

Arielle Simmons, Planner/GIS Specialist

Bridging GIS Environments: using FWTools, OGR/GDAL, and Google Maps/Fusion Tables API.

NEARC



Northeast Arc Users Group

Questions about Open Source

- WHO
- WHY
- HOW

Questions about Open Source

- **WHO** ...uses it?
- **WHY** ...use it?
- **HOW** ...can I use it?

Questions about Open Source

- **WHO**uses it?
- **WHY** ...use it?
- **HOW** ...can I use it?

Pioneer Valley Planning Commission



43 Communities w/ Population range: 426 to 152,000

Pioneer Valley Planning Commission

PVPC : GIS/DATA CENTER

- ⌋ 2 Full Time GIS staff & only 2 ArcInfo Licenses
- ⌋ 1 Full Time Data Specialist
- ⌋ 1 Full Time Graphic Artist for desktop/web publishing
- ⌋ 2-3 part-time interns (1 GIS, 2 Data w/ some GIS knowledge)

Pioneer Valley Planning Commission

Pioneer Valley Planning Commission Staff

- | 10 Support Staff
- 9 Transportation Planners (+ 4 interns)
- 9 Land Use Planners (+ 2-3 interns)
- 8 Community Development Planners (+ 2-3 interns)
- | 5 Administrators....

Pioneer Valley Planning Commission

Pioneer Valley Planning Commission Staff

- | 10 Support Staff
- 9 Transportation Planners (+ 4 interns)
- 9 Land Use Planners (+ 2-3 interns)
- 8 Community Development Planners (+ 2-3 interns)
- | 5 Administrators....

All wanting maps yesterday! –J.Scace

Introducing QGIS....

...Attempts to Fill the Resource Void...

Ø QGIS – Open Source GIS

(January 24, 2013 : Formally Introduced

10 staff to QGIS by offering a 2-hr training session ...)



Ø New **Solutions** bring New **Problems** :

-increasing staff GIS access = immediate need for improved data management

- Also... *many* data files (approx. 20 years worth) need to be converted from proprietary formatting (i.e. File GDB) into open source QGIS-friendly formats (i.e. Personal GDB, .shp). Preferably clipped to the PVPC region and properly projected.

Questions about Open Source

- WHO ...uses it?
- **WHY** ...use it?
- HOW ...can I use it?

Grandpa's Rules...

Kenneth J Simmons



- Ø Math Teacher turned Electrical Engineer
- Ø Programmer/Problem Solver
- Ø Early Career: NASA (Hubble), Astrodata, Start Up's <?> , Disneyland (Pirates of the Caribbean)...
- Ø Late Career: Sr. System's Engineer at Lockheed-Martin -Skunk Works

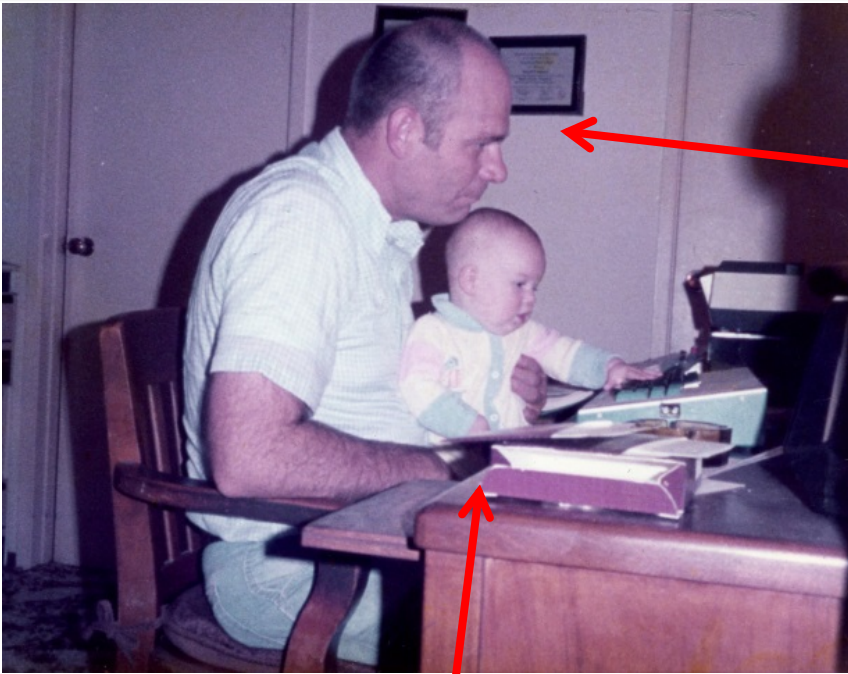
...NOT EVEN close to being his most challenging job. ...

Which he started in 1982, when he became....

