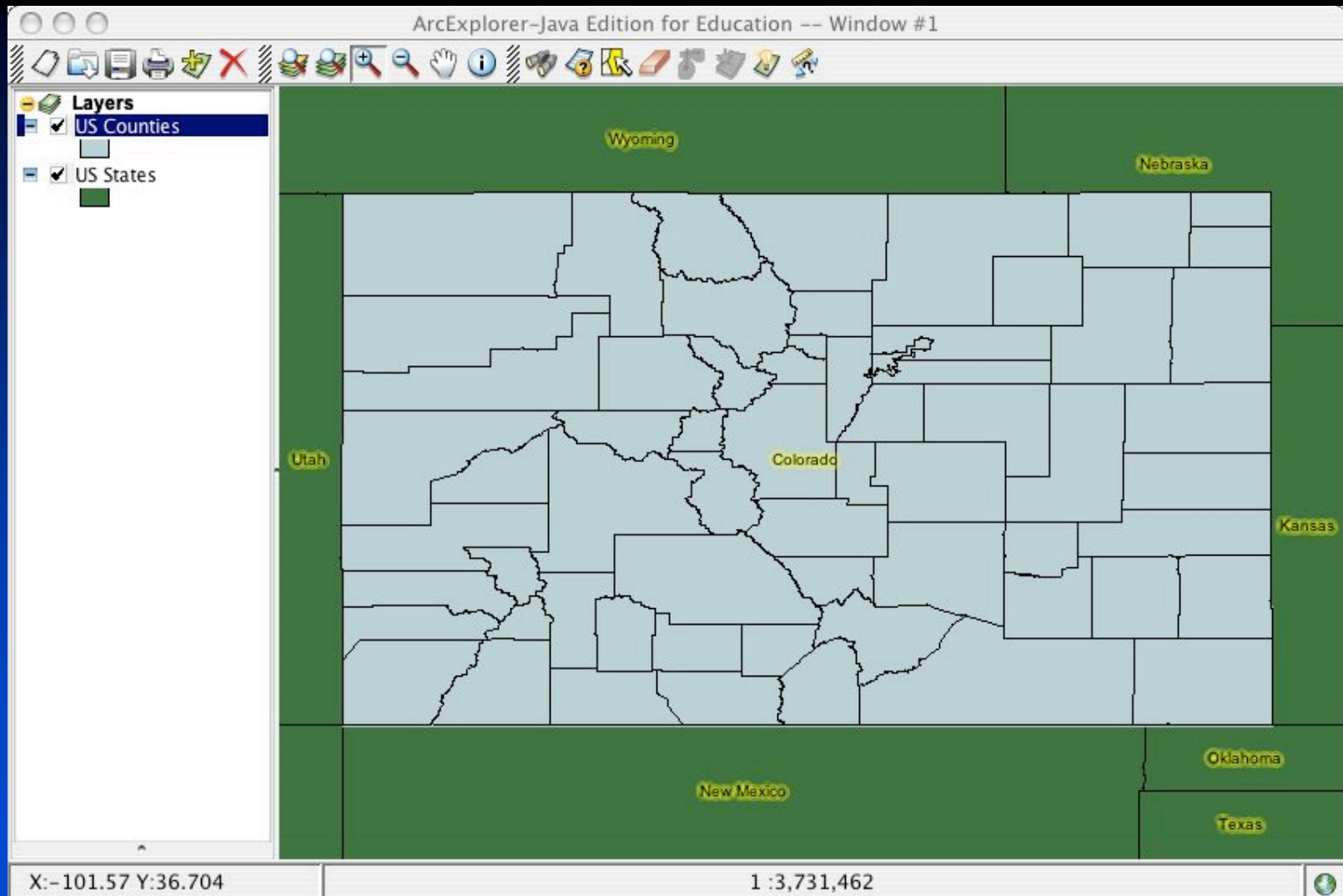
- ESRI offers a free viewer:
 - ◆ <http://www.esri.com/software/arcexplorer/download.html>

Non-Spatial Attributes

- Features can have non map-related data that comes along for the ride:
 - ◆ Right-click, Attribute Table

Data Layers

- You can add multiple layers to a map
 - ◆ Each layer can be styled



More Data, Please

The screenshot shows a web browser window titled "Map Layers" with the URL <http://nationalatlas.gov/maplayers.html>. The page features the National Atlas logo and a navigation menu with categories like Agriculture, Biology, Boundaries, Climate, Environment, Geology, Government, History, Mapping, People, Transportation, and Water. A "MAP MAKER" button is also visible. The main content area is titled "Map Layers" and contains a paragraph explaining the National Atlas Map Maker's functionality. Below this, a section titled "Map Layers by Chapter" lists various map layers organized by subject.

Map Layers

In the National Atlas Map Maker, you can assemble, view, and print your own maps. You can choose from hundreds of layers of geographic information to make maps. Each map layer can be displayed individually or mixed with others as you tailor a map to your needs. For example, you can make a map showing America's streams and lakes. And you can add new map layers showing additional geographic information, such as state boundaries, county boundaries, roads, railroads, and towns and cities.

This index lists all of the map layers currently contained in nationalatlas.gov™, by broad subject categories that correspond to the chapters of the National Atlas.

Map Layers by Chapter		
↓ Agriculture	↓ Environment	↓ People
↓ Biology	↓ Geology	↓ Transportation
↓ Boundaries	↓ History	↓ Water
↓ Climate	↓ Map Reference	

More *International* Data, Please

- Canada:
 - ◆ <http://www.geobase.ca/>



More *Local* Data, Please

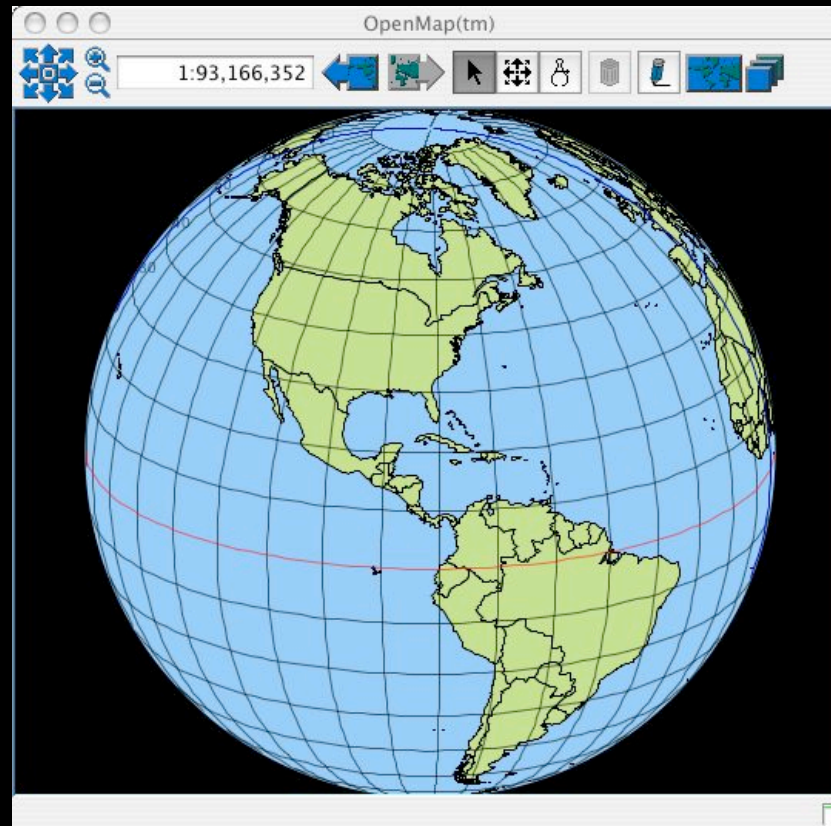
- Most state and municipal government agencies offer free data as well



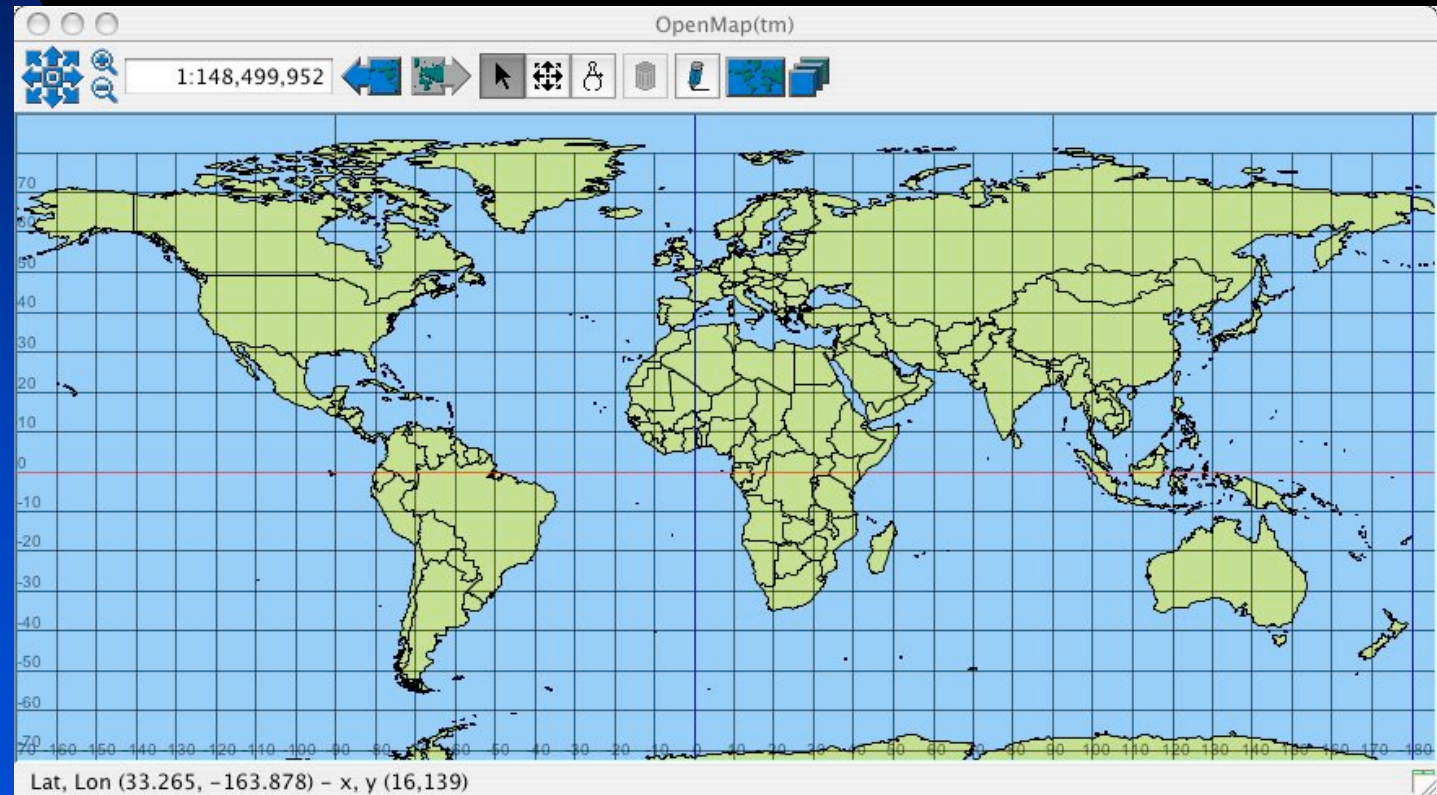
Act 2: Projections

When Good Data Goes Bad...

■ Newsflash: The Earth is Round!



