

Getting More out of Your Maps

Georeferencing and Optimizing Digitized Maps for Use in Web Applications

Megan Slemons and Jay Varner, Emory University
March 7, 2016

Who we are



Megan Slemons

GIS Librarian in the Emory Center for Digital
Scholarship

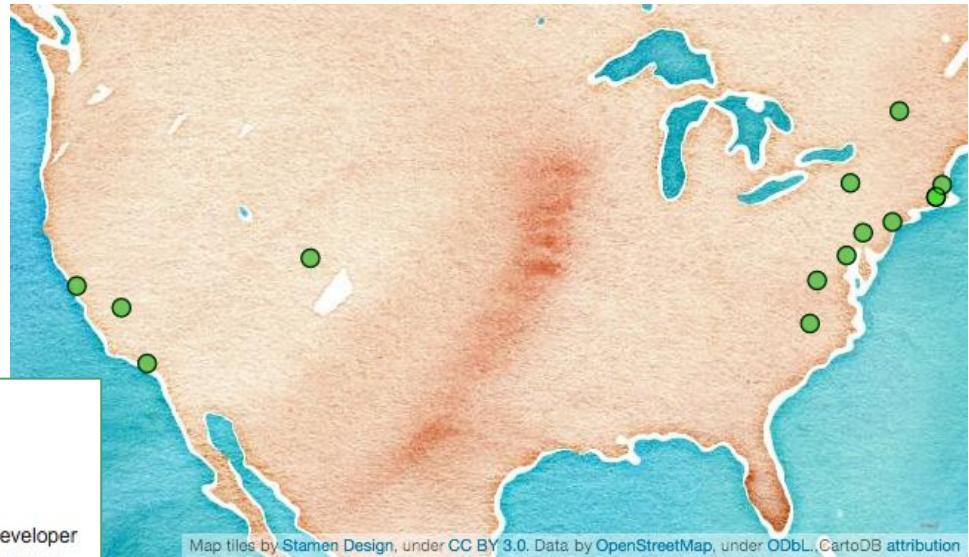
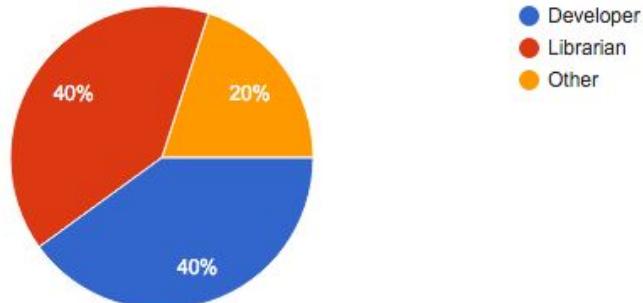


Jay Varner @jayvarner

Senior Software Engineer for Emory University
Libraries

Who are you?

What is your primary role? (10 responses)



The Problem

This...



Held by: NYU

Documentation at: <http://hdl.handle.net/2451/34189>



Leaflet | © OpenStreetMap contributors, © Ca

And also this...



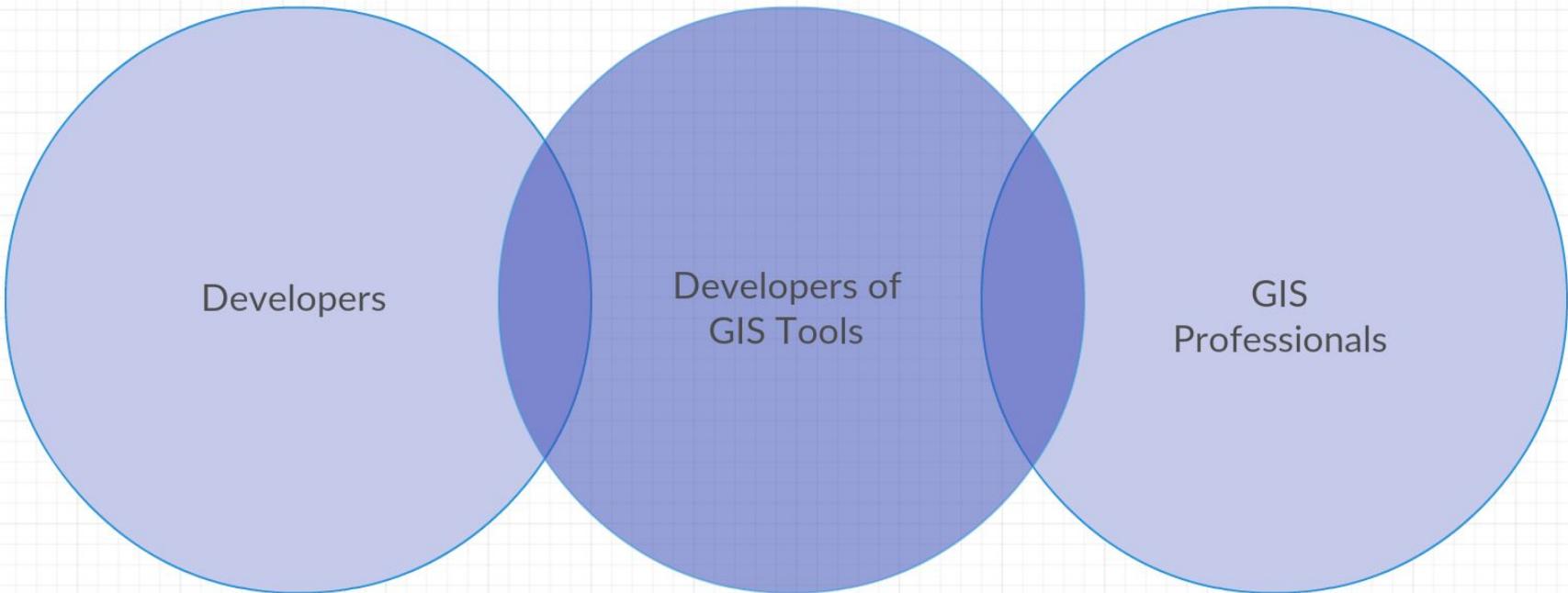
Held by Harvard

More details at <http://opengeometadata.stanford.edu/metadata/org.opengeoporta>



Working with GIS Software

The Mutual Knowledge Gap



Georeferencing

What is georeferencing?

“Georeferencing” means assigning a known location in physical space to something to something so that it can be analyzed and used with other geographic data.

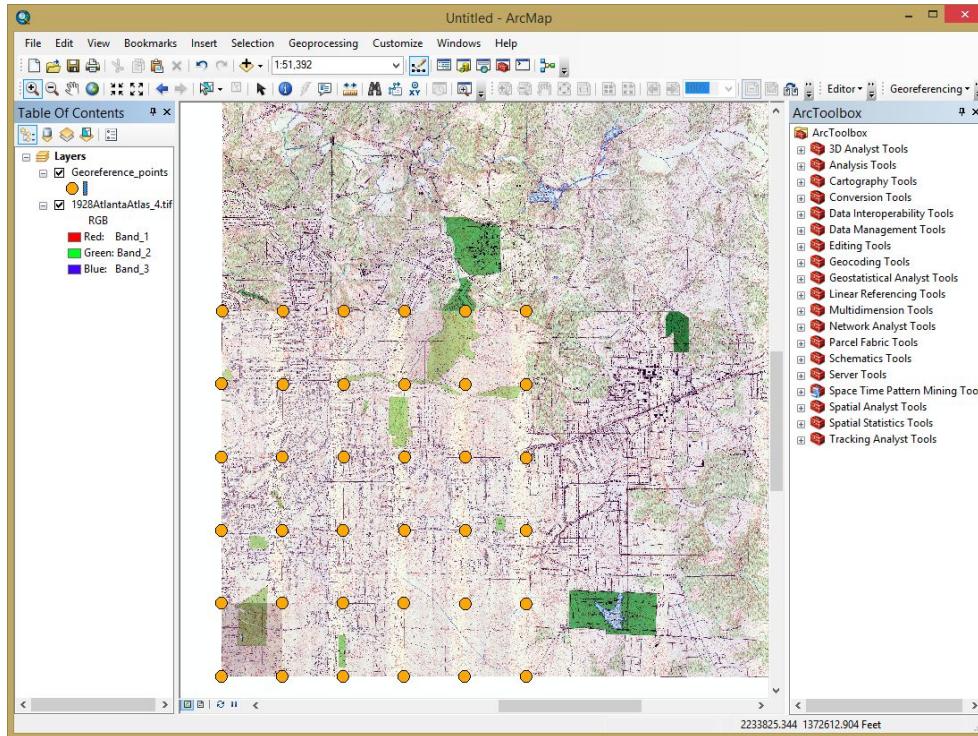
There are 2 main approaches:

1. When you have coordinates written on the map/image.
2. When there are no coordinates and you have to use features on the map compared to another already-georeferenced layer.