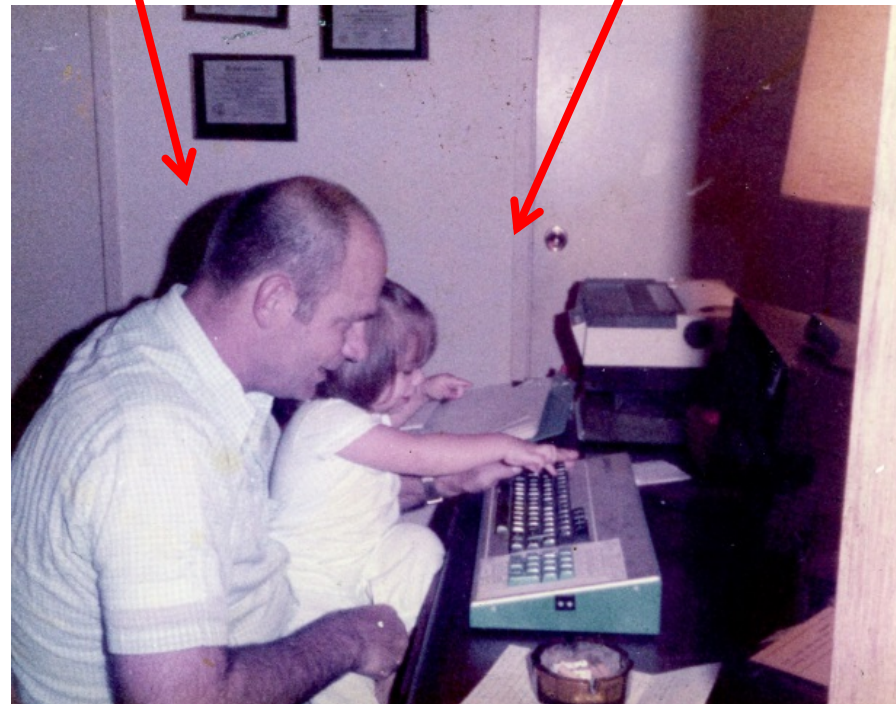
Grandpa's Rules...

The Babysitter!

Me (age 1.9)



Brother
(age <1)



Grandpa's Rules...

Rule #1: You can NEVER have too many tools....



Grandpa's Rules...

Rule #1: You can NEVER have too many tools....

Rule #2: When in DOUBT read the instructions....

Why use Open Source?

ü PROs

- Affordable
- Can create efficiency & speedier delivery (once you know the libraries/APIs)
- Fast Bug Fixes
- & you can NEVER have too many tools....

ü CONs

- Smaller User Community
- Sparse & more Technical documentation (..Technical, but not illegible!)
- Extra installation steps.... (i.e. setting up the "PATH" and "GDAL_DATA" variables)

...and if all else fails – just **READ the INSTRUCTIONS...** J

Questions about Open Source

- WHO ...uses it?
- WHY ...use it?
- **HOW** ...can I use it?

FWTools & OGR/GDAL

OVERVIEW:

FWTools – a packaged set of installs that INCLUDES the GDAL/OGR libraries (...granted a somewhat outdated version...)

GDAL stands for the [Geospatial Data Abstraction Library](#) .

- TWO different libraries combined into one!
 - 1) GDAL for manipulating geospatial **raster data**
 - 2) OGR for manipulating geospatial **vector data**

..many of the download and resource sites (including **FWTools**) sometimes truncate the name to just “GDAL” ...

FWTools & OGR/GDAL

- ONE drawback : installation process...however, FWTools is a bit easier and a bit more packaged than the OGR/GDAL download.
- ONE positive thing: Multi-supported languages...
 - Ø Run OGR/GDAL functions from the FWTools shell OR Windows Terminal (*modifying the 'Environmental Variables' is necessary)...
 - Ø Bindings are available for Python, Perl, VB6, R, Ruby, Java, C#/.Net....

WHERE TO DOWNLOAD:

If you are 'Just Starting Out'...

FWTools download website:
<http://fwtools.maptools.org/>

WARNING : This is *an old* installer. Python bindings DO NOT ALWAYS work.

OR

If you are ready to 'Get Serious'....

OGR/GDAL download website:
<http://www.gisinternals.com/sdk/>

WARNING : Make sure you download the right bindings for your version of Python. This is the best, most up-to-date package to install.. but not as simple as FWTools.

Using FWTools & OGR/GDAL

Problem : New QGIS user needs me to (quickly) re-project all 11 of their shapefiles to the NAD 1983 Massachusetts State Plane Coordinate System, Mainland Zone, 2001. Meters.

-A quick review of the data shows that all the files are in one folder/directory...

-AND ALL the projections are 'GCS_WGS_1984' – thus I am not required to use transformations...

Solution : Use the FWTools installation package and write a .bat file that loops through all the shapefiles and re-writes their projection....

...In order to get the most out of the FWTools install it is recommended that you set the Environmental Variables For both the PATH ...and the GDAL_DATA variables.

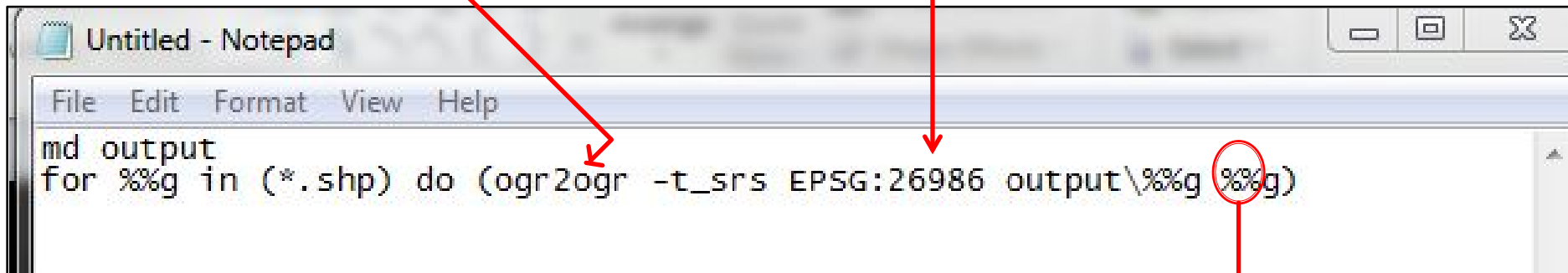
Using FWTools & OGR/GDAL

Solution : Using the FWTools installation package write a .bat file that loops through all the shapefiles and re-writes their projection....