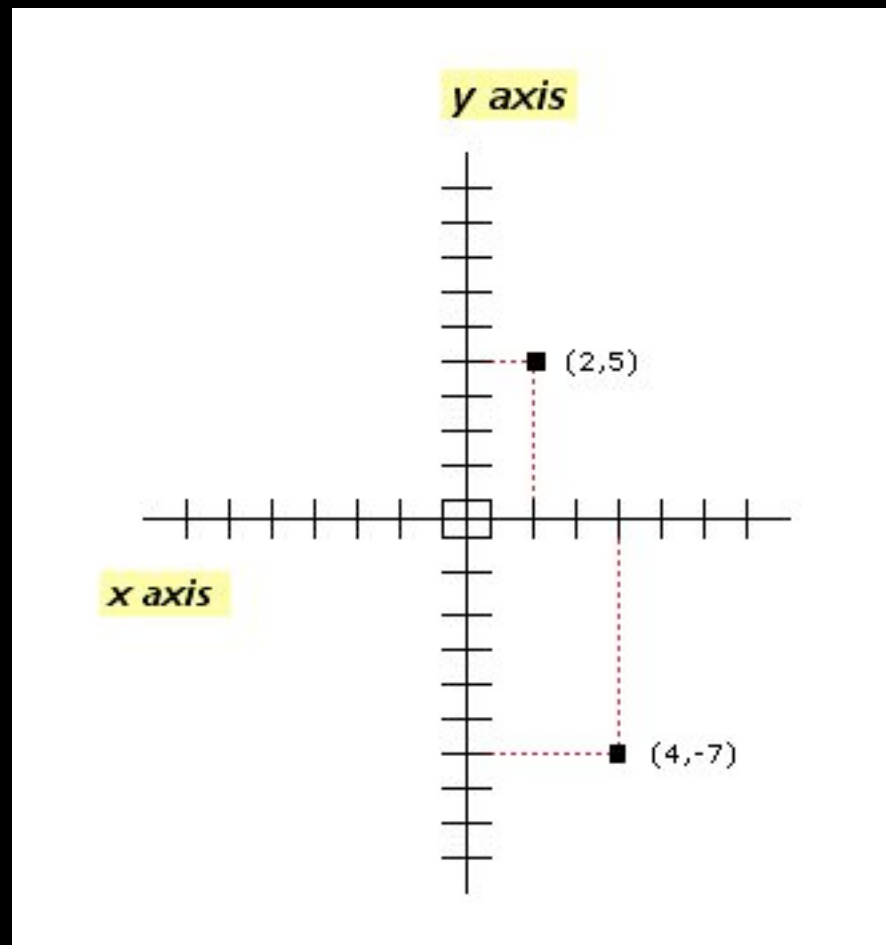
■ This Just In: Maps are Flat!



Projections

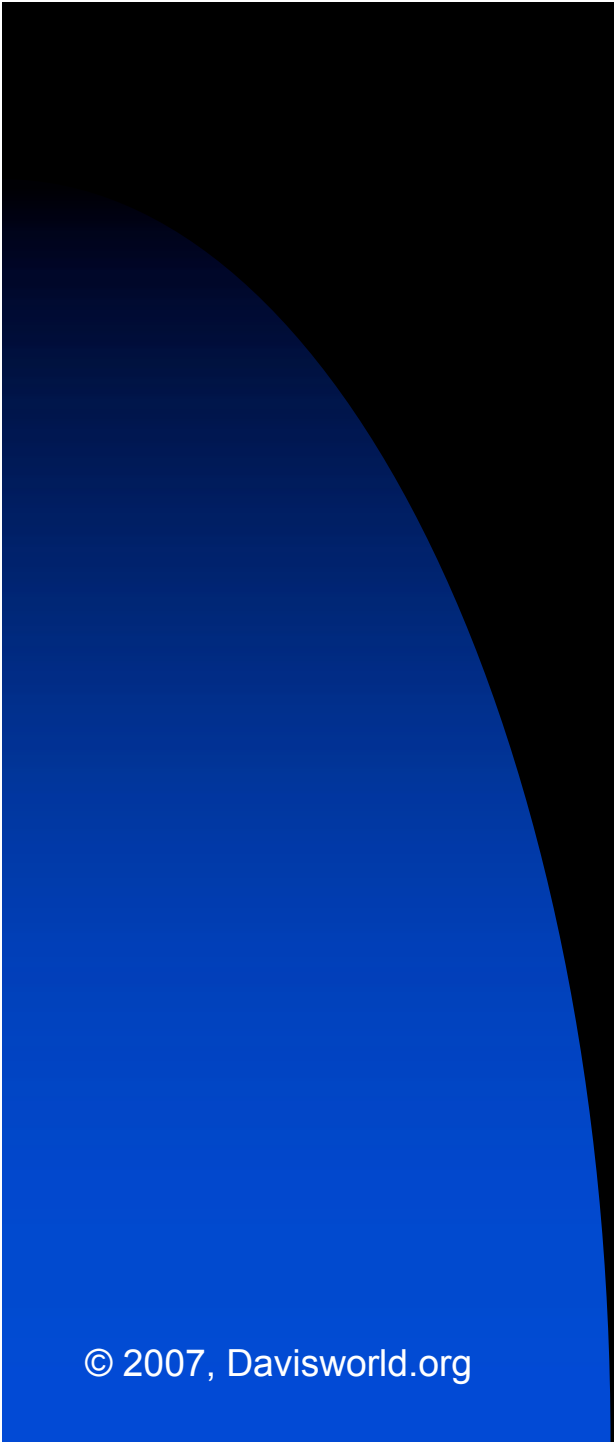
- Map Projections:
 - ◆ Taking fundamentally 3-dimensional data and portraying it in 2-dimensions
 - ◆ This always introduces error:
 - ★ Distance
 - ★ Direction
 - ★ Shape
 - ★ Area

Cartesian Planes



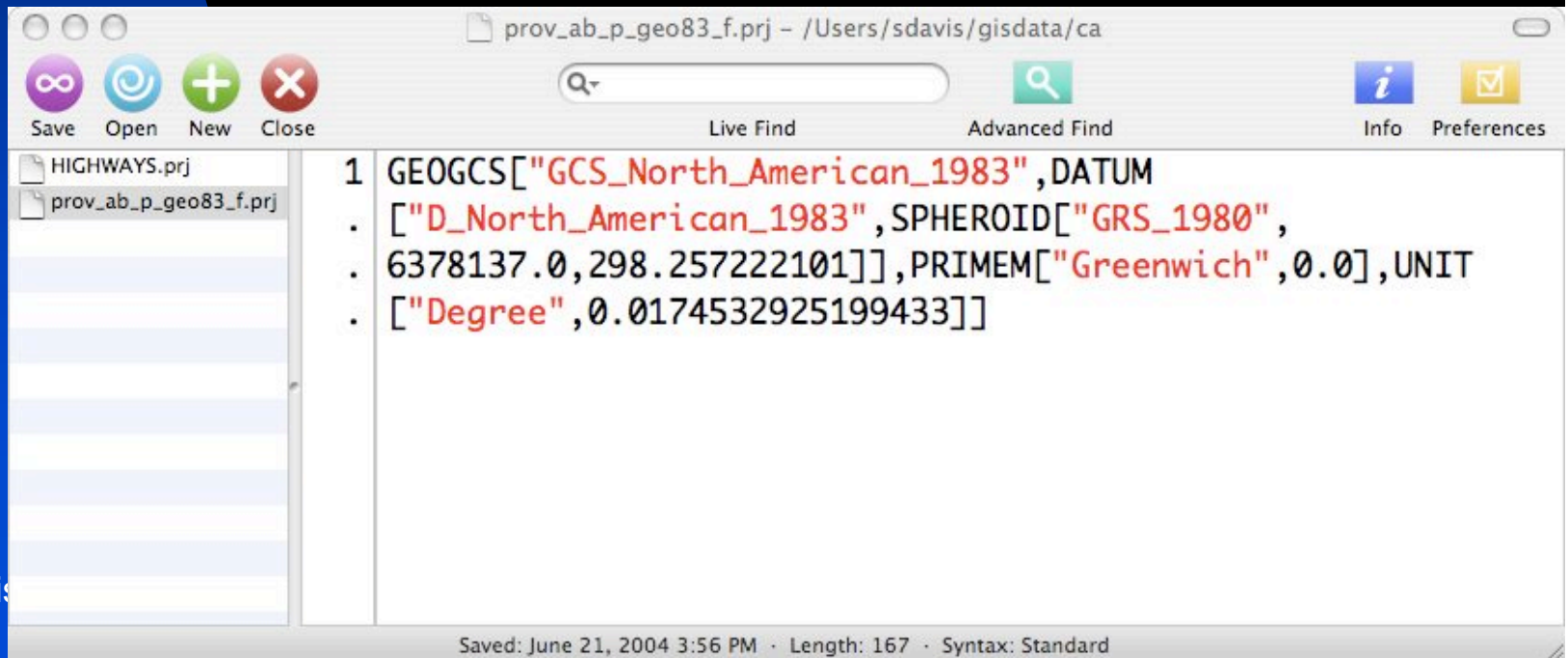
The (Mapping) Problem with Cartesian Planes

- On a Cartesian plane, the X and Y lines form perfect squares
- On a globe, only the lines of Latitude are perfectly parallel
 - ◆ Line of Longitude converge at the poles
 - ◆ This yields trapezoids that ultimately turn into triangles

- 
- Different projections attempt to minimize map distortions
 - ◆ Unprojected data is common at the World and Country level
 - ◆ Projected data is common at the State and Local level
 - ★ Common projections are “State Plane” and “UTM”

Shapefile Projections

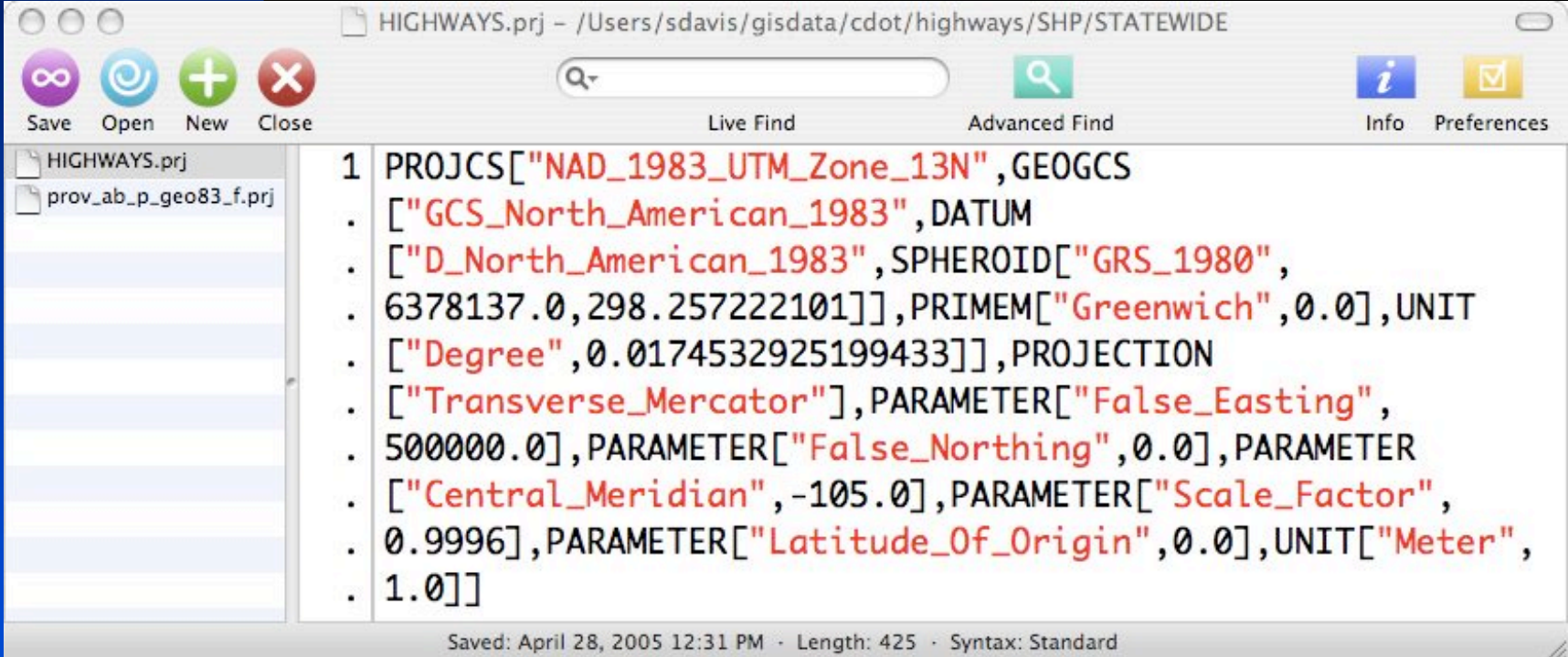
- If a shapefile has been reprojected, it should contain a .prj file
 - ◆ Canada:



```
1 GEOGCS["GCS_North_American_1983", DATUM
. ["D_North_American_1983", SPHEROID["GRS_1980",
. 6378137.0, 298.257222101]], PRIMEM["Greenwich", 0.0], UNIT
. ["Degree", 0.0174532925199433]]
```