In my opinion, ArcGIS is the best option for georeferencing. However,

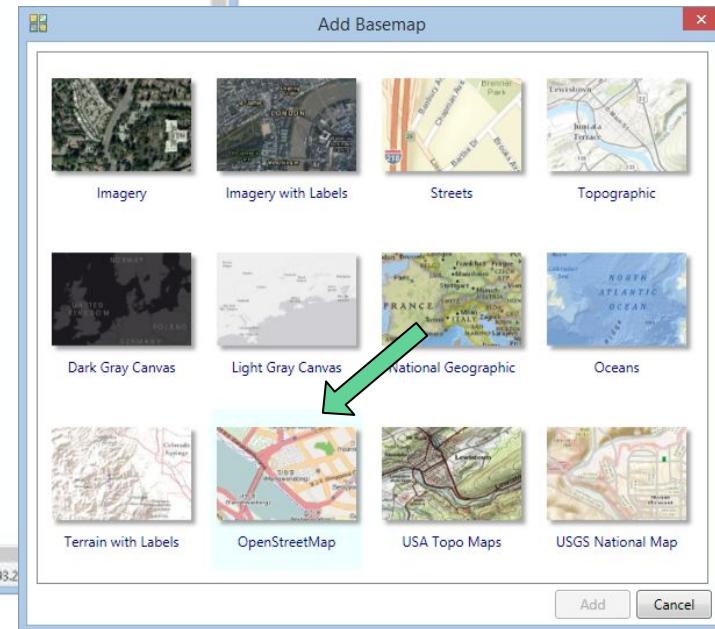
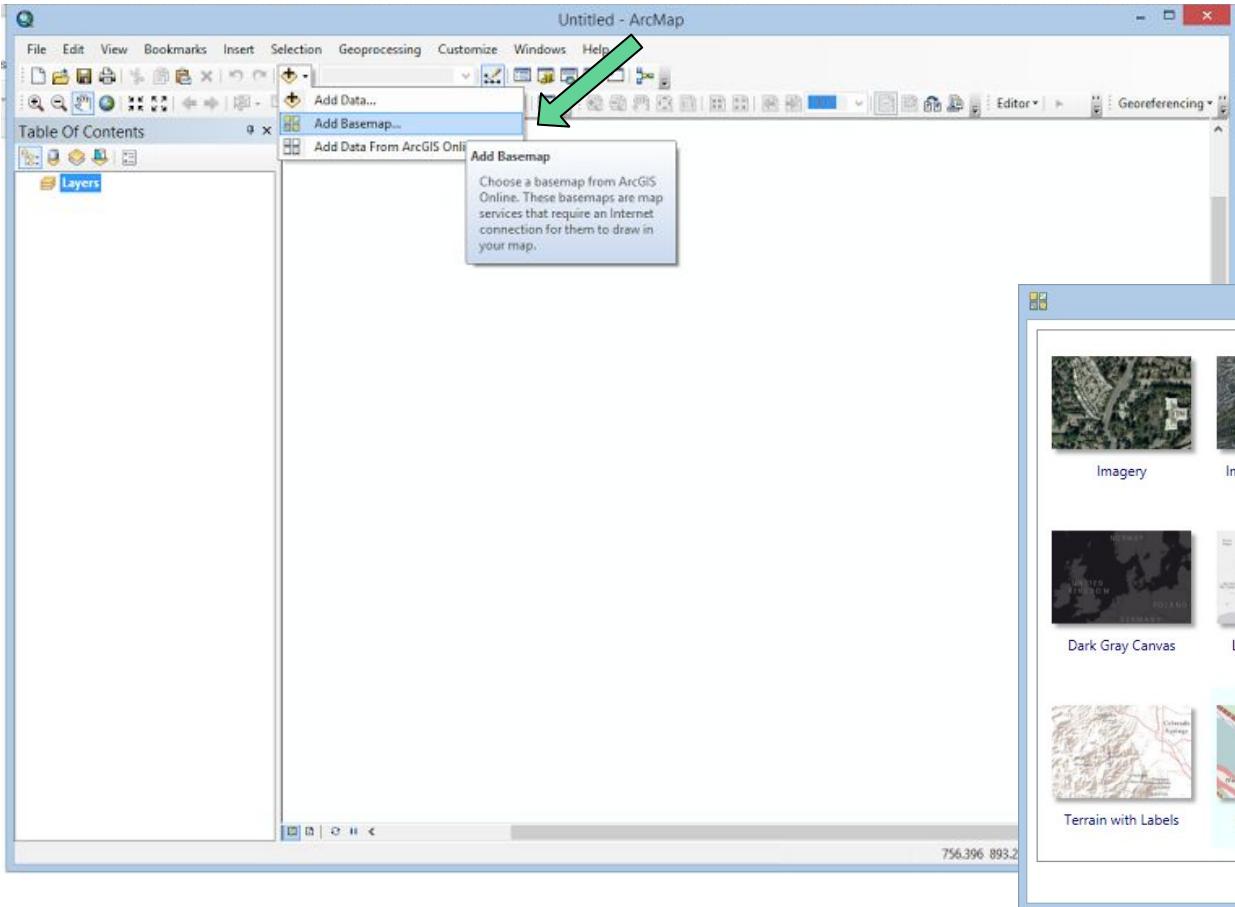
- it's Windows-only
- it's expensive (have to have an institutional license)

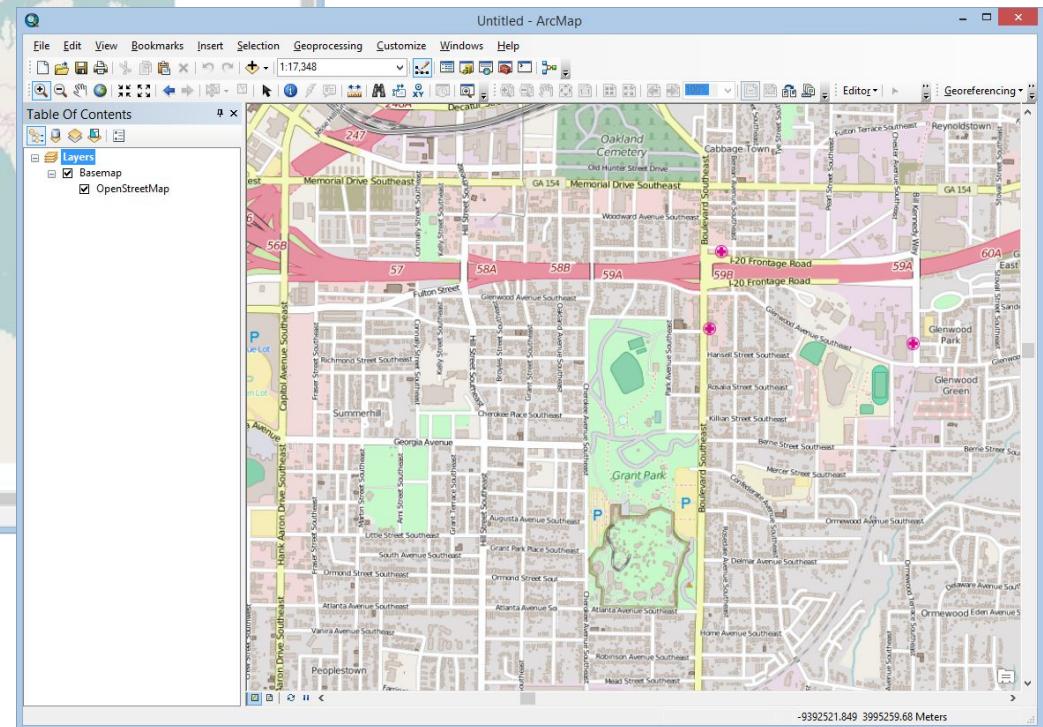
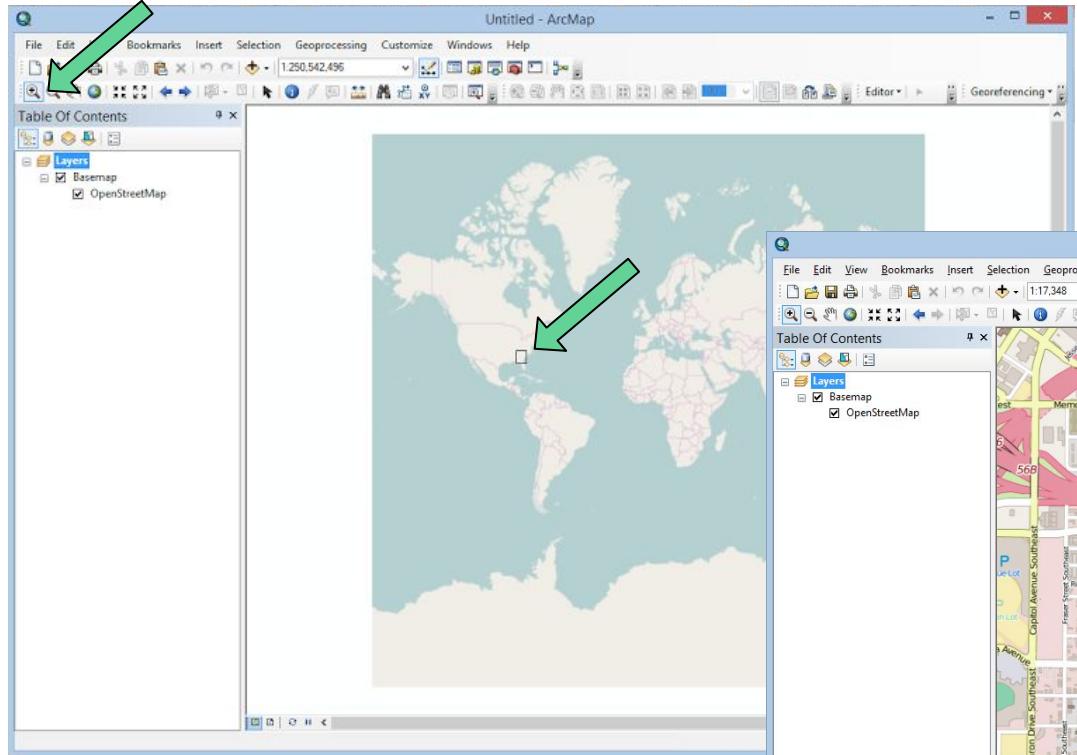
Using ArcGIS