Ø Step 1: Open a text editor -- Notepad will work --Enter the following:

'ogr2ogr' – command line tool that converts ogr data sources..

EPSG # for NAD 83 Mass State Plane



```
md output
for %%g in (*.shp) do (ogr2ogr -t_srs EPSG:26986 output\%%g %%g)
```

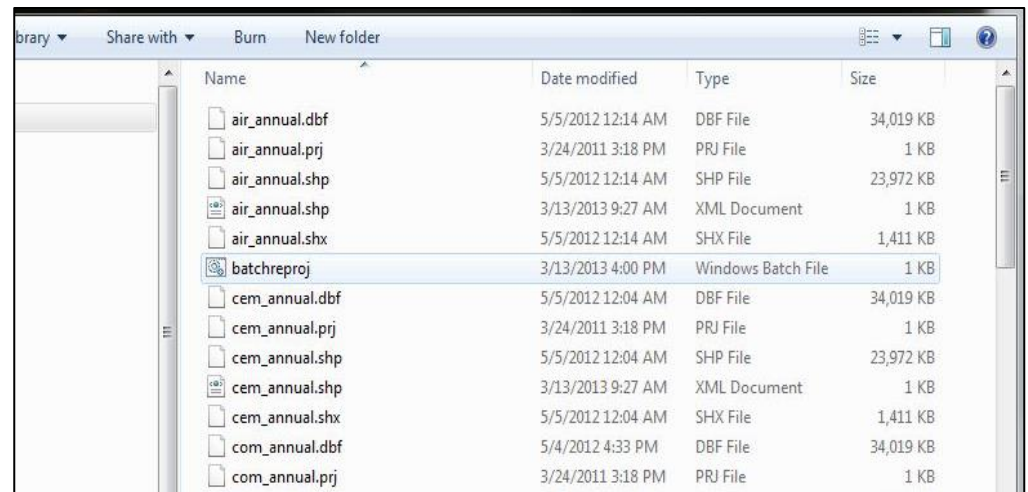
'%' = command line parameters
'%%' = in a .bat file are treated like a single '%' in the cmd

Using FWTools & OGR/GDAL

Solution : Using the FWTools installation package write a .bat file that loops through all the shapefiles and re-writes their projection....

Ø Step 2: Save as a .bat file...by going to File>Save As> AND entering 'batchreproj.bat' in the entry field.

Ø Step 3 : Place the batchreproj.bat file in the SAME folder as the data.



Using FWTools & OGR/GDAL

Solution : Using the FWTools installation package write a .bat file that loops through all the shapefiles and re-writes their projection....

Ø Step 4: Run the .bat file by:

1) Open MS DOS command line terminal (Start Menu>(type 'cmd' into the 'Search Programs and Folder' box>Select and enter).

```
C:\Users\asimmons\Desktop\test>cd C:\WorkOrder\WO_47_Uulcan_PUTA_Shapefile\UulcanSHP\shps390\
```

2) In the Terminal type cd and the <folder directory path> where the data & .bat file are