Saved: June 21, 2004 3:56 PM · Length: 167 · Syntax: Standard

■ Colorado:

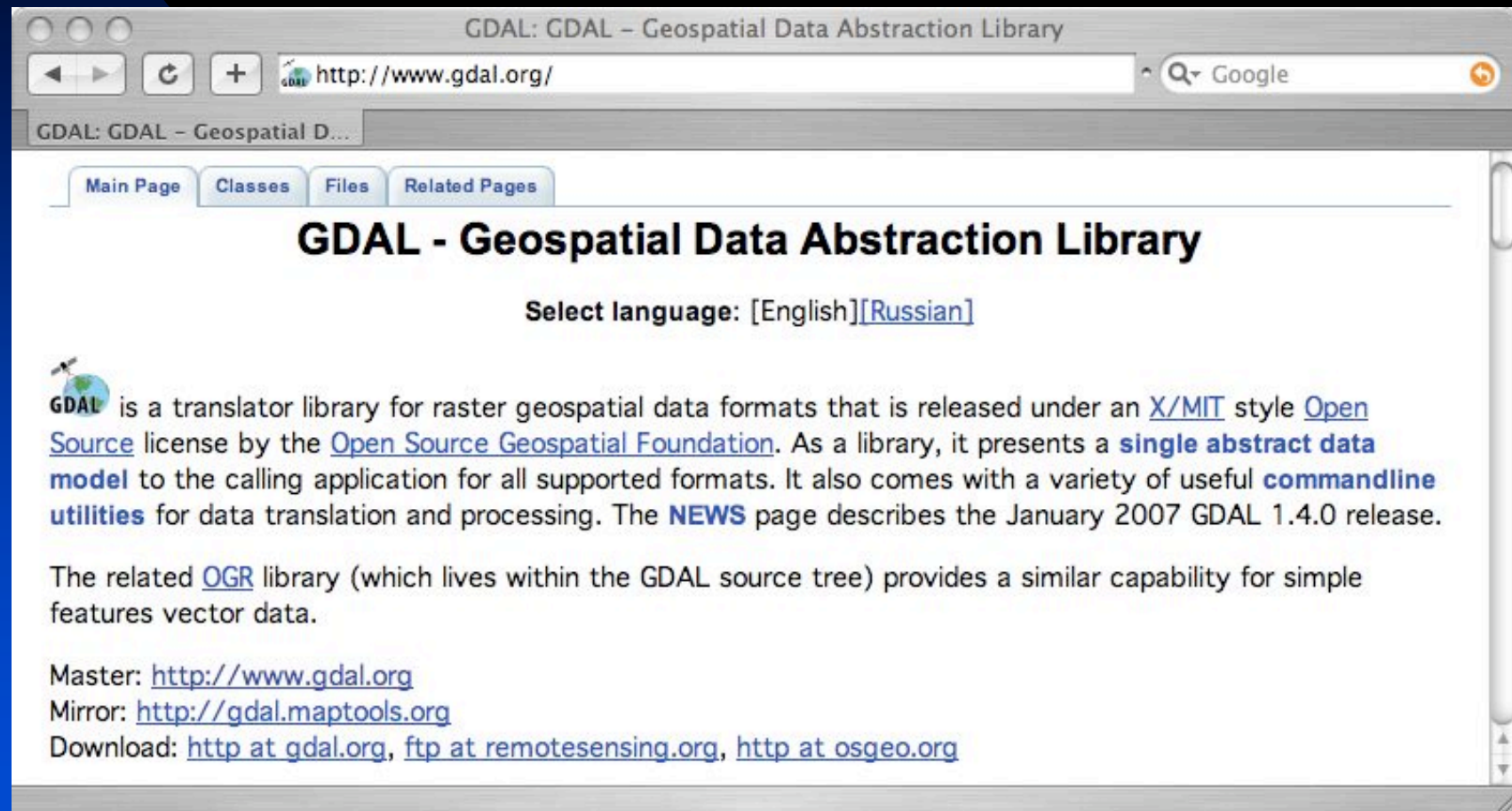


The screenshot shows a GIS application window titled "HIGHWAYS.prj - /Users/sdavis/gisdata/cdot/highways/SHP/STATEWIDE". The window has a menu bar with "Save", "Open", "New", and "Close" buttons. Below the menu bar is a search bar with "Live Find" and "Advanced Find" options. On the right side of the menu bar are "Info" and "Preferences" buttons. The main area of the window displays a list of files on the left and a text editor on the right. The text editor shows the PROJCS definition for "NAD_1983_UTM_Zone_13N".

```
1 PROJCS["NAD_1983_UTM_Zone_13N",GEOGCS
. ["GCS_North_American_1983",DATUM
. ["D_North_American_1983",SPHEROID["GRS_1980",
. 6378137.0,298.257222101]],PRIMEM["Greenwich",0.0],UNIT
. ["Degree",0.0174532925199433]],PROJECTION
. ["Transverse_Mercator"],PARAMETER["False_Easting",
. 500000.0],PARAMETER["False_Northing",0.0],PARAMETER
. ["Central_Meridian",-105.0],PARAMETER["Scale_Factor",
. 0.9996],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",
. 1.0]]
```

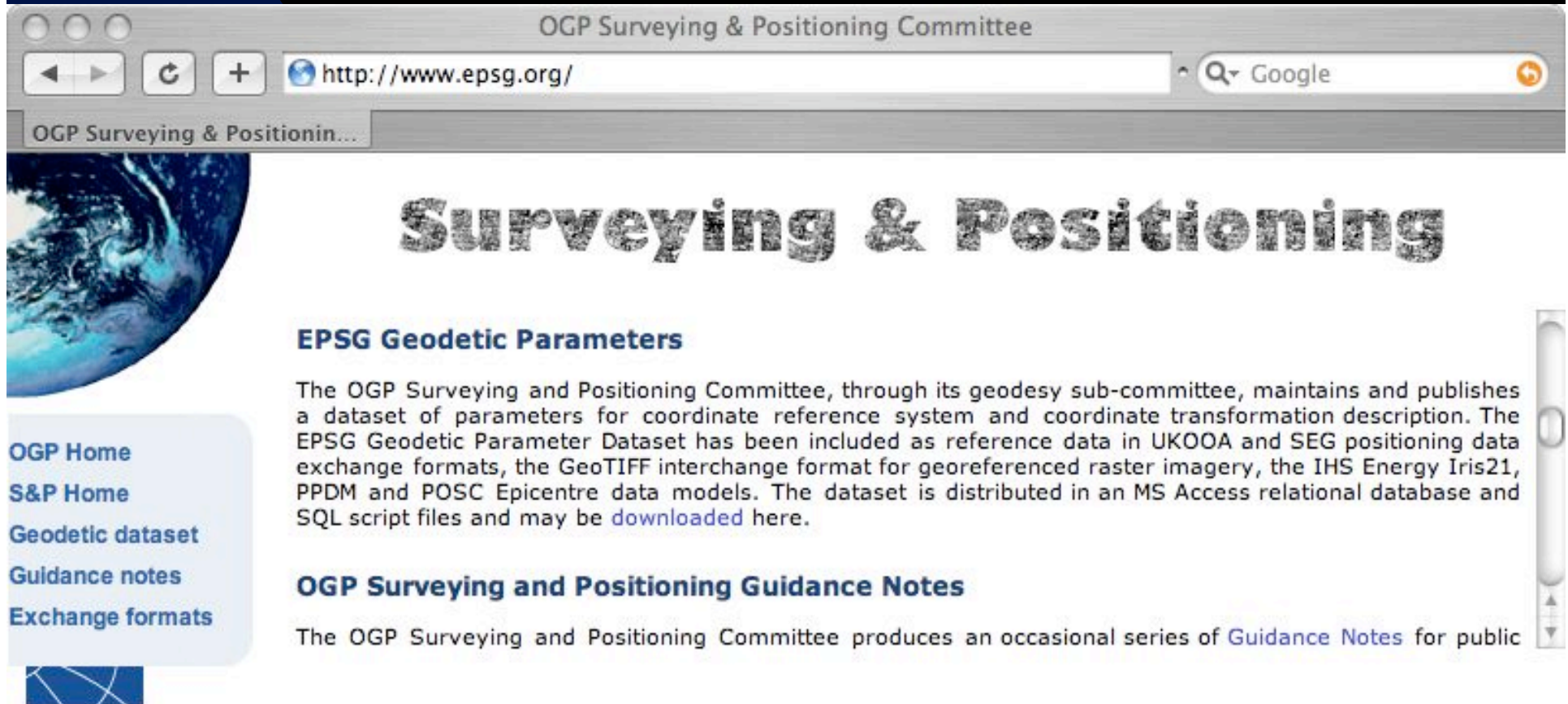
Saved: April 28, 2005 12:31 PM · Length: 425 · Syntax: Standard