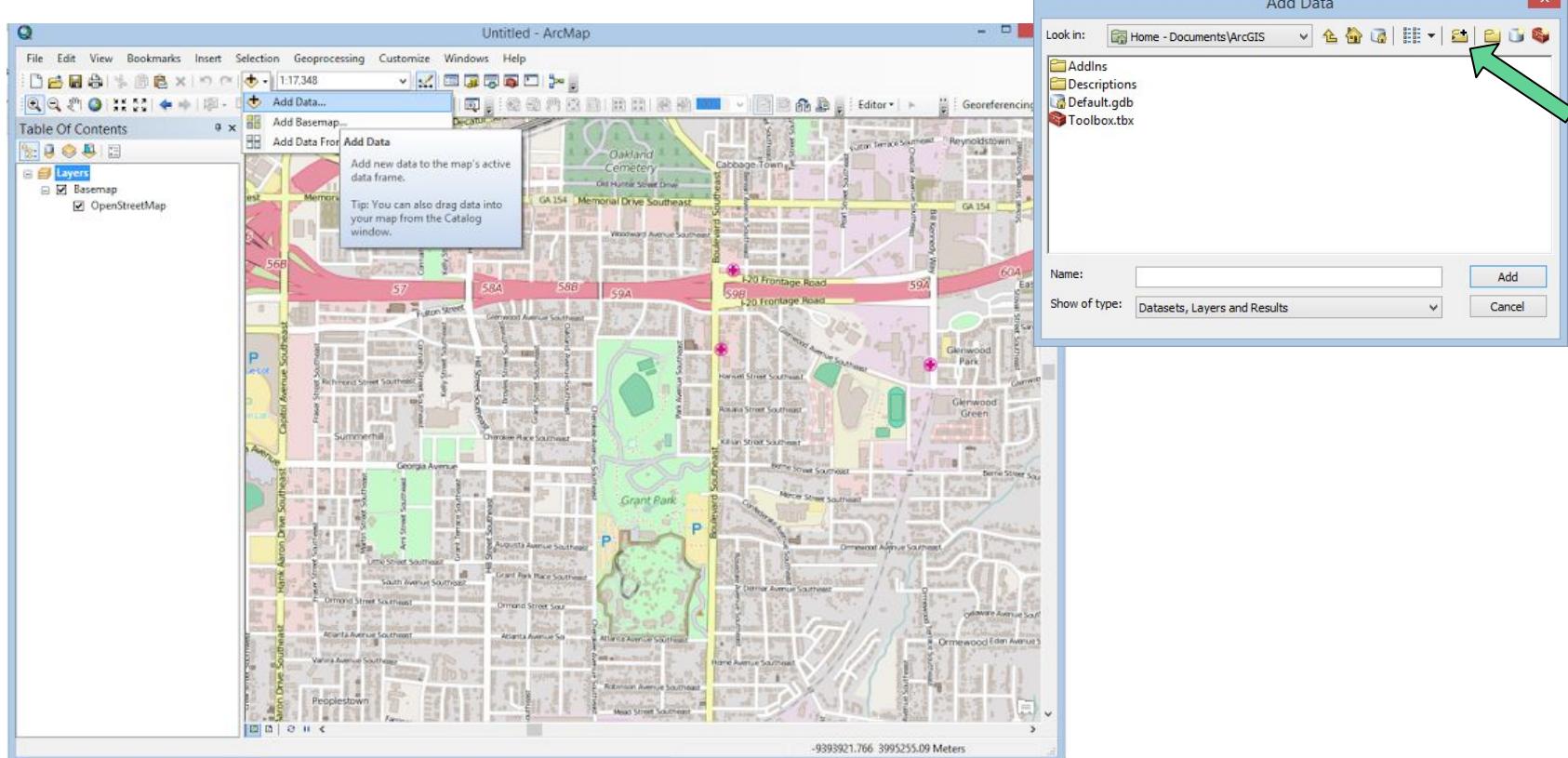


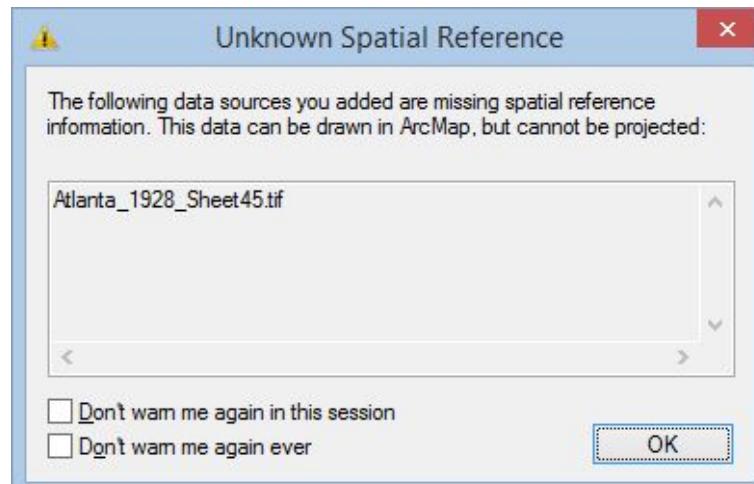
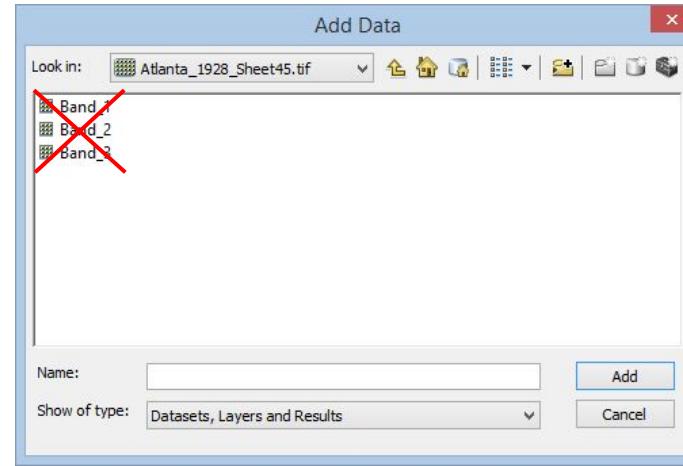
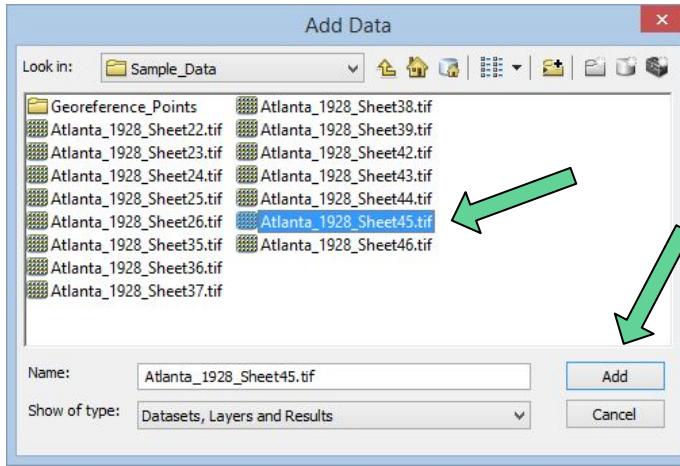
Basic steps to georeference when you have written coordinates on the map:

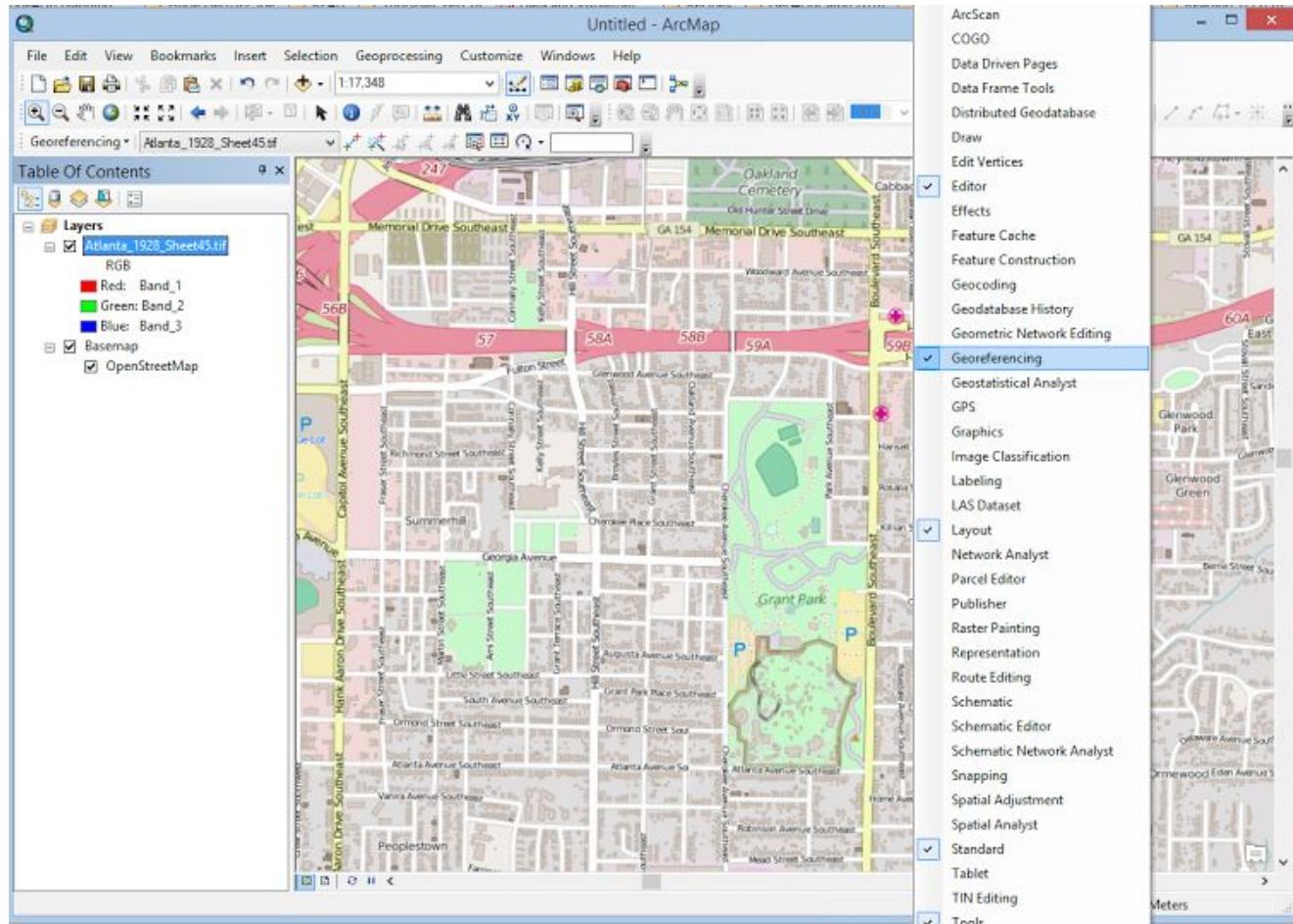
1. (optional) Crop maps to “collarless” version with no frame - just the map (Photoshop is a great tool for this)
2. Create a table of the 4 corner coordinates of each map
3. If necessary, convert each DMS coord pair to decimal degrees using online converter
4. Bring table in as event layer
5. Convert to a point shapefile (will create a dot grid)
6. Use the Georeferencing toolbar to match image corners to points