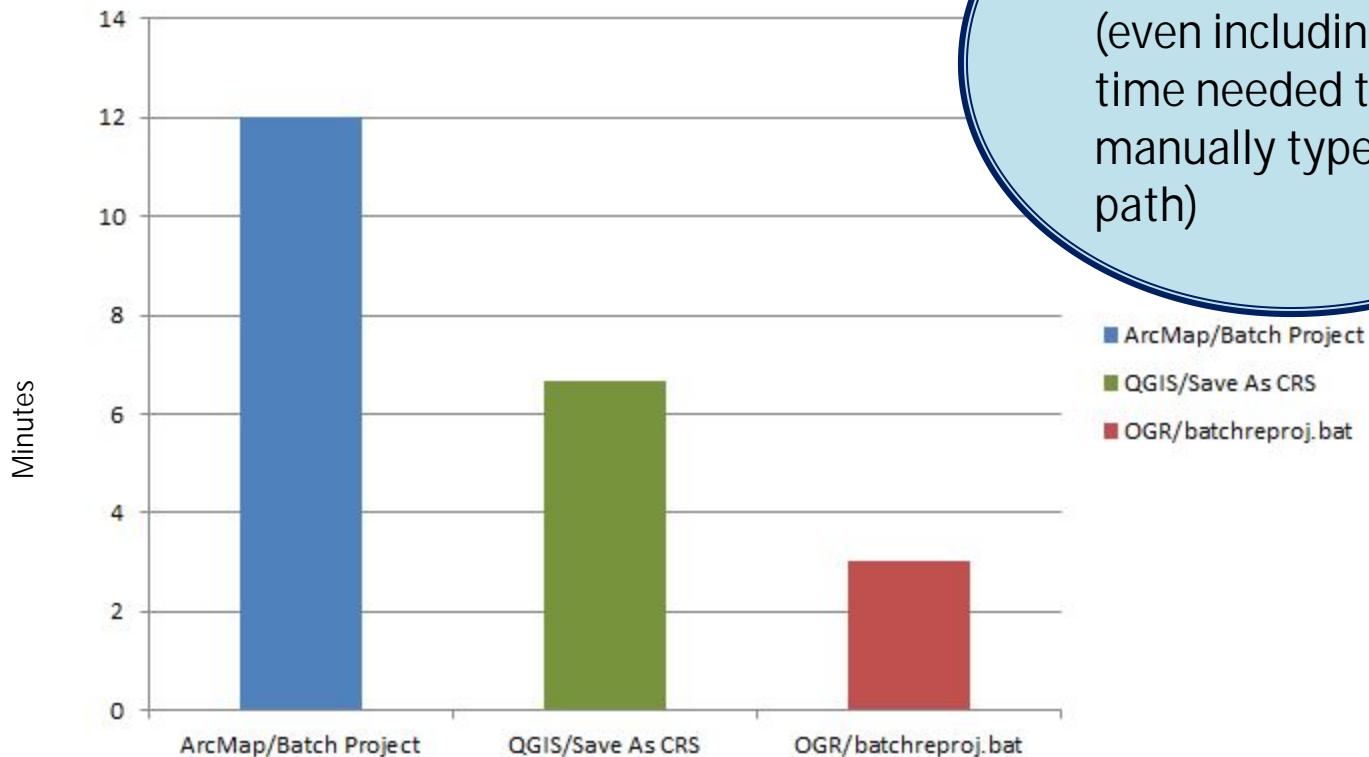
```
C:\WorkOrder\WO_47_Uulcan_PUTA_Shapefile\UulcanSHP\shps390>C:\WorkOrder\WO_47_Uulcan_PUTA_Shapefile\UulcanSHP\shps390\batchreproj.bat
```

3) Enter the .bat file from command line (it will be the full path and file name of the .bat file). After you do, hit enter and **RUN...**

Using FWTools & OGR/GDAL

Solution : Using the FWTools installation package write a .bat file that loops through all the shapefiles and re-writes their projection....

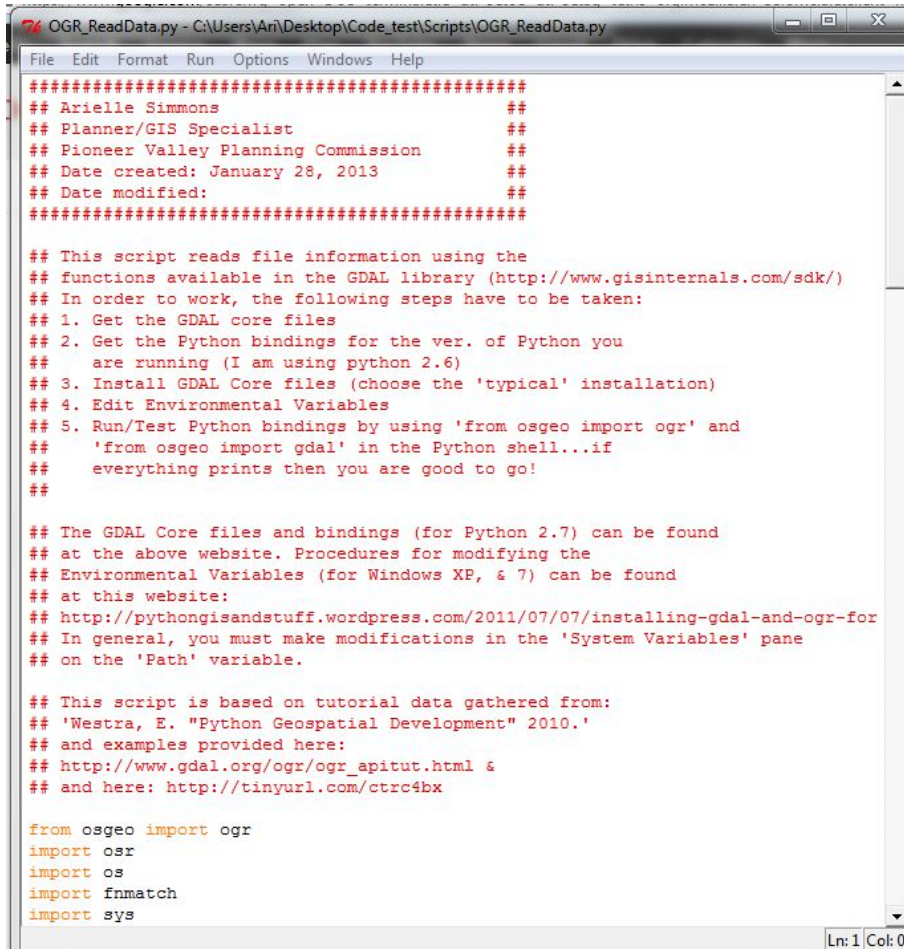
How long would it take to do this same task in....?