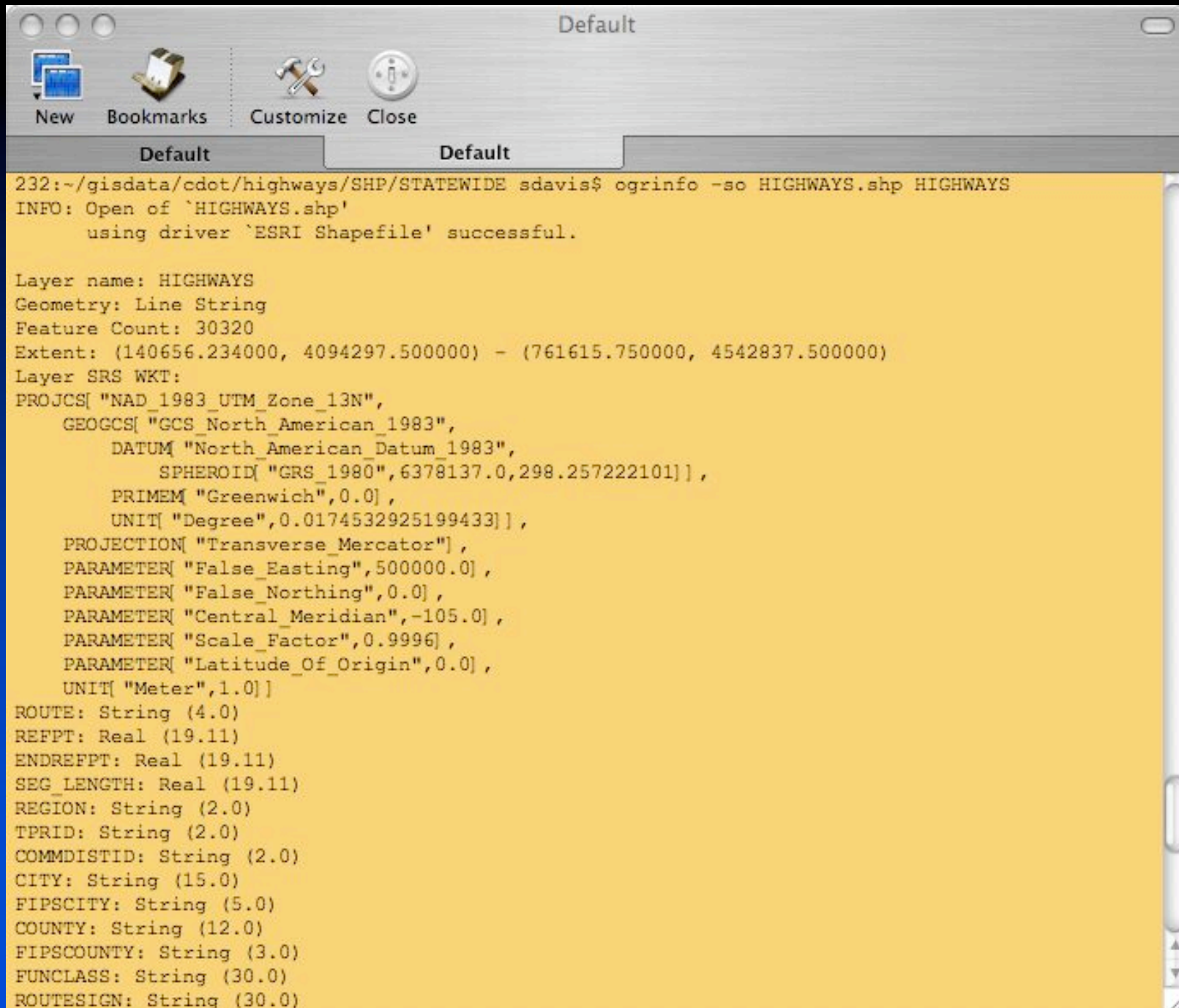
Reprojection Utility



- To reproject from “UTM 13 N” to “Unprojected NAD83”
 - ◆ `ogr2ogr -t_srs EPSG:4269 co-hw.shp highways.shp`
- For info
 - ◆ `ogrinfo -so highways.shp highways`
- A very common projection is WGS 84, or EPSG: 4326

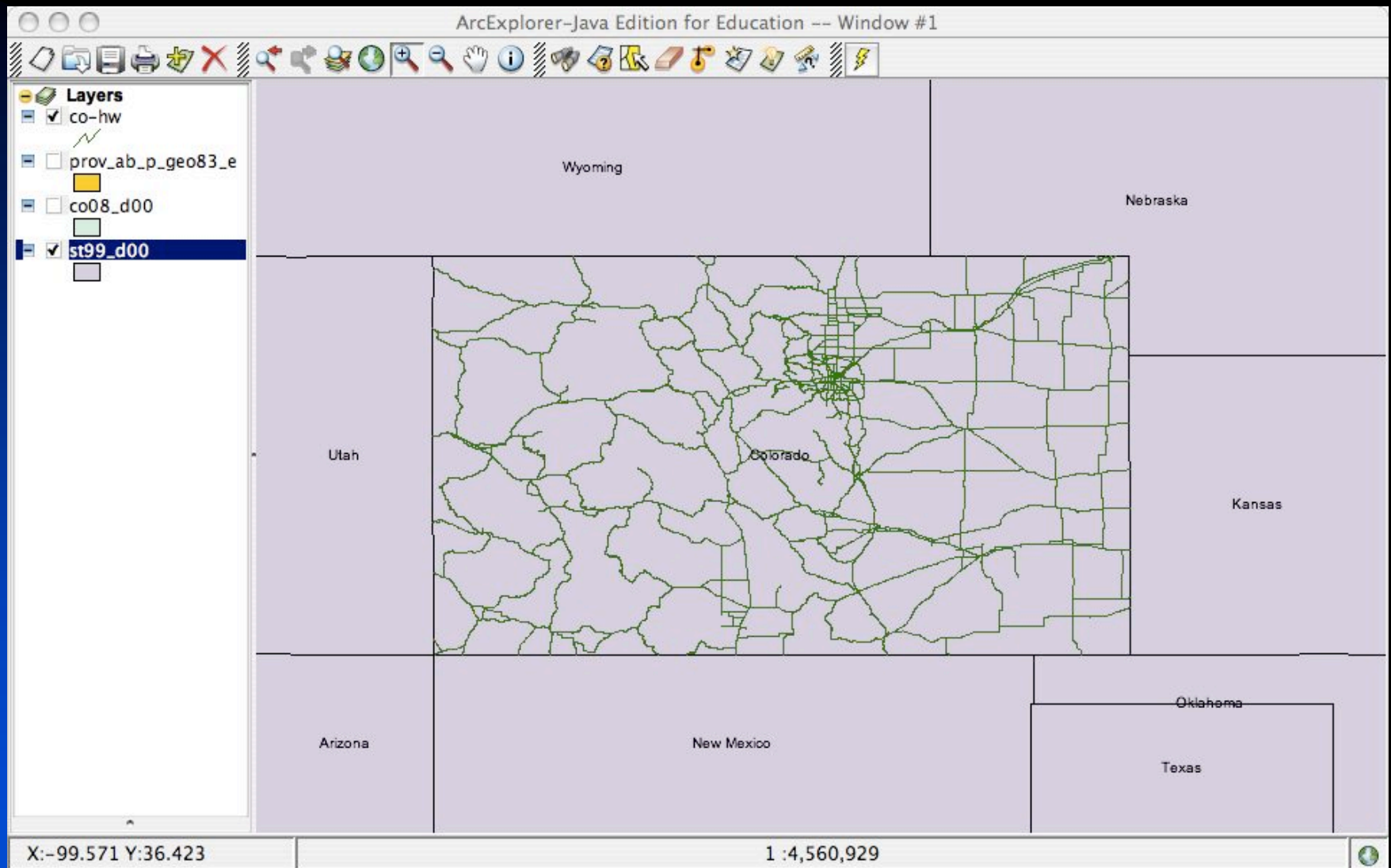
EPSG Codes





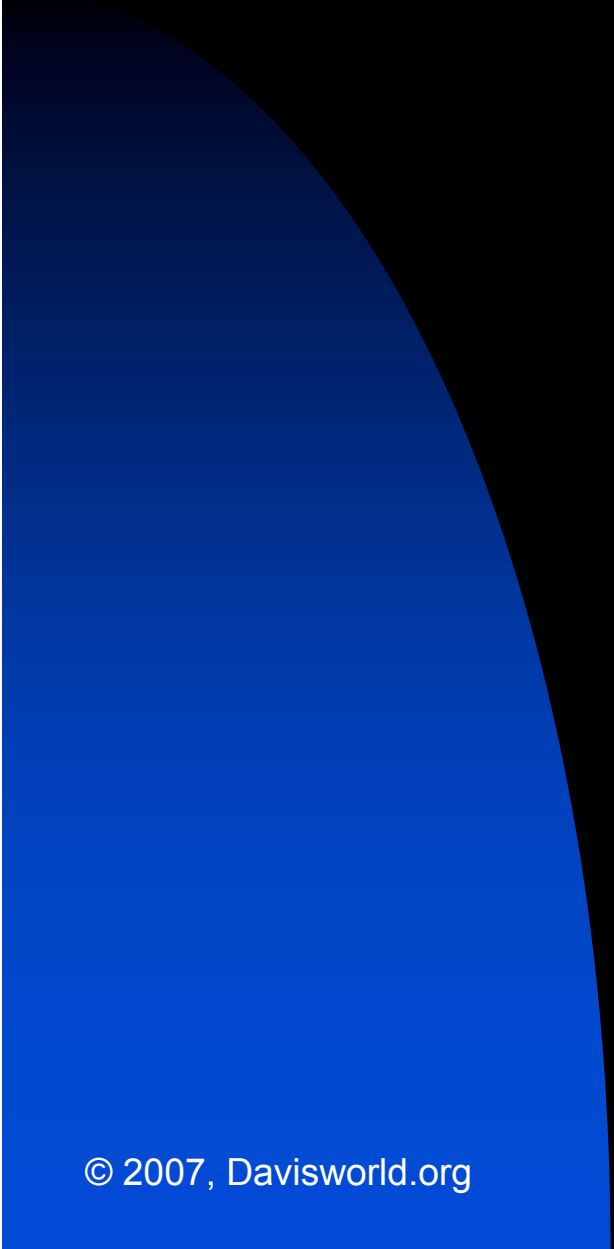
```
232:~/gisdata/cdot/highways/SHP/STATEWIDE sdavis$ ogrinfo -so HIGHWAYS.shp HIGHWAYS
INFO: Open of `HIGHWAYS.shp'
      using driver `ESRI Shapefile' successful.

Layer name: HIGHWAYS
Geometry: Line String
Feature Count: 30320
Extent: (140656.234000, 4094297.500000) - (761615.750000, 4542837.500000)
Layer SRS WKT:
PROJCS["NAD_1983_UTM_Zone_13N",
  GEOGCS["GCS_North_American_1983",
    DATUM["North_American_Datum_1983",
      SPHEROID["GRS_1980",6378137.0,298.257222101]],
    PRIMEM["Greenwich",0.0],
    UNIT["Degree",0.0174532925199433]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["False_Easting",500000.0],
  PARAMETER["False_Northing",0.0],
  PARAMETER["Central_Meridian",-105.0],
  PARAMETER["Scale_Factor",0.9996],
  PARAMETER["Latitude_Of_Origin",0.0],
  UNIT["Meter",1.0]]
ROUTE: String (4.0)
REFPT: Real (19.11)
ENDREFPT: Real (19.11)
SEG_LENGTH: Real (19.11)
REGION: String (2.0)
TPRID: String (2.0)
COMMDISTID: String (2.0)
CITY: String (15.0)
FIPSCITY: String (5.0)
COUNTY: String (12.0)
FIPSCOUNTY: String (3.0)
FUNCLASS: String (30.0)
ROUTESIGN: String (30.0)
```