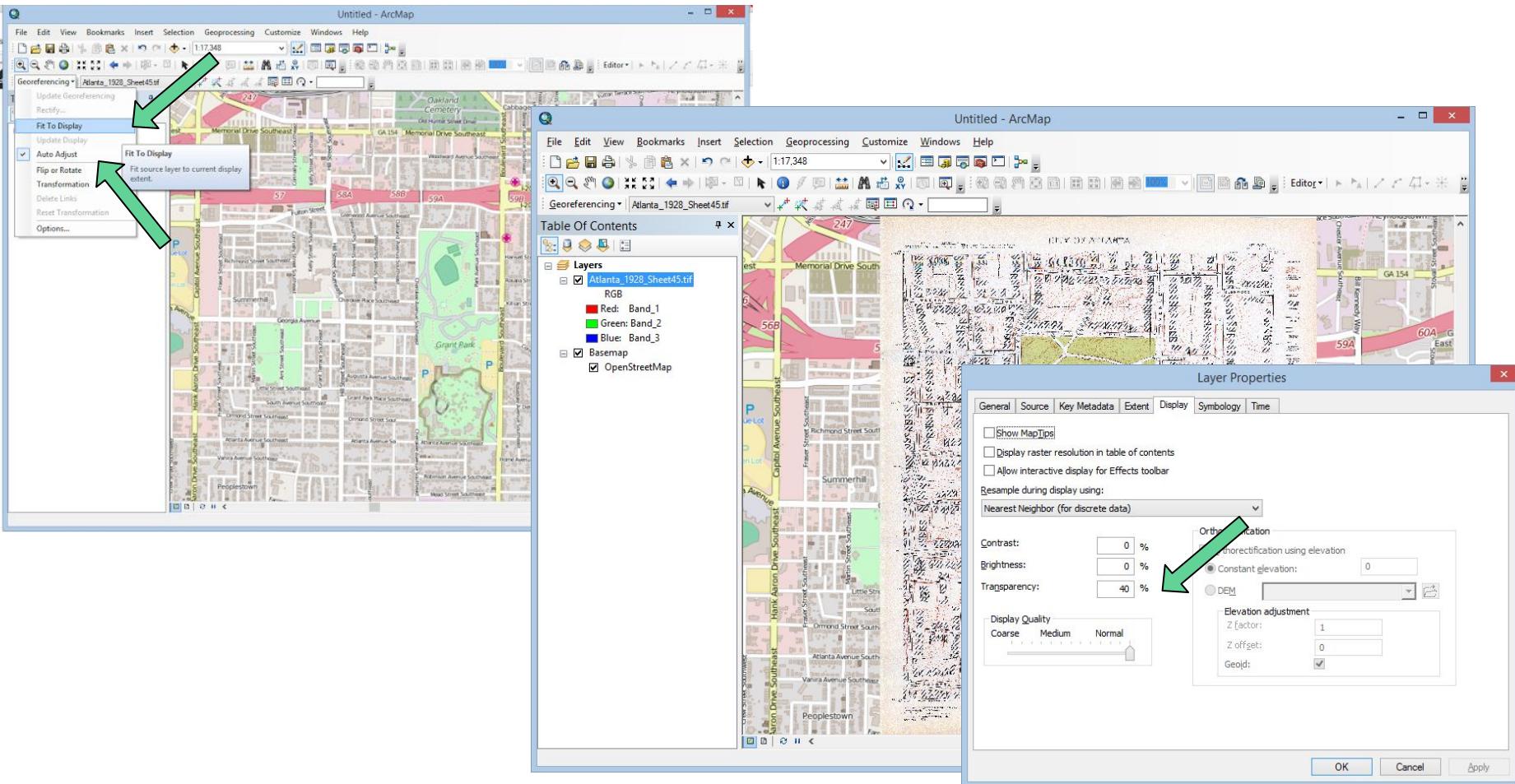


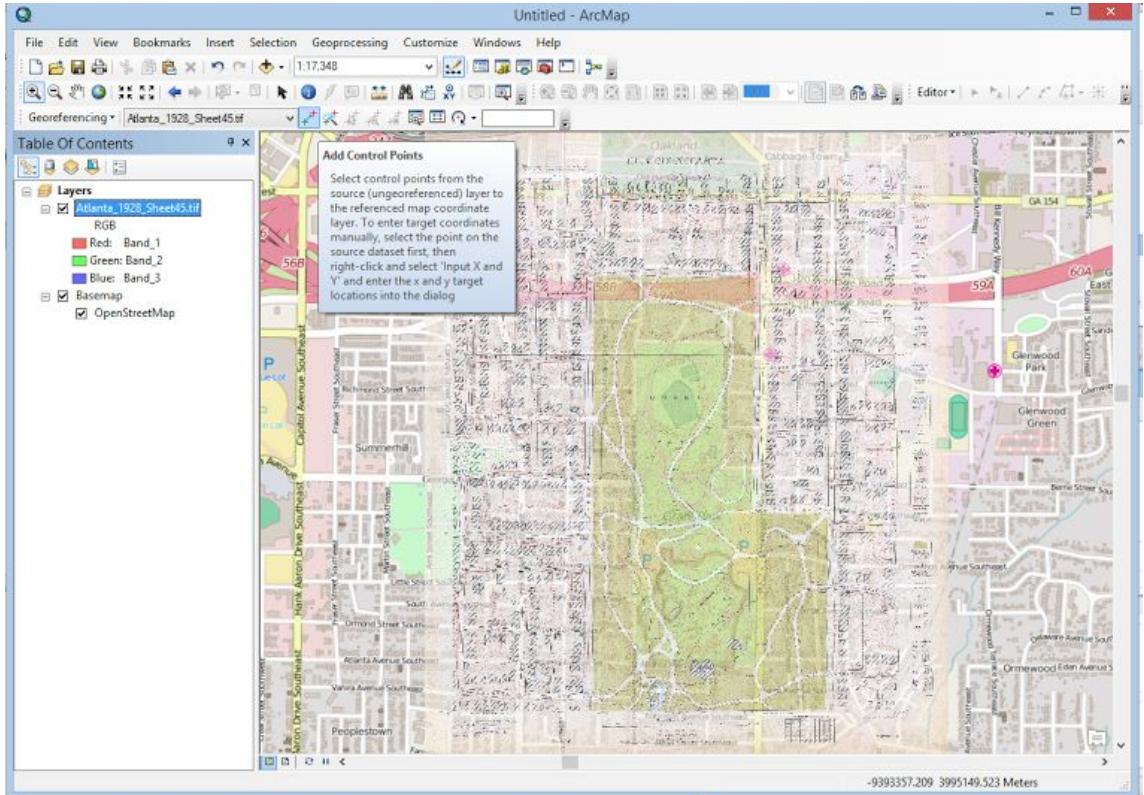


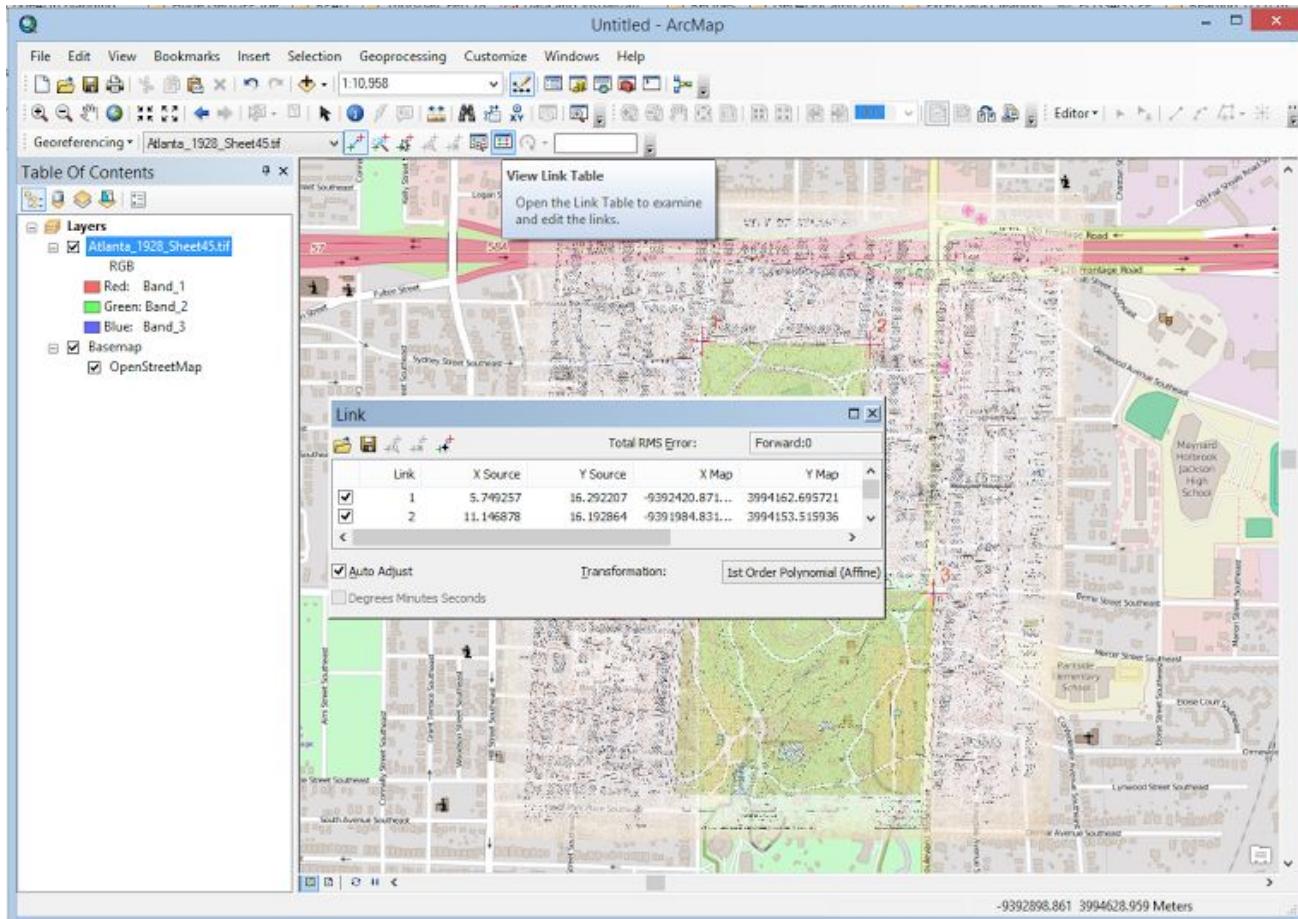


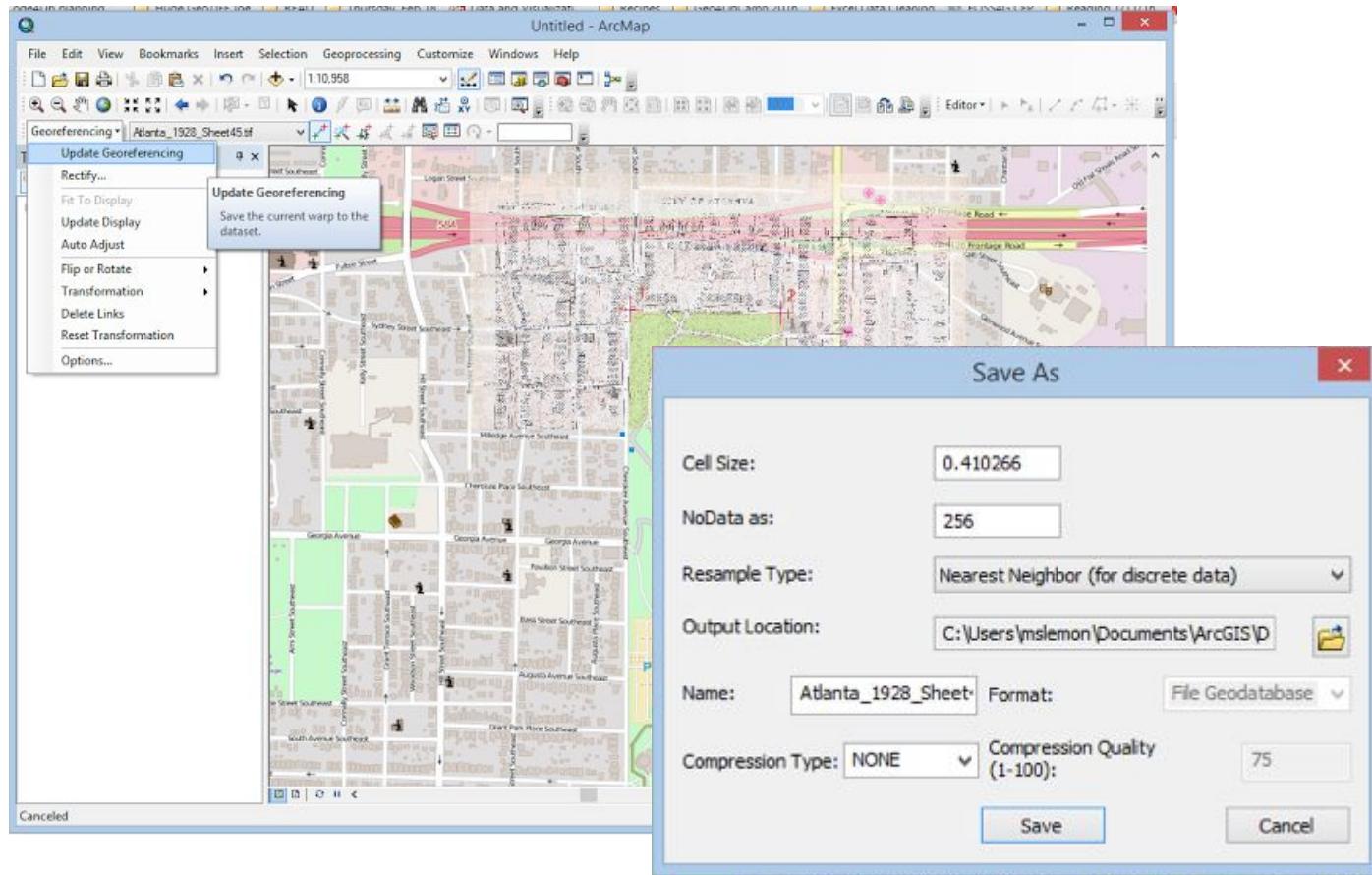










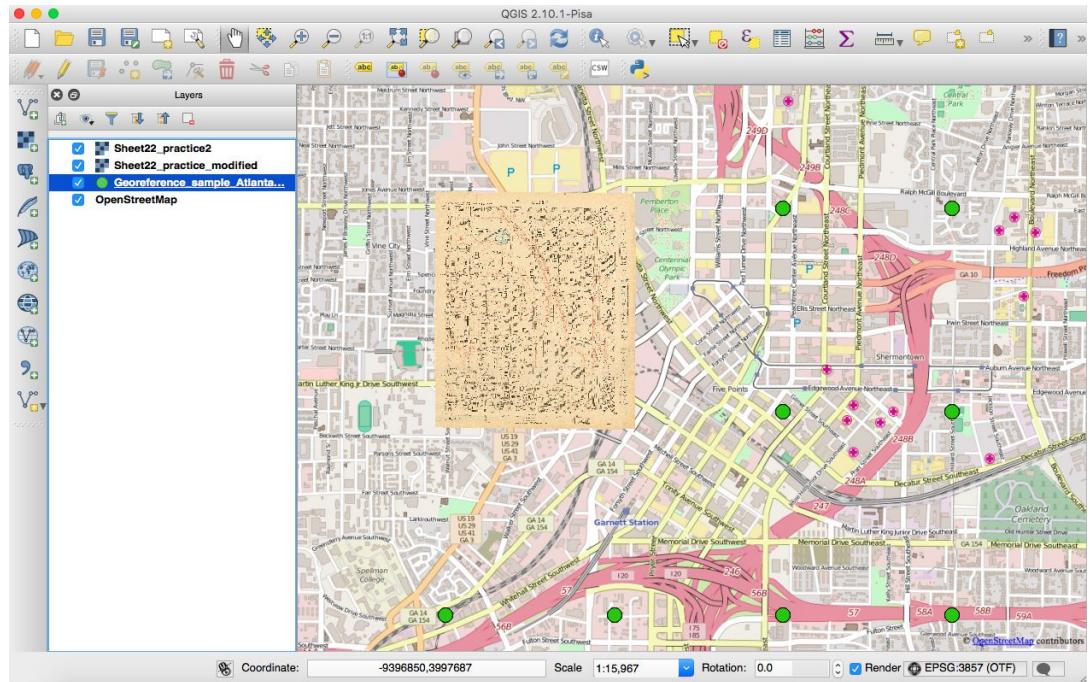


Using QGIS

We'll do a hands-on demo for this part.

Once you have QGIS installed, you'll need to activate the Georeferencer and OpenLayers plugins.

The grid we are georeferencing to is in NAD_1983_StatePlane_Georgia_West_FIPS_1002_Feet(US), or EPSG 2240, as this is the most accurate coordinate system for this specific area.