OGR/GDAL was **4x FASTER** then ArcMAP (even including the time needed to manually type the path)

Processor: Intel (QUAD) Core™ i7-3770 CPU @ 3.40GHz RAM: 16.0 GB

Testing some Python code with OGR/GDAL functions...



```
OGR_ReadData.py - C:\Users\Ari\Desktop\Code_test\Scripts\OGR_ReadData.py
File Edit Format Run Options Windows Help
#####
## Arielle Simmons ##
## Planner/GIS Specialist ##
## Pioneer Valley Planning Commission ##
## Date created: January 28, 2013 ##
## Date modified: ##
#####

## This script reads file information using the
## functions available in the GDAL library (http://www.gisinternals.com/sdk/)
## In order to work, the following steps have to be taken:
## 1. Get the GDAL core files
## 2. Get the Python bindings for the ver. of Python you
## are running (I am using python 2.6)
## 3. Install GDAL Core files (choose the 'typical' installation)
## 4. Edit Environmental Variables
## 5. Run/Test Python bindings by using 'from osgeo import ogr' and
## 'from osgeo import gdal' in the Python shell...if
## everything prints then you are good to go!
##

## The GDAL Core files and bindings (for Python 2.7) can be found
## at the above website. Procedures for modifying the
## Environmental Variables (for Windows XP, & 7) can be found
## at this website:
## http://pythongisandstuff.wordpress.com/2011/07/07/installing-gdal-and-ogr-for
## In general, you must make modifications in the 'System Variables' pane
## on the 'Path' variable.

## This script is based on tutorial data gathered from:
## 'Westra, E. "Python Geospatial Development" 2010.'
## and examples provided here:
## http://www.gdal.org/ogr/ogr_apitut.html &
## and here: http://tinyurl.com/ctrc4bx

from osgeo import ogr
import ogr
import os
import fnmatch
import sys
```

- **Problem**: I am in a hurry, ArcMap is already shut down... and a staff member asks me:

"...what can you tell me about these 11 shapefiles...?"

..what projection..

...what fields...

..how much data is there...

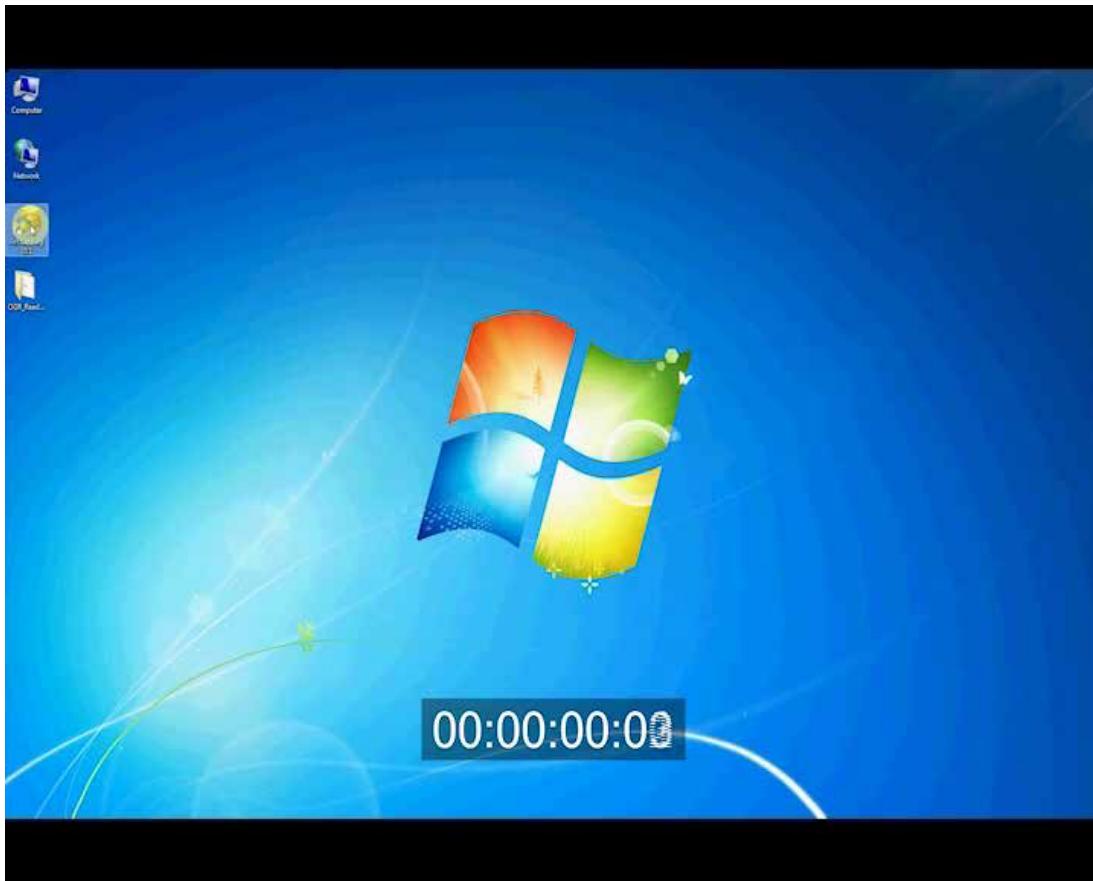
..NOW?!?"