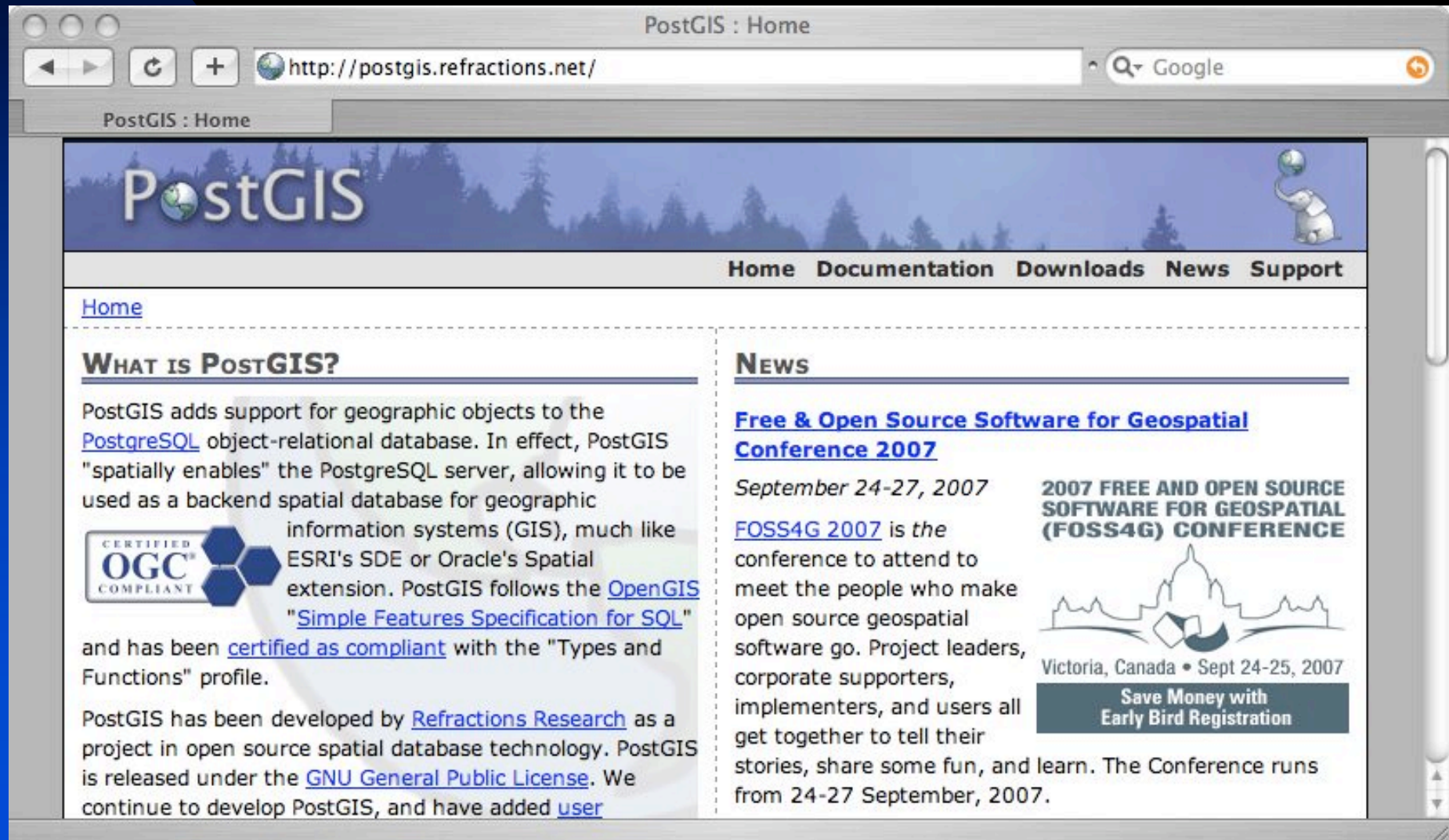


A large, solid blue curved shape on the left side of the slide, resembling a quarter-circle or a stylized 'C' shape, extending from the bottom left towards the top left.

Act 3: Spatial Databases

- 
- Why bother with a database?
 - ◆ Centralize many scattered files
 - ◆ Provide security
 - ◆ Indexing
 - ◆ Cross dataset queries

PostgreSQL + PostGIS



The screenshot shows a web browser window with the title "PostGIS : Home". The address bar contains "http://postgis.refractions.net/" and the search bar contains "Google". The page features a blue header with the "PostGIS" logo and a small elephant icon. Below the header is a navigation bar with links: "Home", "Documentation", "Downloads", "News", and "Support". The main content area is divided into two columns. The left column is titled "WHAT IS PostGIS?" and contains text about PostGIS adding support for geographic objects to PostgreSQL, its certification by OGC, and its development by Refrations Research. The right column is titled "NEWS" and features an announcement for the "2007 FREE AND OPEN SOURCE SOFTWARE FOR GEOSPATIAL (FOSS4G) CONFERENCE" held in Victoria, Canada, from September 24-27, 2007. The announcement includes a call to action to "Save Money with Early Bird Registration".

PostGIS : Home

http://postgis.refractions.net/ Google

PostGIS : Home

PostGIS

Home Documentation Downloads News Support

[Home](#)

WHAT IS PostGIS?

PostGIS adds support for geographic objects to the [PostgreSQL](#) object-relational database. In effect, PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS), much like ESRI's SDE or Oracle's Spatial extension. PostGIS follows the [OpenGIS "Simple Features Specification for SQL"](#) and has been [certified as compliant](#) with the "Types and Functions" profile.

PostGIS has been developed by [Refrations Research](#) as a project in open source spatial database technology. PostGIS is released under the [GNU General Public License](#). We continue to develop PostGIS, and have added [user](#)

NEWS

[Free & Open Source Software for Geospatial Conference 2007](#)

September 24-27, 2007

[FOSS4G 2007](#) is the conference to attend to meet the people who make open source geospatial software go. Project leaders, corporate supporters, implementers, and users all get together to tell their stories, share some fun, and learn. The Conference runs from 24-27 September, 2007.

2007 FREE AND OPEN SOURCE SOFTWARE FOR GEOSPATIAL (FOSS4G) CONFERENCE

Victoria, Canada • Sept 24-25, 2007

Save Money with Early Bird Registration

Importing Shapefiles

- Convert the Shapefile to a SQL import script
 - ◆ `shp2sql -s 4269 st99_d00.shp us_states > us_states.sql`
- Create the database
 - ◆ `/usr/local/postgresql/bin/createdb g4wd`
- Now import the SQL
 - ◆ `psql -f us_states.sql -d g4wd`

Looking Around

- Log into PostgreSQL
 - ◆ `psql g4wd`
- Look for the new table
 - ◆ `\dt`
- Describe the table schema
 - ◆ `\d us_states`

Default (94,26)

Default

Default

```

g4wd=# \d us_states

```

Column	Type	Modifiers
gid	integer	not null default nextval('us_states_gid_seq'::regclass)
area	numeric	
perimeter	numeric	
st99_d00_	bigint	
st99_d00_i	bigint	
state	character varying(2)	
name	character varying(90)	
lsad	character varying(2)	
region	character varying(1)	
division	character varying(1)	
lsad_trans	character varying(50)	
the_geom	geometry	

```

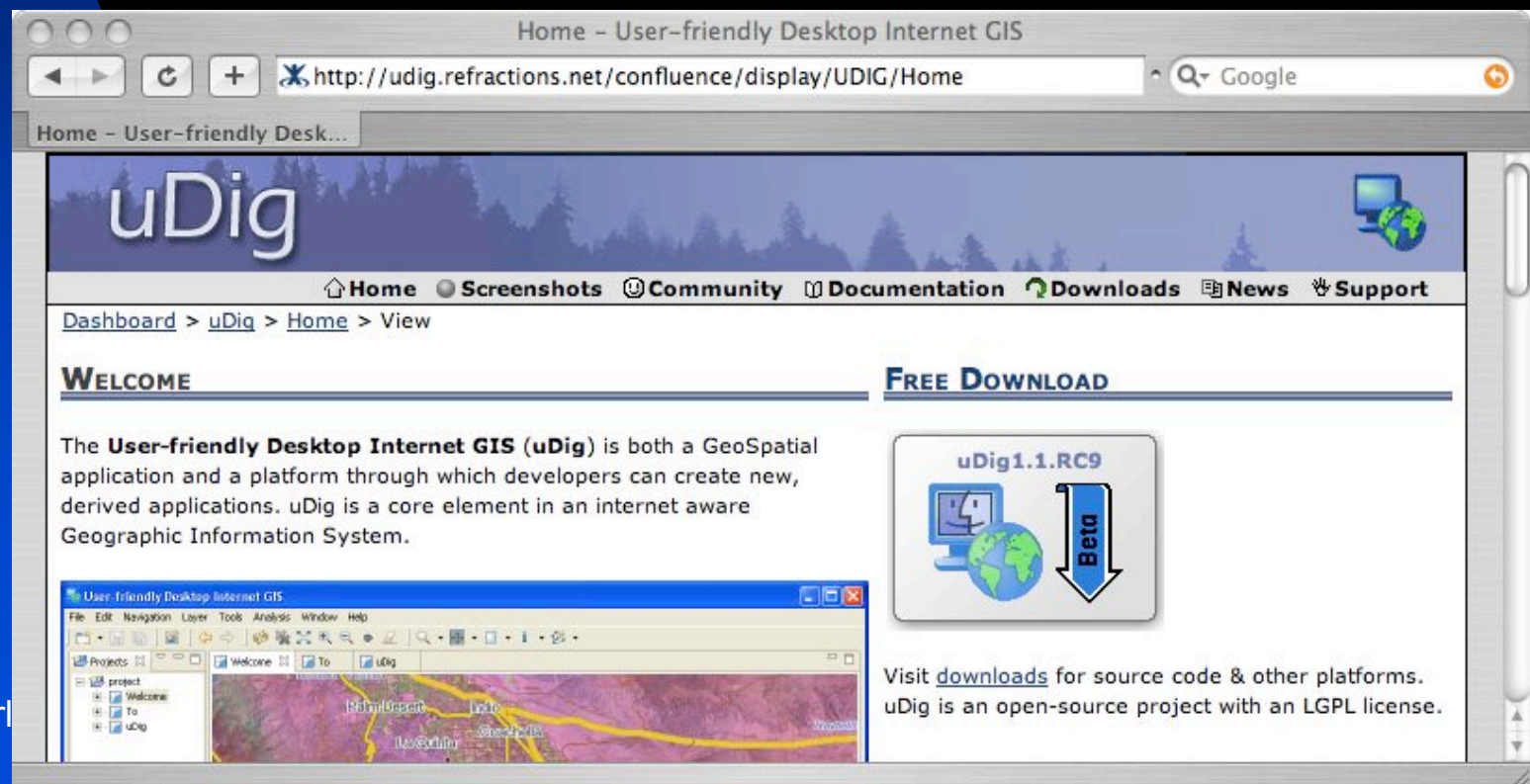
Indexes:
    "us_states_pkey" PRIMARY KEY, btree (gid)
Check constraints:
    "enforce_dims_the_geom" CHECK (ndims(the_geom) = 2)
    "enforce_geotype_the_geom" CHECK (geometrytype(the_geom) = 'MULTIPOLYGON'::text OR the_ge
IS NULL)
    "enforce_srid_the_geom" CHECK (srid(the_geom) = 4269)
g4wd=#

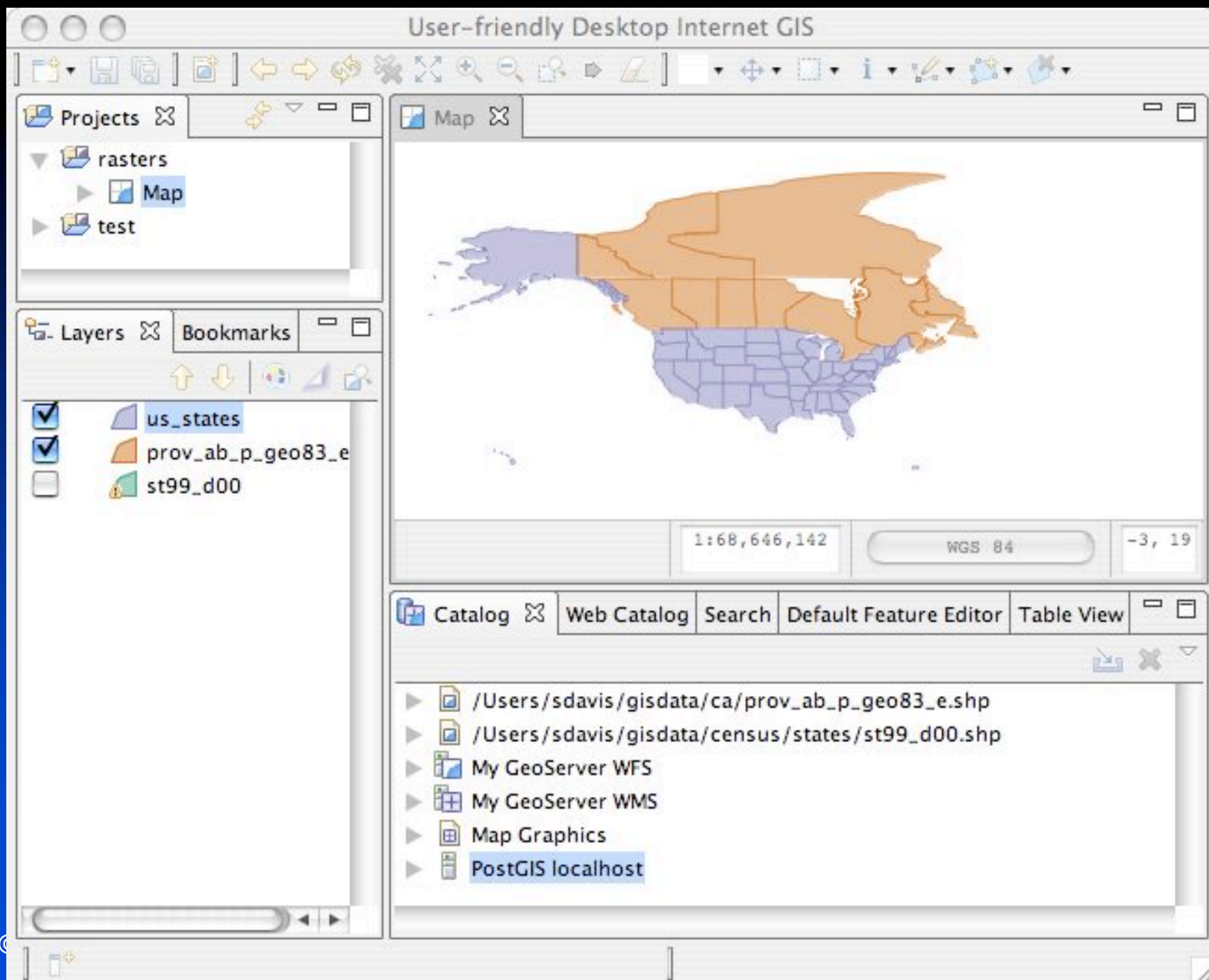
```