Everyone gets a map for the next exercise

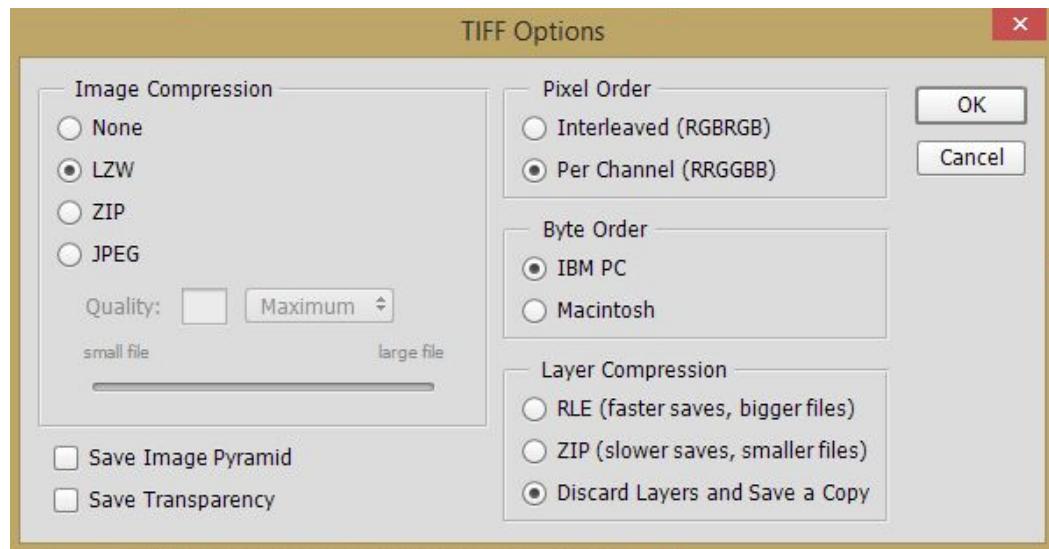
KEY MAP CITY OF ATLANTA										
1	2	3	4	5	6	7	8	9	10	
20	19	18	17	16	15	14	13	12	11	
21	22	23	24	25	26	27	28	29	30	
40	39	38	37	36	35	34	33	32	31	
41	42	43	44	45	46	47	48	49	50	

File Formats for Saving

GeoTIFF

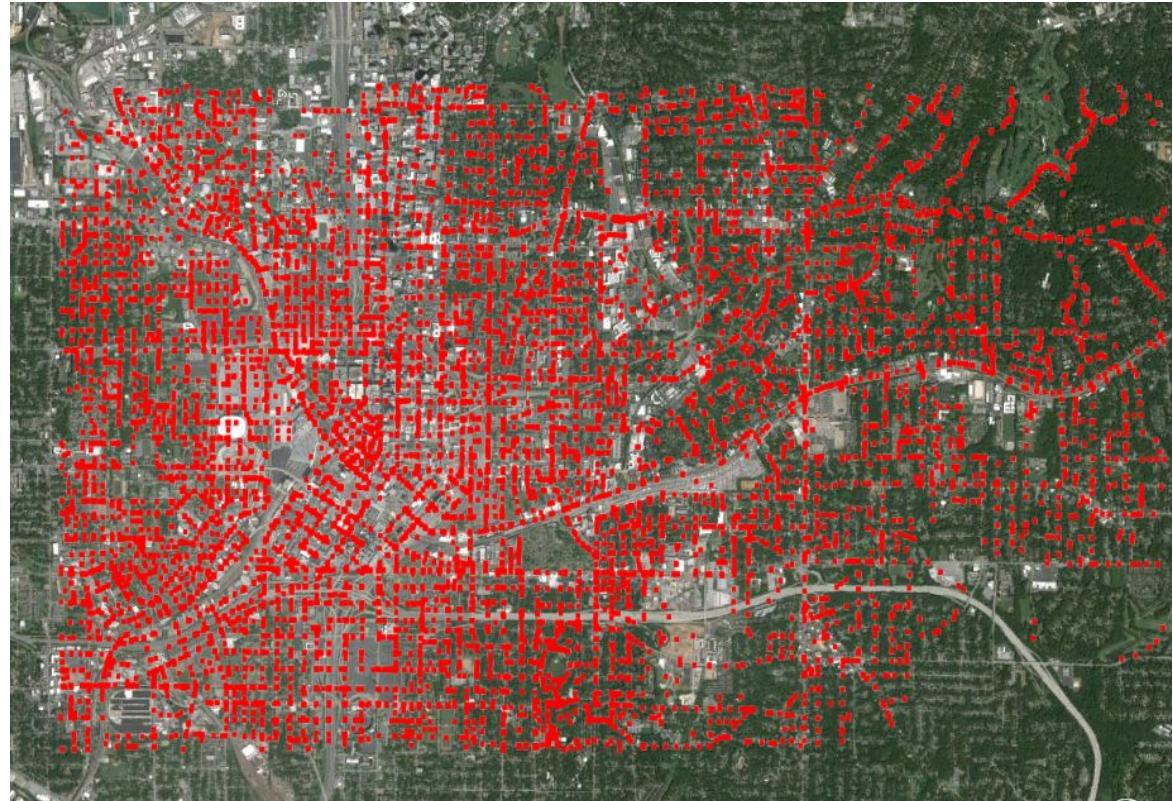
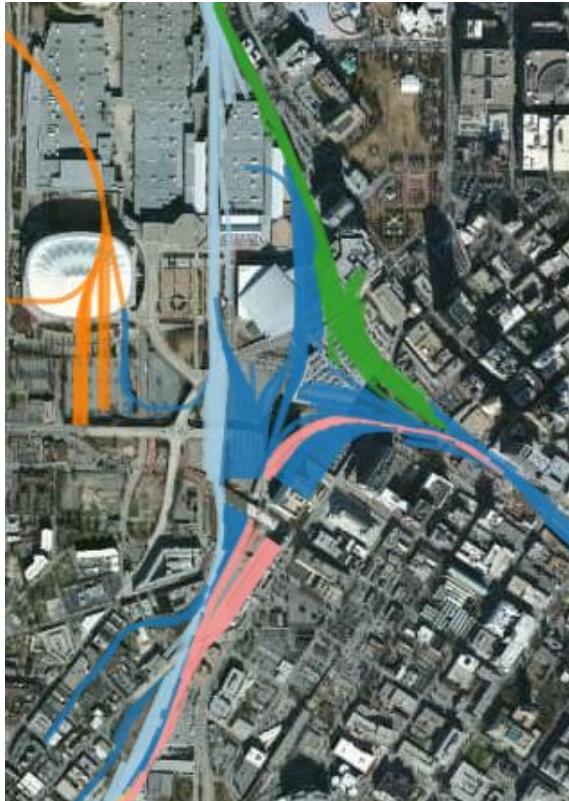
This is usually the best way to go.