- **Solution**: Open IDLE and run my ready-made Python script with OGR/GDAL functions!

>> *OGR_ReadData.py*

Testing some Python code with OGR/GDAL functions...

- How long would this take to do manually?



Computer Specs:

Processor: Intel® Core™
i7-3770 CPU @ 3.40GHz

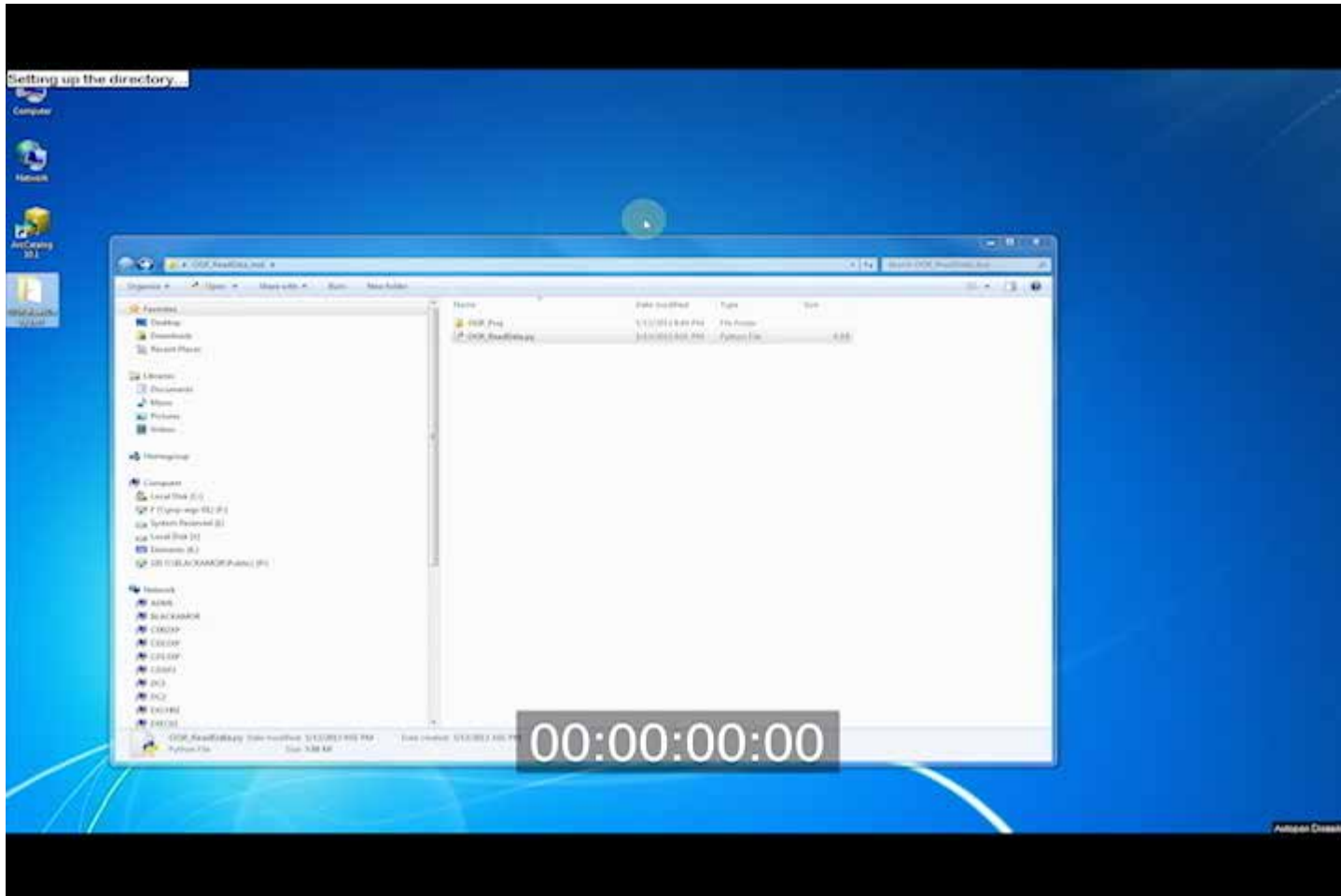
RAM: 16.0 GB

System: 64-bit

...at least a couple minutes per file....and there are 11 files...

Testing some Python code with OGR/GDAL functions...

>> *OGR_ReadData.py*



Testing some Python code with OGR/GDAL functions...

OGR data drivers...

(A driver is an object that knows how to interact with certain data types...like shapefiles...)

```
C:\Users\asinmons\Desktop\OGR_ReadData_test\OGR_Proj\tot_fall.shp
['ID', 'O_XMIN', 'O_XMAX', 'O_YMIN', 'O_YMAX', 'TOTCO2']
Number of Features: 180492
C:\Users\asinmons\Desktop\OGR_ReadData_test\OGR_Proj\tot_fall.shp has the WKT spatial reference: GE
OGCS["GCS_WGS_1984",DATUM["WGS_1984",SPHEROID["WGS_84",6378137,298.257223563]],PRIMEM["Greenwich",0],
UNIT["Degree",0.017453292519943295]]
EPSG:4326
```

STEP#1

Using os.walk & fnmatch functions, I construct a list of shapefile paths to pass

In my driver.Open() function

```
shpList = []
```

```
for path, dirs, files in os.walk(folder):
```

```
    for filename in fnmatch.filter(files, pattern):
```

```
        shpList.append(os.path.join(path, filename))
```

```
....
```

```
for shp in shpList:
```

```
    dataset = driver.Open(shp, 0)
```

Driver Open() method returns a DataSource object...