Visualizing the Data

- uDig

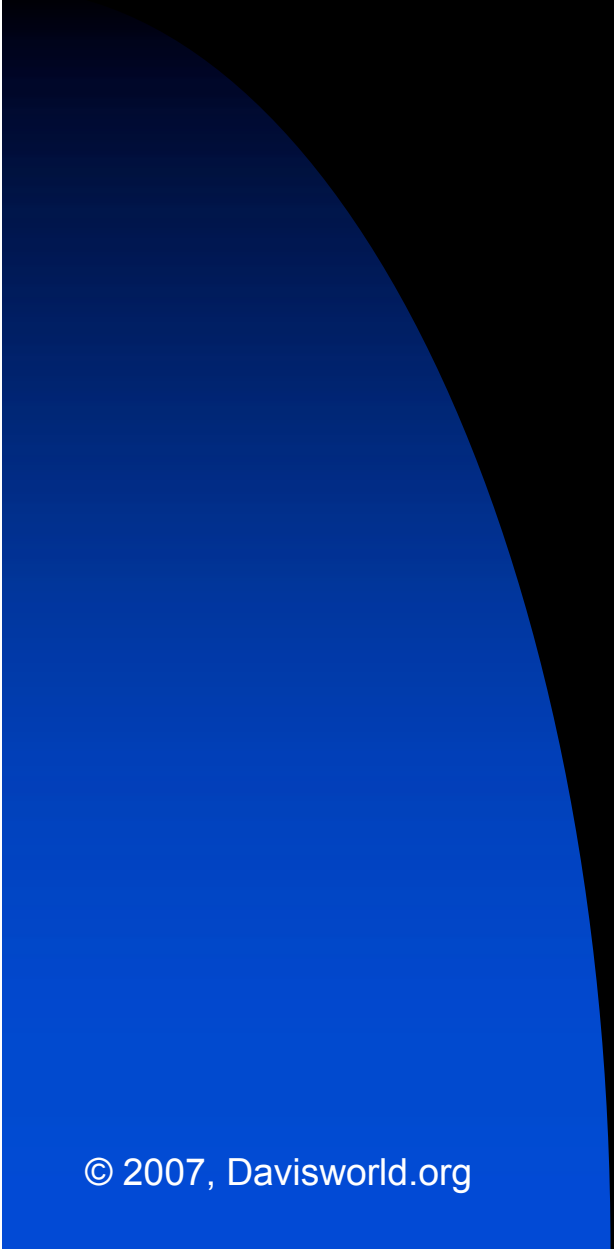
- ◆ <http://udig.refractions.net>







Act 4: Web Services

- 
- Google Maps is undeniably cool, but it is a proprietary interface
 - There is an open set of standards that allows you to do the same sort of thing...

Who is the OGC?

- The Open Geospatial Consortium
 - ◆ <http://www.opengis.org>

The Open Geospatial Consortium, Inc. (OGC) is a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services.



What does the OGC offer?

- Using the OGC interface, there are three basic things an application can ask:
 - ◆ What services do you offer?
 - ◆ What will the format of the output be?
 - ◆ Give me the data, please.

GetCapabilities

- The first thing a program will do is ask the OGC server what data layers it has to offer
 - ◆ This is called a GetCapabilities request

```
http://localhost:8888/geoserver/wfs?  
service=WFS&  
version=1.0.0&  
request=GetCapabilities
```

GetCapabilities

- And here is the response:

```
- <FeatureTypeList>
  - <Operations>
    <Query/>
  </Operations>
  - <FeatureType>
    <Name>wfs:co_cities_poly</Name>
    <SRS>EPSG:4326</SRS>
    <LatLongBoundingBox minx="-109.018116" miny="37.00000485324387" maxx="-1
  </FeatureType>
  - <FeatureType>
    <Name>wfs:co_counties_poly</Name>
    <SRS>EPSG:4326</SRS>
    <LatLongBoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
  </FeatureType>
  - <FeatureType>
    <Name>wfs:co_highways_line</Name>
    <SRS>EPSG:4326</SRS>
    <LatLongBoundingBox minx="-109.06008268030426" miny="36.9936887423584" n
  </FeatureType>
  - <FeatureType>
    <Name>wfs:us_states_poly</Name>
    <SRS>EPSG:4326</SRS>
    <LatLongBoundingBox minx="-179.14734" miny="17.884813" maxx="179.77847" 1
  </FeatureType>
</FeatureTypeList>
```

DescribeFeatureType

- Once you find a layer that looks interesting, you can ask the service to describe it:

```
http://localhost:8888/geoserver/wfs?  
version=1.0.0&  
service=WFS&  
request=DescribeFeatureType&  
typename=us_states_poly
```

DescribeFeatureType

- And here is the response:

```
<xsd:schema targetNamespace="http://www.ionicssoft.com/wfs" elementFormDefault="qualified" version="0.1">
  <xsd:import namespace="http://www.opengis.net/gml" schemaLocation="http://www.opengis.net/namespaces/gml/core/feature.xsd"/>
  <xsd:element name="us_states_poly" substitutionGroup="gml:_Feature" type="wfs:us_states_poly"> </xsd:element>
  - <xsd:complexType name="us_states_poly">
    - <xsd:complexContent>
      - <xsd:extension base="gml:AbstractFeatureType">
        - <xsd:sequence>
          <xsd:element name="AREA" minOccurs="0" nillable="true" type="xsd:double"> </xsd:element>
          <xsd:element name="PERIMETER" minOccurs="0" nillable="true" type="xsd:double"> </xsd:element>
          <xsd:element name="ST99_D00_" minOccurs="0" nillable="true" type="xsd:int"> </xsd:element>
          <xsd:element name="ST99_D00_I" minOccurs="0" nillable="true" type="xsd:int"> </xsd:element>
          <xsd:element name="STATE" minOccurs="0" nillable="true" type="xsd:string"> </xsd:element>
          <xsd:element name="NAME" minOccurs="0" nillable="true" type="xsd:string"> </xsd:element>
          <xsd:element name="LSAD" minOccurs="0" nillable="true" type="xsd:string"> </xsd:element>
          <xsd:element name="REGION" minOccurs="0" nillable="true" type="xsd:string"> </xsd:element>
          <xsd:element name="DIVISION" minOccurs="0" nillable="true" type="xsd:string"> </xsd:element>
          <xsd:element name="LSAD_TRANS" minOccurs="0" nillable="true" type="xsd:string"> </xsd:element>
          <xsd:element name="GEOMETRY" minOccurs="0" nillable="true" type="gml:PolygonPropertyType"> </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```


GetFeature

- Finally, you can request the data layer:

```
http://localhost:8888/geoserver/wfs?  
version=1.0.0&  
service=WFS&  
request=GetFeature&  
typename=us_states_poly
```

GetFeature

- And here is the response:

- ◆ This is called GML (Geometry Markup Language)