- OK to use LZW compression
- Metadata/georeferencing information is all stored internally (i.e. you have just a single .tif file)
- 4 GB max file size



Extracting Features

Tracing features to create new data



Uploading your files to the web

GeoServer

There are 4 steps to add a map to GeoServer

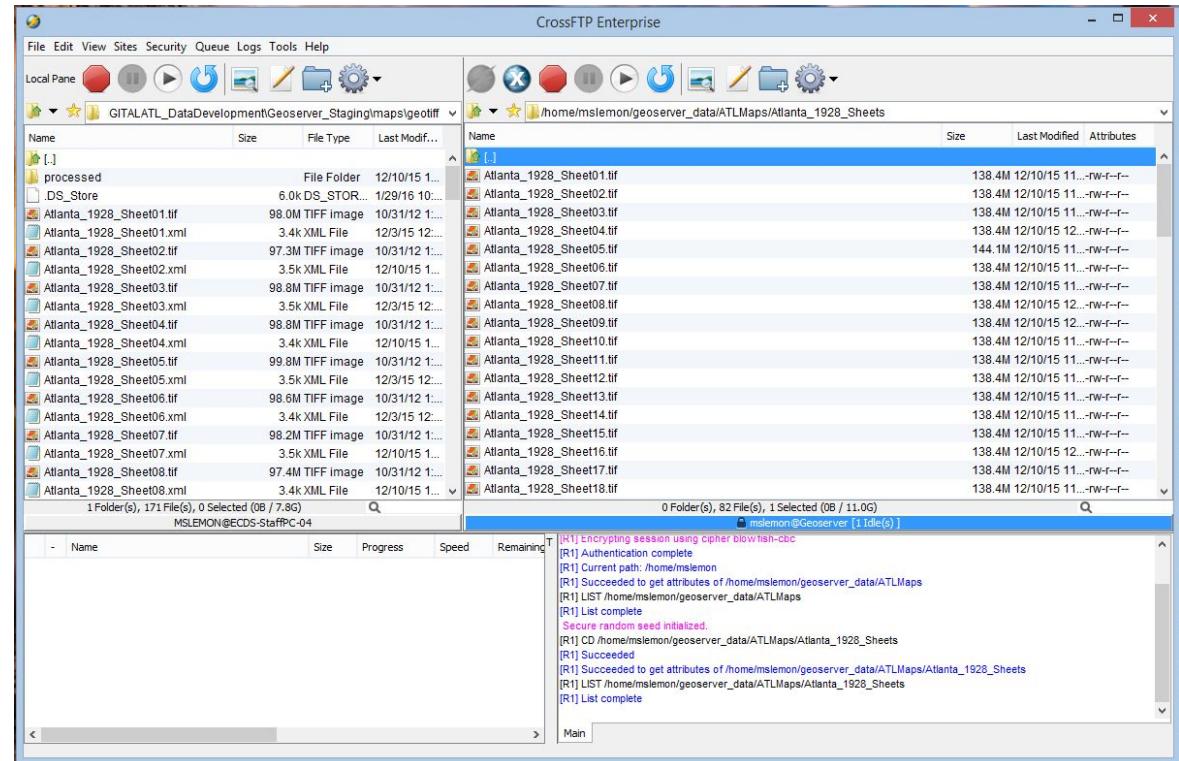
1. Upload the *GeoTiff* to the server
2. Create a *Workspace*
3. Create a *Store*
4. Publish the *Layer*
5. Update

Transfer your files to your server

Here is where you actually upload the files via SFTP/SSH to your server space. You will need:

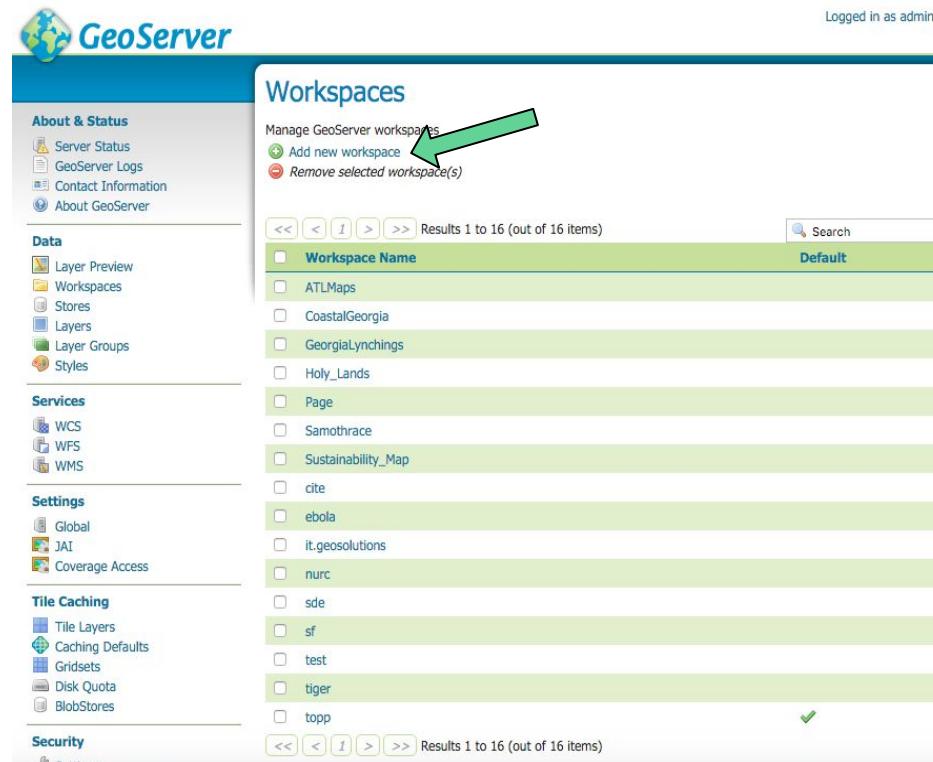
The server URL

A username and password



Create a workspace

- This creates a collection or group of layers.
- You could have just one workspace for all of your materials, or you can split them into sub-groups (workspaces) based on shared characteristics.
- The **Name** field cannot be duplicated within a workspace, which helps with managing unique IDs for layers.



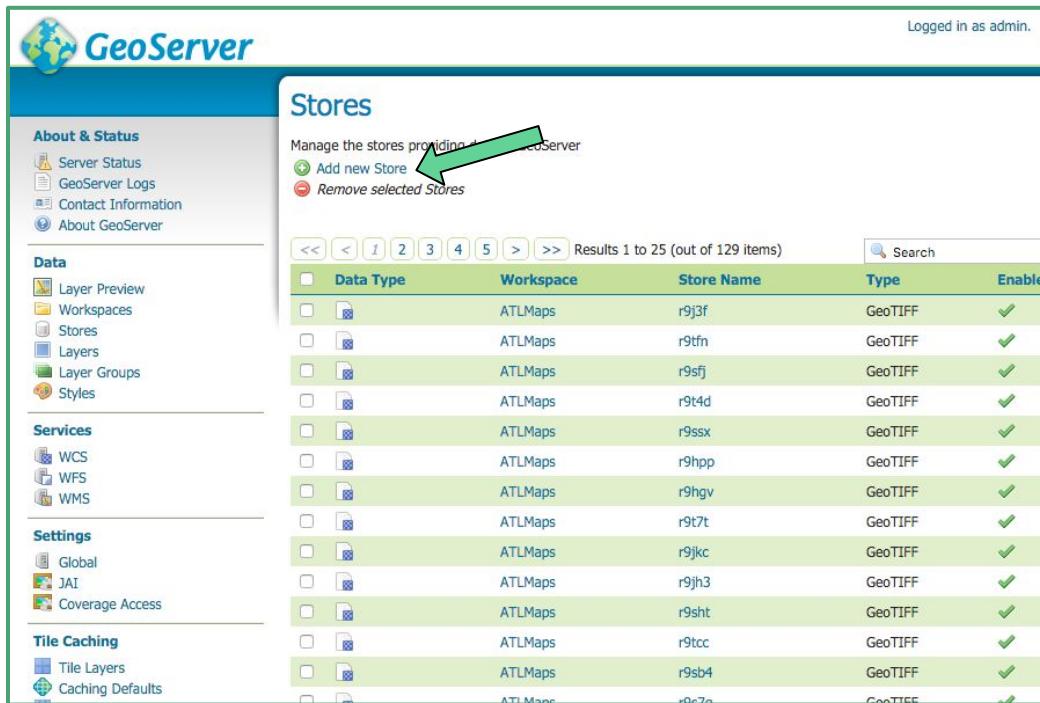
The screenshot shows the GeoServer administration interface. At the top right, it says "Logged in as admin". The main title is "Workspaces". Below it, there's a section for managing workspaces with two buttons: "Add new workspace" (highlighted by a green arrow) and "Remove selected workspace(s)".

The central part of the screen lists 16 workspaces, each with a checkbox and a name:

- ATLMaps
- CoastalGeorgia
- GeorgiaLynchings
- Holy_Lands
- Page
- Samothrace
- Sustainability_Map
- cite
- ebola
- it.geosolutions
- nurc
- sde
- sf
- test
- tiger
- topp

At the bottom, there are navigation buttons for "Results 1 to 16 (out of 16 items)".