Open(<filename>, <update>)

0 = Read-only...

1 = Writeable..

Testing some Python code with OGR/GDAL functions...

OGR/GDAL supports:

...What is better than being 'fast'..... Supporting multiple data formats!!!.

- **Shapefiles, personal geodatabases, ArcSDE, FileGDB**
- **MapInfo, GRASS, Microstation**
- **Military Elevation Data (.dt0, .dt1, .dt2), ENVI (.hdr), Erdas Imagine (.img), Irdisi, USGS ASCII DEM, GDAL Virtual (.vrt)**
- **ASCII Gridded XYZ, Arc/Info ASCII Grid**
- **TIGER/Line, SDTS, GML, KML**
- **MySQL, PostgreSQL, Oracle Spatial, Informix, ODBC**
- **...and many many more!**

Format Name	Creation	Georeferencing
Arc/Info Binary Coverage	No	Yes
Atlas BNA	Yes	No
Comma Separated Value (.csv)	Yes	No
DODS/OPeNDAP	No	Yes
ESRI Personal GeoDatabase	No	Yes
ESRI ArcSDE	No	Yes
ESRI Shapefile	Yes	Yes
FMEObjects Gateway	No	Yes
GeoJSON	No	Yes
Geoconcept Export	Yes	Yes
GML	Yes	Yes
GMT	Yes	Yes
GPX	Yes	Yes
GRASS	No	Yes
INTERLIS	Yes	Yes
KML	Yes	No
Mapinfo File	Yes	Yes
Microstation DGN	Yes	No
Memory	Yes	Yes
MySQL	No	No
OGDI Vectors	No	Yes
ODBC	No	Yes
Oracle Spatial	Yes	Yes
PostgreSQL	Yes	Yes
S-57 (ENC)	No	Yes
SDTS	No	Yes
SQLite	Yes	No
UK NTF	No	Yes
U.S. Census TIGER/Line	No	Yes
VRT - Virtual Datasource	No	Yes
Informix DataBlade	Yes	Yes

Testing some Python code with OGR/GDAL functions...

...Because OGR/GDAL is so flexible , my scripting capabilities have expanded...

>>downloadData.py

A series of python scripts that I have been working on that automates the downloading, unzipping, organizing, transferring, and exporting of different MassGIS data types (Note: I can run this using Windows Task Scheduler).

Uses OGR/GDAL functions as well as functions from the Shapely library (Shapely = library capable of performing complex geometric calculations -- like buffer – on geo-spatial data)...

ALSO *because OGR/GDAL is so flexible , I do not have any license limitations on what machine I can work on and where I can work ...*

Fusion Tables/Google Maps...

- **Problem** : I have a WebGIS product to deliver in a week or less time frame (<40 hours).
 - The request only requires ONE data layer (2000+ Building polygons)
 - Has to display at least 8 fields of information
 - Minimal or no money for long-term data storage or server admin work...
- **Solution** : Google Maps/Fusion Tables API (w/ Google visualizations)

Hatfield Solar Map

Address:

Select a building outline or search for an address to identify buildings that you'd like to select.

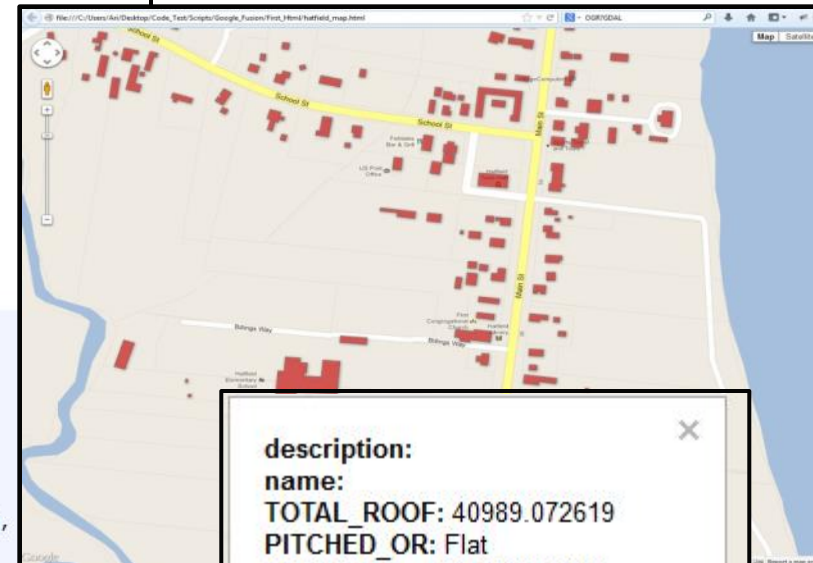
Buildings are symbolized according to roof size, and the possibility of increased panel placement.



A WebGIS in an hour (or less)...