```
<gml:featureMember>
- <au1:us_states_poly fid="us_states_poly.165">
  <au1:AREA>2.803919812051006</au1:AREA>
  <au1:PERIMETER>2.201919233137796</au1:PERIMETER>
  <au1:ST99_D00_>167</au1:ST99_D00_>
  <au1:ST99_D00_I>166</au1:ST99_D00_I>
  <au1:STATE>08</au1:STATE>
  <au1:NAME>Colorado</au1:NAME>
  <au1:LSAD>01</au1:LSAD>
  <au1:REGION>4</au1:REGION>
  <au1:DIVISION>8</au1:DIVISION>
  <au1:LSAD_TRANS/>
- <au1:GEOMETRY>
  - <gml:Polygon srsName="EPSG:4326">
    - <gml:outerBoundaryIs>
      - <gml:LinearRing srsName="EPSG:4326">
        - <gml:coordinates>
          -107.918421,41.002036 -107.6913358243142,41.002104
          -107.573624,41.00231513722544 -107.5215053632723
          -107.31779446240112,41.00296721336712 -107.305312
          -107.30252872253156,41.002934686503345 -107.24119
          -107.1160809891758,41.00313681928827 -107.000606
          -106.453859,41.002057 -106.439563,41.001978 -106.43
```



(...Astoundingly boring, isn't it?)

Data vs Pretty Pixels

- Data services:
 - ◆ WFS (Web Feature Service)
- Portrayal services:
 - ◆ WMS (Web Mapping Service)

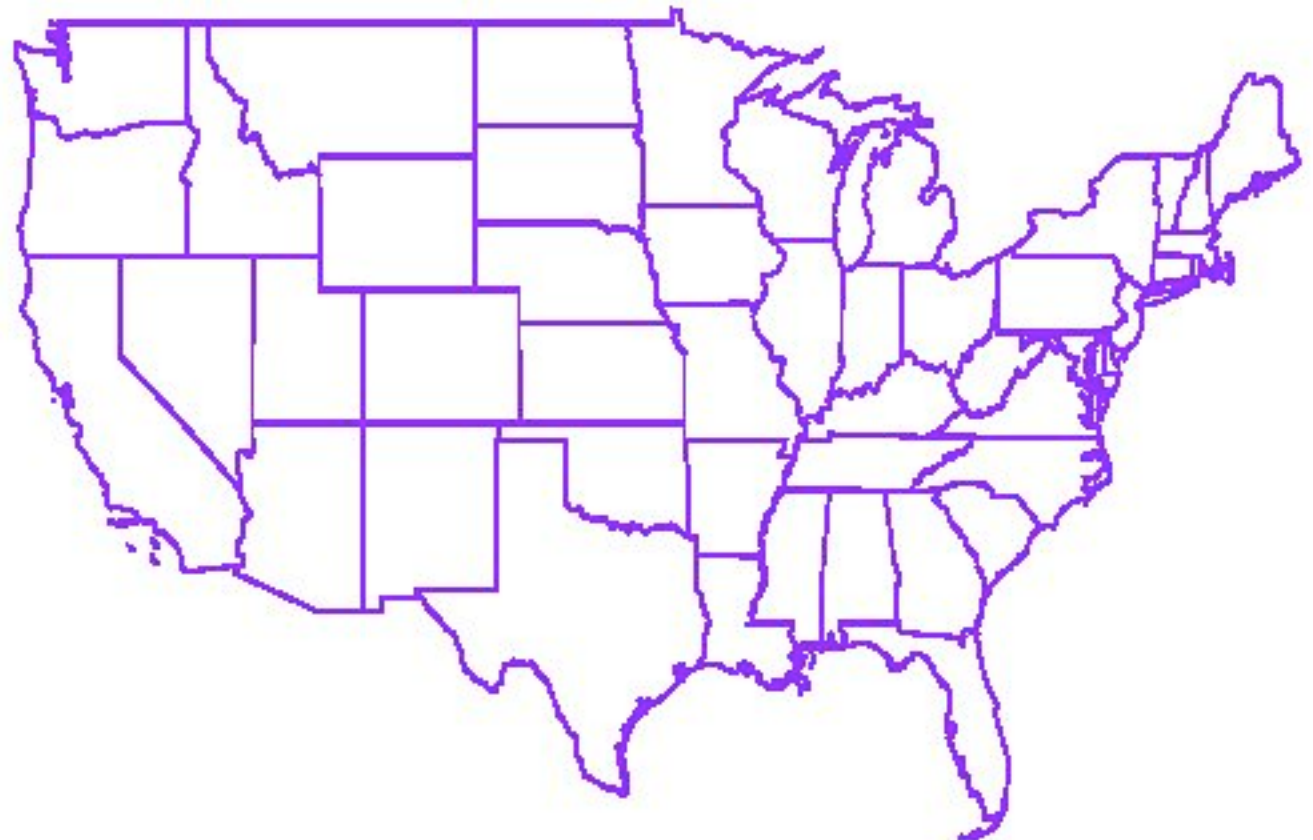
Web Mapping Service

- GetMap request:

```
http://localhost:8888/geoserver/wms?  
VERSION=1.1.1&  
REQUEST=GetMap&  
SRS=EPSG:4326&  
BBOX=-126,20,-66,52&  
WIDTH=500&  
HEIGHT=500&  
LAYERS=us_states_poly&  
STYLES=&  
FORMAT=image/png&  
BGCOLOR=0xffffffff&  
TRANSPARENT=FALSE&  
EXCEPTIONS=application/vnd.ogc.se_inimage
```

Web Mapping Service

- ...returns a single layer



- Since the OGC standards are open, anyone can implement them:

OGC® "Making location count"

OGC® Home | OGC Network™ | OGC User™ | OGC Forum

About Standards Programs Press Events Regions Resources

Search

Resources

- Compliance Testing
 - Download Test Suites
- Registered Products
 - All Registered Products
 - Implementation Statistics
 - View By Specification
 - Compliant Products
 - Implementing Products
 - Register Your Products
- OGC Cookbooks
- Demonstrations
- Email Lists
- FAQs
 - Purpose & Structure

HOME » RESOURCE

All Registered Products

- Jump to organization listing

52°North					Top
Product Name	OGC Spec	Type	Contact	Date	
52N Sensor Observation Service 2-01-00	SOS 0.1.4	Server	Walkowski, Alexander C.	2006-05-16	

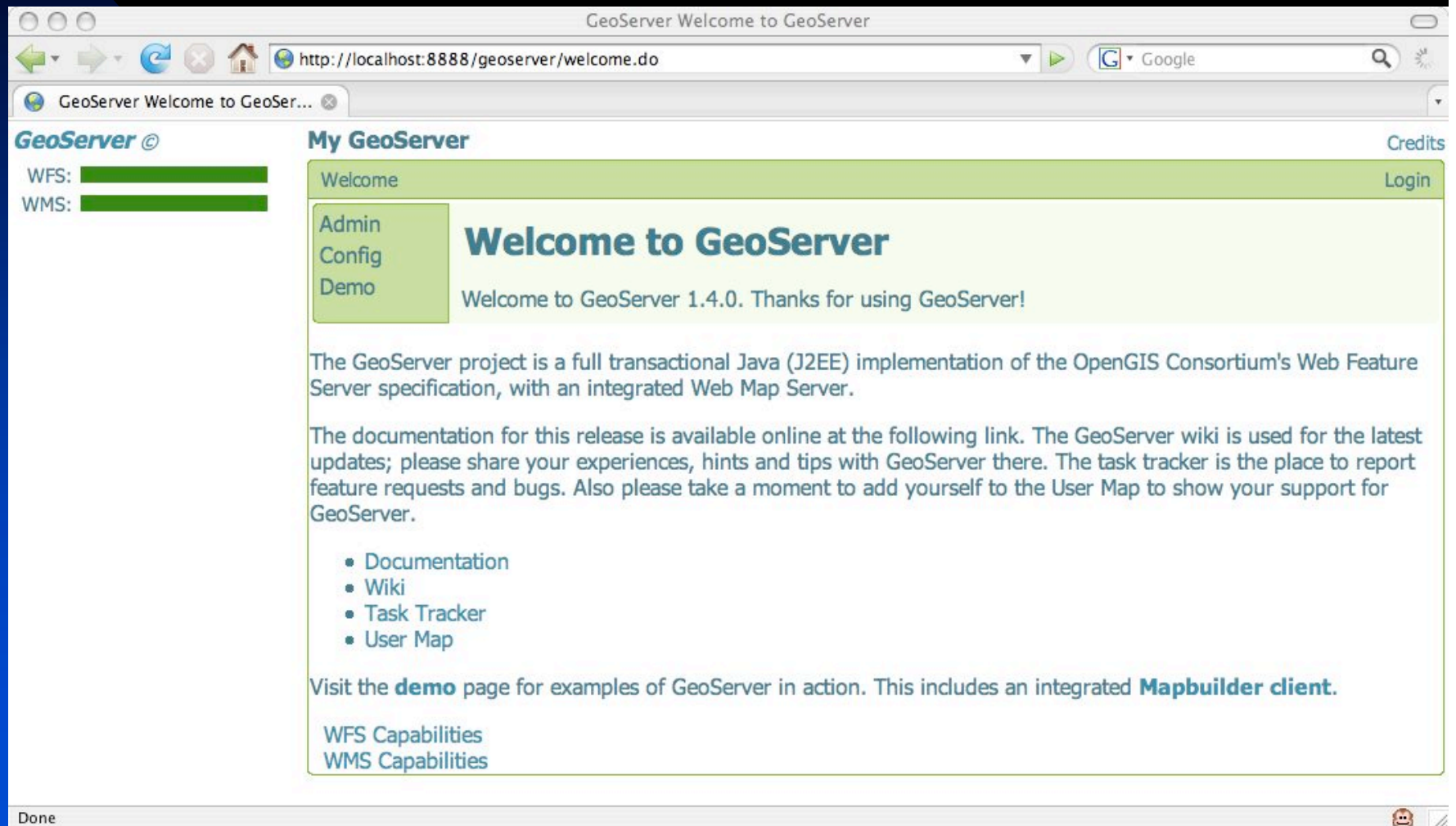
ABACO srl					Top
Product Name	OGC Spec	Type	Contact	Date	
	WMS 1.1.1				



- Java-based
 - ◆ <http://docs.codehaus.org/display/GEOS/Home>
- Runs in a servlet container
- Provides OGC (Open Geospatial Consortium) web services
 - ◆ WMS
 - ◆ WFS



Welcome Screen



GeoServer DataStore Editor

http://localhost:8888/geoserver/config/data/storeNewSubmit.do

GeoServer DataStore Editor

GeoServer ©

Data:

GeoServer Jan 14, 1:23 PM

Configuration Jan 14, 1:23 PM

XML Jan 14, 1:23 PM

Apply Save Load

My GeoServer

Credits Contact: null

Welcome | Config | Data | Stores | Edit

Logout

DataStore Editor

Edit a source of spatial information

DataStore ID: local_postgis

Enabled: ☒

Namespace:

Description:

* host:

* port:

schema:

* database:

* user:

passwd:

wkb enabled:

loose bbox:

Submit Reset

* = required field

Done

MapBuilder

- Pure JavaScript web client for OGC
 - ◆ <http://mapbuilder.sourceforge.net>
- Included with GeoServer

Mix 'n Match

- Since OGC is a standard interface, you can mix data from across the web with your own local layers

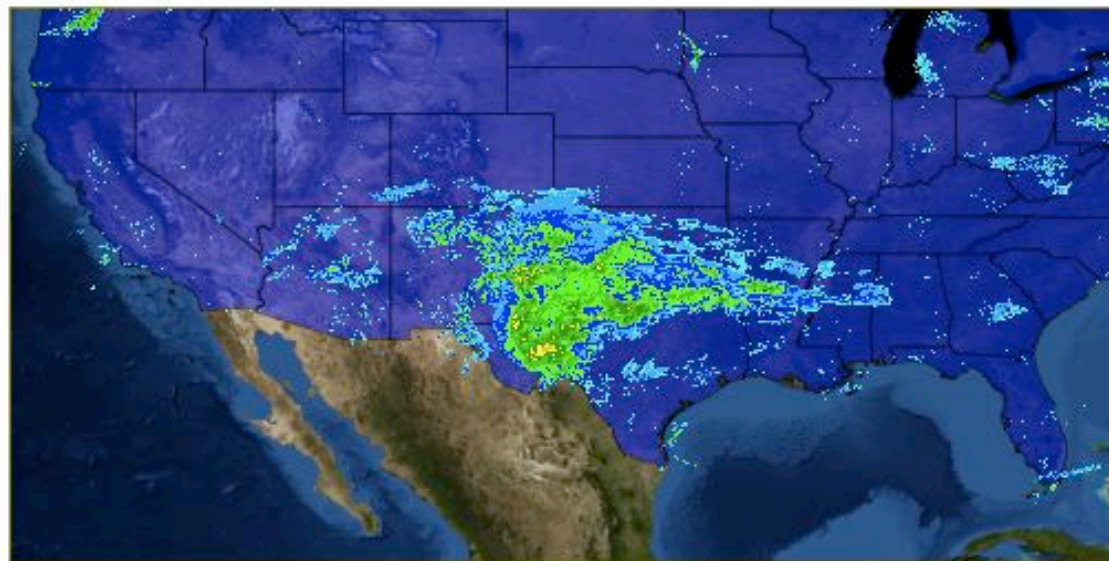
NASA + Live Weather