Create a store



The screenshot shows the GeoServer interface with the 'Stores' page selected. The left sidebar contains links for About & Status, Data (Layer Preview, Workspaces, Stores, Layers, Layer Groups, Styles), Services (WCS, WFS, WMS), Settings (Global, JAI, Coverage Access), and Tile Caching (Tile Layers, Caching Defaults). The main content area has a title 'Stores' and a subtitle 'Manage the stores providing data to your GeoServer'. It includes buttons for 'Add new Store' (with a green arrow pointing to it) and 'Remove selected Stores'. Below is a table listing 25 stores (out of 129 items) with columns for Data Type (ATLMaps), Workspace, Store Name (e.g., r9j3f, r9tfn, r9sfj, r9t4d, r9ssx, r9hpp, r9hgv, r9t7t, r9jkc, r9jh3, r9sht, r9tcc, r9sb4, r9c2a), Type (GeoTIFF), and Enabled (checkmark). The table has a search bar at the top right.

Data Type	Workspace	Store Name	Type	Enabled
ATLMaps		r9j3f	GeoTIFF	✓
ATLMaps		r9tfn	GeoTIFF	✓
ATLMaps		r9sfj	GeoTIFF	✓
ATLMaps		r9t4d	GeoTIFF	✓
ATLMaps		r9ssx	GeoTIFF	✓
ATLMaps		r9hpp	GeoTIFF	✓
ATLMaps		r9hgv	GeoTIFF	✓
ATLMaps		r9t7t	GeoTIFF	✓
ATLMaps		r9jkc	GeoTIFF	✓
ATLMaps		r9jh3	GeoTIFF	✓
ATLMaps		r9sht	GeoTIFF	✓
ATLMaps		r9tcc	GeoTIFF	✓
ATLMaps		r9sb4	GeoTIFF	✓
ATLMaps		r9c2a	GeoTIFF	✓

A store is where you connect your raster or vector data. There are Geoserver extensions available for multiple data formats. We'll focus on GeoTIFF, under Raster Data Sources.

Create a store (continued)

GeoServer

New data source

Choose the type of data source you wish to configure

Vector Data Sources

- Directory of spatial files (shapefiles) - Takes a directory of shapefiles
- PostGIS - PostGIS Database
- PostGIS (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature in Shapefile - ESRI(tm) Shapefiles (*.shp)
- Web Complex Feature Server (NG) - Provides access to the Complex the ability to perform transactions on the server (when supported / allow)
- Web Feature Server (NG) - Provides access to the Features published on the server (when supported / allowed).

Raster Data Sources

- ArcGrid - Arc Grid Coverage Format
- GeoTIFF - Tagged Image File Format with Geographic information
- Gtopo30 - Gtopo30 Coverage Format
- ImageMosaic - Image mosaicking plugin
- ImagePyramid - Image pyramidal plugin
- JPK2 (Direct) - JP2K (Direct) Coverage Format
- WorldImage - A raster file accompanied by a spatial data file

Other Data Sources

- WMS - Cascades a remote Web Map Service

Settings

- Global
- JAI
- Coverage Access

Title Caching

- Tile Layers
- Caching Defaults
- Gridsets
- Disk Quota
- BlobStores



GeoServer

Add Raster Data

Description

GeoTIFF
Tagged Image File Format with Geo

Basic Store Info

Workspace *

ATLMaps

Data Source Name *

Atlanta_1928_Sheet45

Description

Atlanta_1928_Sheet45

Enabled

Connection Parameters

URL *

file:/data/example.extension

Save **Cancel**

Name	Last modified
Atlanta_1928_Sheet01.tif	Dec 10
Atlanta_1928_Sheet02.tif	Dec 10
Atlanta_1928_Sheet03.tif	Dec 10
Atlanta_1928_Sheet04.tif	Dec 10
Atlanta_1928_Sheet05.tif	Dec 10
Atlanta_1928_Sheet06.tif	Dec 10
Atlanta_1928_Sheet07.tif	Dec 10
Atlanta_1928_Sheet08.tif	Dec 10
Atlanta_1928_Sheet09.tif	Dec 10
Atlanta_1928_Sheet10.tif	Dec 10
Atlanta_1928_Sheet11.tif	Dec 10
Atlanta_1928_Sheet12.tif	Dec 10
Atlanta_1928_Sheet13.tif	Dec 10

Name your store, and then navigate to the file location on the server.

Publish the layer

This step turns the store into the final published layer.

There are many options on this page:

- Fill in the Name and Title (Name is the unique/non-duplicating field)
- Add a description under Abstract
- You may add a URL link to a metadata file
- Check the native and declared coordinate systems.
- Make sure the bounding boxes contain values.
- Save the layer.

New Layer

Add a new layer

On stores you can also create a new coverage view by merging different coverages as a multibands coverage. [Configure view ...](#)

Here is a list of resources contained in the store 'Example_Code4Lib'. Click on the layer you wish to configure

<< < [1](#) > >> Results 1 to 1 (out of 1 items) [Search](#)

Published	Layer name	Action
	Atlanta_1928_Sheet01	Publish

<< < [1](#) > >> Results 1 to 1 (out of 1 items)

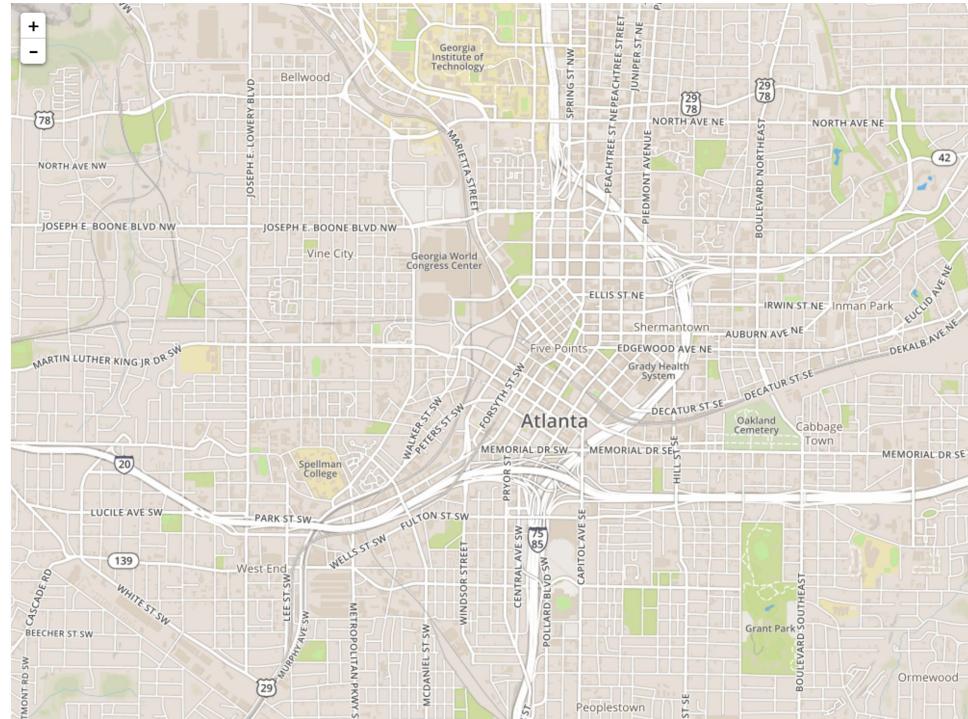


Tile the Map and Host it Yourself

GDAL comes with a *gdal2tiles.py* utilitly that can create a tiles of your georeferenced image. It can be tricky to get right.

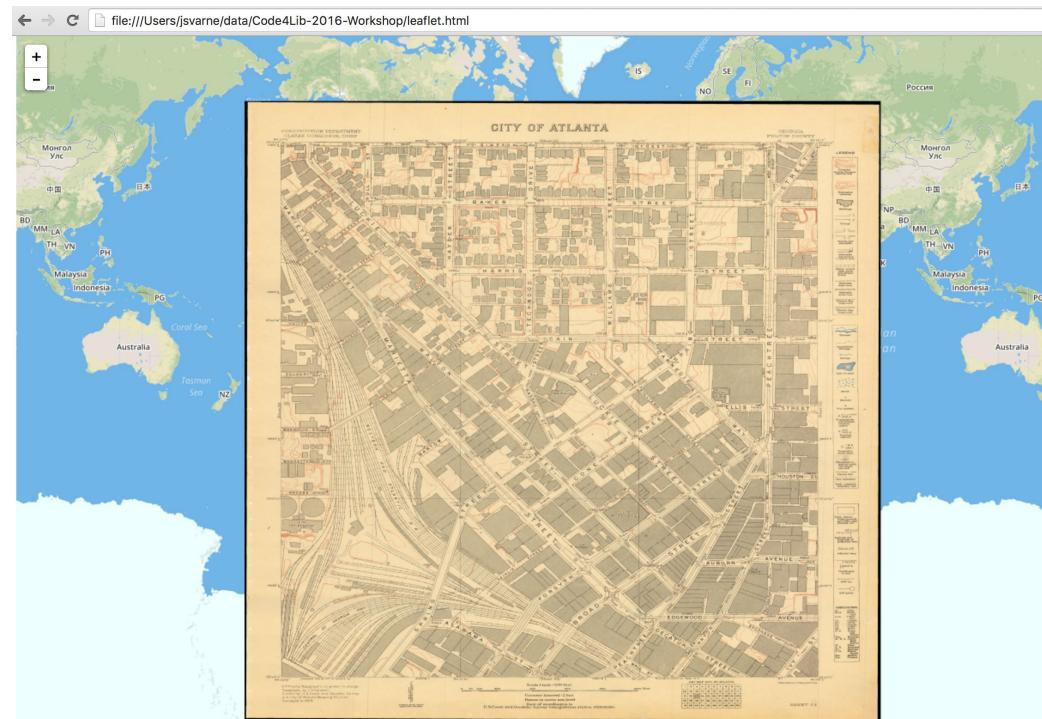
Tile the Map and Host it Yourself

I got a lot of this:



Tile the Map and Host it Yourself

A good bit of this:



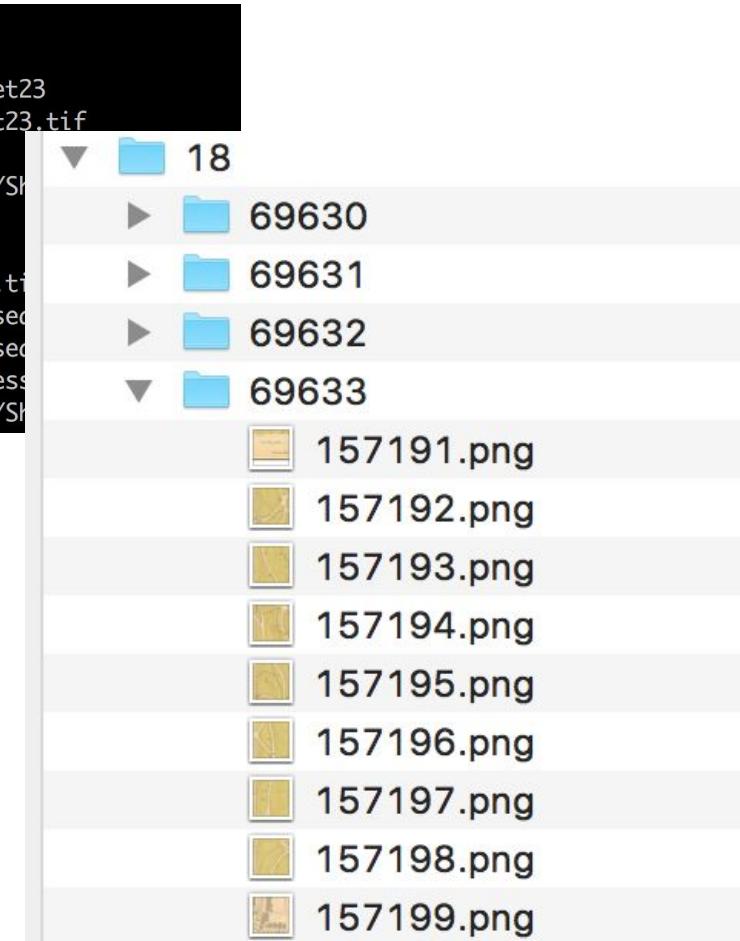
Tile the Map and Host it Yourself

After many attempts, I got it right:

```
gdal2tiles.py -p raster -z 0-7 -w all Sheet23.tif Sheet23  
gdal2tiles.py -d -p raster -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -p raster -z 0-20 -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -p raster -z 1-20 -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -p geodetic -z 1-20 -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -s EPSG:4326 -z 1-20 geotiffs/processed/Sheet23.tif  
gdal2tiles.py -s EPSG:4326 -z 1-20 atlanta_1928_sheet45.tif  
gdal2tiles.py -s EPSG:3857 -z 1-20 atlanta_1928_sheet45.tif
```

WML-JVARNER / jsvarne 23:11:08 [~]

```
$ cat ~/.bash_history | grep -i gdal2tiles  
gdal2tiles.py -p raster -z 0-7 -w all Sheet23.tif Sheet23  
gdal2tiles.py -d -p raster -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -p raster -w none geotiffs/processed/Sheet23.tif  
gdal2tiles.py --version  
gdal2tiles.py --version  
gdal2tiles.py -d -p raster geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -p raster -z 0-20 -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -p raster -z 1-20 -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -d -p geodetic -z 1-20 -k geotiffs/processed/Sheet23.tif  
gdal2tiles.py -s EPSG:4326 -z 1-20 geotiffs/processed/Sheet23.tif
```



Disk Usage

Tiles 187M

GeoTIFF 138M

Make the tiles web accessible

1. Where to store: File server, S3, Google Cloud Storage, etc.
2. Distribute CloudFront, CloudFlare, Web server* etc.

* Apache, NGINX, etc

- <http://disc.library.emory.edu/atlanta1928topo/tilesTopo/>

Displaying Maps on the Web

Options

JavaScript libraries/APIs to render maps online:

- Leaflet
- OpenLayers

Examples with Leaflet and OpenLayers are in the GitHub repo:

<https://github.com/emory-libraries-ecds/Code4Lib-2016-Workshop>

.....

- Google
- Others I haven't looked at

Why I Like Leaflet

- Lighter weight
- You can more easily access the various layers in the DOM
- Leaflet has plugins

Processing with GDAL

Special thanks to

Eric Willoughby, Georgia State University

Amanda Henley, University of North Carolina at Chapel Hill

Their help was invaluable in figuring out this next part...

Overview of the process

Change the projection to WGS 1984



Change the internal tiling of the GeoTIFF



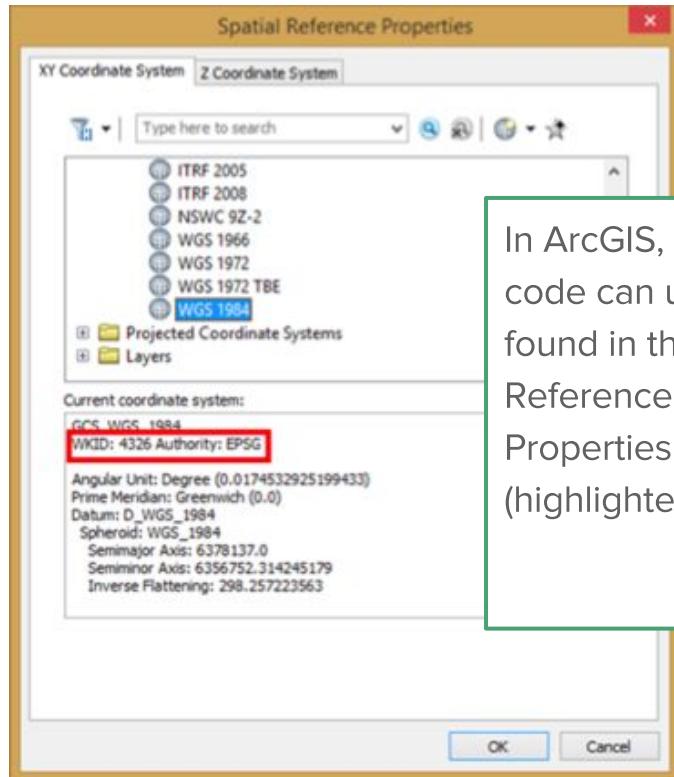
Add overviews

How to determine the EPSG code of your data

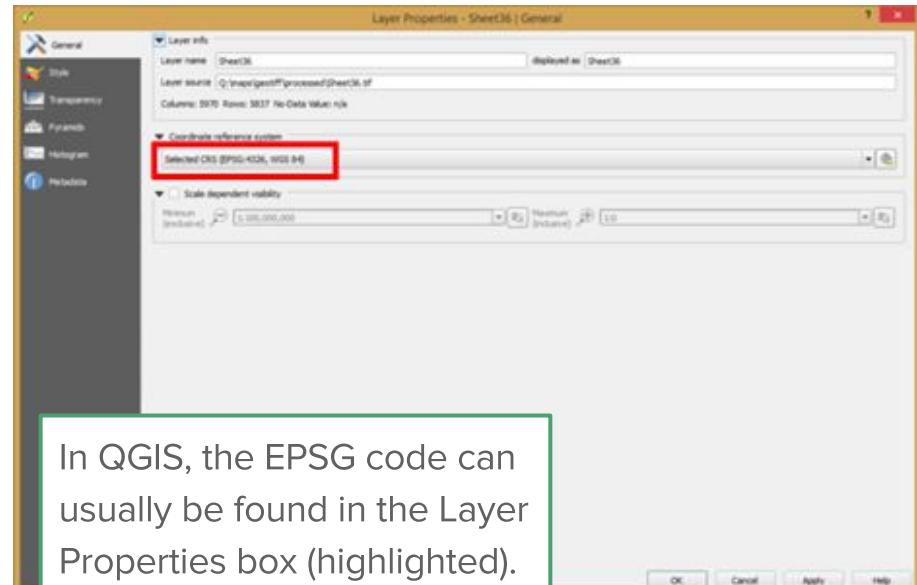
Use *gdalinfo* to verify the EPSG code of a geotiff.

```
WML-JVARNER.home / jsvarne 12:11:55 [ ~/Sandbox/geoserverUpload ]
$ gdalinfo sheet45-3857.tif
Driver: GTiff/GeoTIFF
Files: sheet45-3857.tif
Size is 5362, 6351
Coordinate System is:
PROJCS["WGS 84 / Pseudo-Mercator",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
            SPHEROID["WGS 84",6378137,298.25723563,
                AUTHORITY["EPSG","7030"]],
            AUTHORITY["EPSG","6326"]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433],
        AUTHORITY["EPSG","4326"]],
    PROJECTION["Mercator_1SP"],
    PARAMETER["central_meridian",0],
    PARAMETER["scale_factor",1],
    PARAMETER["false_easting",0],
    PARAMETER["false_northing",0],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    EXTENSION["PROJ4", "+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0
ds=null +wktext +no_defs"],
    AUTHORITY["EPSG","3857"]]
```

How to determine the EPSG code of your data



In ArcGIS, the EPSG code can usually be found in the Spatial Reference Properties box (highlighted).



In QGIS, the EPSG code can usually be found in the Layer Properties box (highlighted).

Step 1: Change the projection (gdalwarp)

In this step, you need to reproject your data from any other coordinate system into WGS 1984 (EPSG 4326).

Geoserver can reproject on-the-fly, but you lose some image quality and use server resources for no real reason.

Sample code:

```
E:\geotiffs\original>gdalwarp -s_srs EPSG:2240 -t_srs EPSG:4326 -r average Sheet  
45.tif Sheet45EPSG4326c.tif
```

Step 2: Change the internal tiling (gdal_translate)

GeoTIFF files are organized in 1 pixel strips by default. We want to retile the internal structure of the GeoTIFF to the dimensions of the tiles we need Geoserver to render.

Geoserver doesn't actually need to load the 1px strips, and can work a lot faster if the GeoTIFF is broken down into blocks ("tiles") that Geoserver can load faster. Geoserver can load only the parts of the image it's being asked for.

Sample code:

```
E:\geotiffs\reprojected>gdal_translate -co "TILED=YES" -co "BLOCKXSIZE=256" -co  
"BLOCKYSIZE=256" Sheet45EPSG4326.tif ../retiled/Sheet45EPSG4326b.tif
```

Step 3: Add overviews (gdaladdo)

Overviews allow only the needed resolution to be displayed at different zoom levels.

The GeoTIFF format is capable of storing overviews internally (rather than externally like are generated in ArcGIS).

Overviews are lost in reprojection, which is why this step needs to be last.

This step increases file size by about 25%.

Sample code:

```
E:\geotiffs\retiled>gdaladdo -r average Sheet45EPSG4326t.tif 2 4 8 16 32  
0...10...20...30...40...50...60...70...80...90...100 - done.
```

A Script to automate some of this

How it works

Using GeoServer's REST API, we can automate the process of making layers available from GeoServer

<http://docs.geoserver.org/stable/en/user/rest/api/>

Thank you!

We'll send an email with links to the slides and other sites of interest by the end of the week.

Feel free to email us if you have any questions:

megan.slemons@emory.edu

jay.varner@emory.edu
@jayvarner

Have a great week at Code4Lib!