```
12 <LayerList>
13   <Layer queryable="1" hidden="0">
14     <Server service="OGC:WMS" version="1.1.1" title="blue marble">
15       <OnlineResource xlink:type="simple" xlink:href="http://wms.jpl.nasa.gov/wms.cgi?"/>
16     </Server>
17     <Name>BMNG</Name>
18     <Title>Blue Marble</Title>
19     <SRS>EPSG:4326</SRS>
20     <FormatList><Format current="1">image/png</Format></FormatList>
21   </Layer>
22   <Layer queryable="1" hidden="0">
23     <Server service="OGC:WMS" version="1.1.1" title="weather">
24       <OnlineResource xlink:type="simple" xlink:href="http://mesonet.agron.iastate.edu/cgi-bin/wms/nexrad/n0r.cgi"/>
25     </Server>
26     <Name>nexrad-n0r-m45m</Name>
27     <Title>Weather</Title>
28     <SRS>EPSG:4326</SRS>
29     <FormatList><Format current="1">image/png</Format></FormatList>
30   </Layer>
31   <Layer queryable="1" hidden="0">
32     <Server service="OGC:WMS" version="1.1.1" title="g4wd:st99_d00 Preview">
33       <OnlineResource xlink:type="simple" xlink:href="../wms"/>
34     </Server>
35     <Name>g4wd:st99_d00</Name>
36     <Title>US</Title>
37     <SRS>EPSG:4326</SRS>
38     <FormatList><Format current="1">image/png</Format></FormatList>
39   </Layer>
```

GeoServer/Map Builder g4wd:st99_d00 preview



lon: lat:



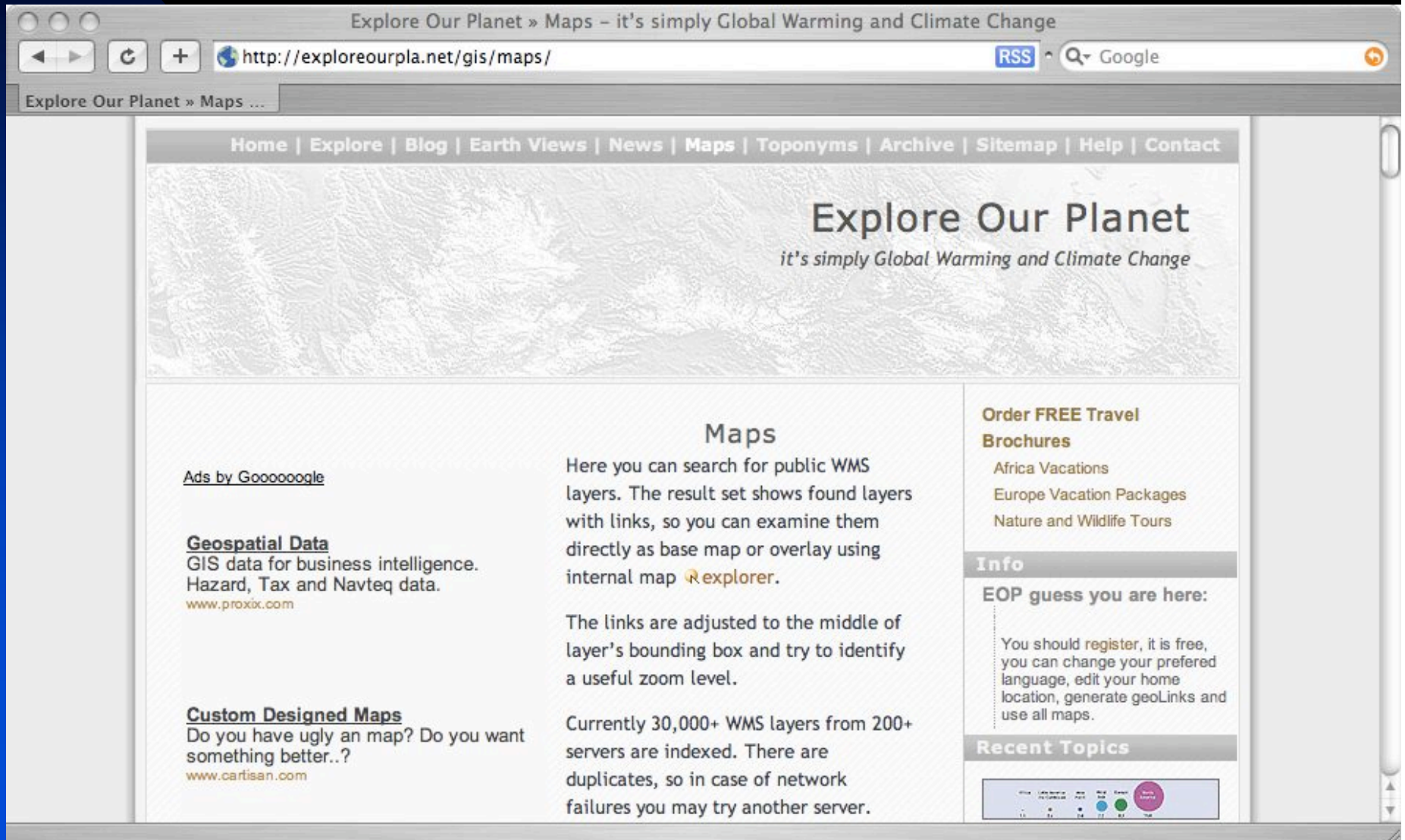
- ☒ ⓘ Weather
- ☒ ⓘ Canada
- ☒ ⓘ US
- ☒ ⓘ Blue Marble

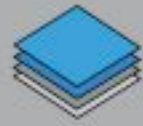
scale 1:37,702,570

Powered by [Community Map Builder](#)



ExploreOurPla.net

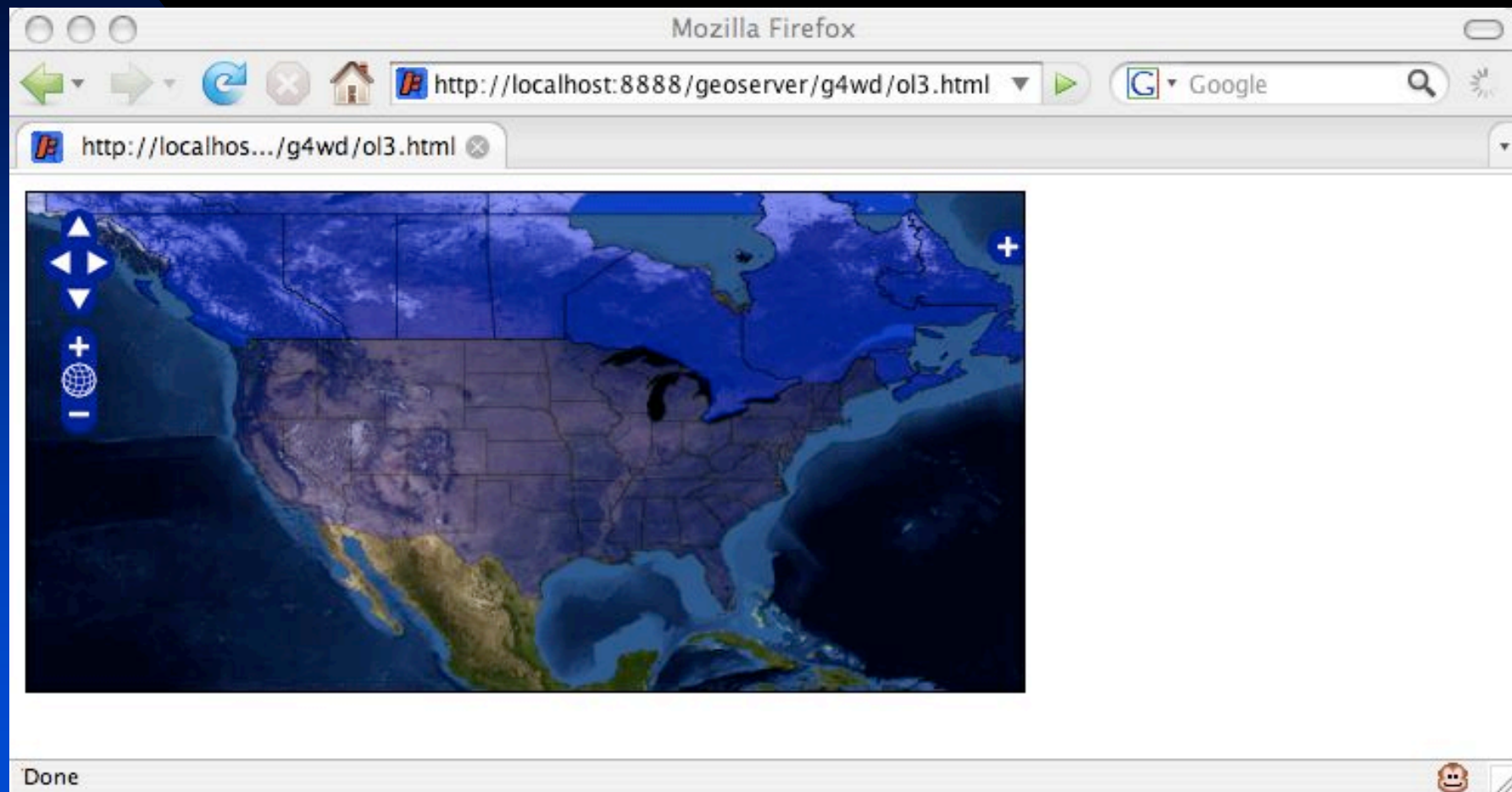




OpenLayers

- Pure JavaScript web client
 - ◆ <http://www.openlayers.org>
- In addition to supporting OGC layers
 - ◆ Google Maps
 - ◆ Yahoo Maps
 - ◆ MS Virtual Earth (Local Live)
 - ◆ WorldWind

One Map...



...One File

```
10 <script src="http://www.openlayers.org/api/OpenLayers.js"></script>
11 <script type="text/javascript">
12     //NOTE: geographic center of the US
13     var lon = -98.583333;
14     var lat = 39.833333;
15     var zoom = 3;
16     var map, us, canada, blueMarble;
17
18     function init(){
19         map = new OpenLayers.Map( $('map') );
20         blueMarble = new OpenLayers.Layer.WMS( "Blue Marble",
21         . "http://wms.jpl.nasa.gov/wms.cgi?", {layers: 'BMNG', format: 'image/png'},
22         . {isBaseLayer:true});
23         map.addLayer(blueMarble);
24
25         us = new OpenLayers.Layer.WMS( "US", "http://localhost:8888/geoserver/wms?",
26         . {layers: 'g4wd:st99_d00', format: 'image/png', transparent: true, {isBaseLayer:false,
27         . opacity:0.5} });
28         map.addLayer(us);
29
30         canada = new OpenLayers.Layer.WMS( "Canada",
31         . "http://localhost:8888/geoserver/wms?", {layers: 'g4wd:prov_ab_p_geo83_e', format:
32         . 'image/png', transparent: true, {isBaseLayer:false, opacity:1.0} });
33         map.addLayer(canada);
34
35         map.setCenter(new OpenLayers.LonLat(lon, lat), zoom);
36         map.addControl( new OpenLayers.Control.LayerSwitcher() );
37     }
38 </script>
```

Conclusion

- You now know:
 - ◆ Vector vs. Raster data
 - ★ Shapefiles
 - ★ Free sources of data and viewers
 - ◆ Projections
 - ★ GDAL and OGR2OGR
 - ◆ Spatial Databases
 - ★ PostgreSQL + PostGIS
 - ◆ Web Services
 - ★ OGC
 - ★ GeoServer
 - ★ MapBuilder
 - ★ OpenLayers

Conclusion

- Thanks for your time!
 - ◆ Questions?
 - ◆ Email:
 - ★ scottdavis99@yahoo.com
 - ◆ Download slides:
 - ★ <http://www.davisworld.org/presentations>