37 lines of Javascript + HTML = 1 WebGIS

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
5   <style type="text/css">
6     html { height: 100% }
7     body { height: 100%; margin: 0; padding: 0 }
8     #hatfield_map { height: 100% }
9   </style>
10  <script type="text/javascript"
11    src="https://maps.googleapis.com/maps/api/js?sensor=false">
12  </script>
13  <script type="text/javascript">
14    function initialize() {
15      var mapOptions = {
16        center: new google.maps.LatLng(42.37098, -72.59818),
17        zoom: 17,
18        mapTypeId: google.maps.MapTypeId.ROADMAP
19      };
20
21      var map = new google.maps.Map(document.getElementById("hatfield_map"),
22        mapOptions);
23
24      var layer = new google.maps.FusionTablesLayer({
25        query: {
26          select: 'geometry',
27          from: '167ErkCtcUvzDW11u30o9ybPuhd5omBxmJkQBxL4'
28        },
29      });
30      layer.setMap(map);
31    }
32  </script>
33 </head>
34 <body onload="initialize()"
35   <div id="hatfield_map" style="width:100%; height:100%"></div>
36 </body>
37 </html>
```



description:
name:
TOTAL_ROOF: 40989.072619
PITCHED_OR: Flat
POTENTIAL_: 28692.350833
NUMBER_OF_: 1619.000000
WATTS: 404750.000000
KWH_PER_YR: 478110.937500
ANNUAL_DOL: 56417.090625
MONTHLY_DO: 4701.424219

A WebGIS in an hour (or less)...

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
5   <style type="text/css">
6     html { height: 100% }
7     body { height: 100%; margin: 0; padding: 0 }
8     #hatfield_map { height: 100% }
9   </style>
10  <script type="text/javascript"
11    src="https://maps.googleapis.com/maps/api/js?sensor=false">
12  </script>
13  <script type="text/javascript">
14    function initialize() {
15      var mapOptions = {
16        center: new google.maps.LatLng(42.37098, -72.59818),
17        zoom: 17,
18        mapTypeId: google.maps.MapTypeId.ROADMAP
19      };
20      var map = new google.maps.Map(document.getElementById("hatfield_map"),
21        mapOptions);
22
23      var layer = new google.maps.FusionTablesLayer({
24        query: {
25          select: 'geometry',
26          from: '167ErkCtcUvzDW11u30o9ybPuhd5omBxmJkQBxL4'
27        },
28      });
29      layer.setMap(map);
30    }
31  </script>
32 </head>
33 <body onload="initialize()"
34   <div id="hatfield_map" style="width:100%; height:100%"></div>
35 </body>
36 </html>
```

HTML : Declare as HTML5 and create a div element to hold the map...

Javascript: Create a Map options object to contain map initialization variables & create a 'map' object.

Javascript: Using the Google Maps Javascript API (v3) I am adding the data from my fusion table (which was imported into my table as a KML file) as a Layers object.

A WebGIS in an hour (or less)...

What the table looks like in Fusion....

Hatfield_Solar_Final
Imported at Tue Feb 05 13:37:18 PST 2013 from WGS84_HatfieldSolar_NO_X_Y.kml.
David Elvin (DElvin@PVPC.org) - Edited on February 11, 2013

File Edit Tools Help Rows 1 Cards 1 Map of geometry

Filter No filters applied. Sorted by TYPE_BLDG_

1-100 of 2,586

de...	name	STRUCT_ID	BLDG_ID	TOTAL_ROOF	PITCHED_OR	POTENTIAL_	NUMBER_OF_	WATTS	KWH_...	ANNUAL_DOL	MONTHLY_DO	TYPE_BLDG_	ELEC_PERMI	TYPE	geometry
		105038_903854	105038_903854	2,566.109	Pitched	1,026.444	55	13,750	16242.18	\$1,916.58	\$159.71	Solar Building Permit for 300kW Systems or less	Yes	EXCELLENT	KML...
		105329_903873	105329_903873	293.115	Pitched	117.246						Solar Building Permit for			
		105503_903994	105503_903994	1,156.104	Pitched	462.441									
		105682_905021	105682_905021	357.038	Pitched	142.815									
		106674_905166	106674_905166	8,168.109	Pitched	3,267.244									
		103602_903814	103602_903814	1,230.369	Pitched	492.148									
		103565_904225	103565_904225	1,368.371	Pitched	547.348									
		108632_907360	108632_907360	1,088.503	Pitched	435.401									

About this table

Name Hatfield_Solar_Final

Attribution David Elvin (DElvin@PVPC.org)

Visibility and reuse Public
Anyone on the internet can find and access. No sign-in required
Download allowed

Reuse license Unknown

Description Imported at Tue Feb 05 13:37:18 PST 2013 from WGS84_HatfieldSolar_NO_X_Y.kml.

Id 1DGswslbC5ijqWHPJvOH1NH7vtkZIPURJun_L5I

Done

A WebGIS in 20-30 hours....

- Customize map symbolization using Fusion Tables function calls...
- Add an address locator (Javascript)
- Add some JQUERY UI...and a sidebar...and a title...and a legend...

```
// Set up a styles array to fill in color for the different polygon types based on the query
// of the value 'TYPE' in the FusionTable...
    var layer = new google.maps.FusionTablesLayer({
        query:{
            select: 'geometry',
            from: '1DGswslbC5ijqWHPJvOH1NH7vltkZIPURJun_L5I'
        },
// Set up the stroke color and opacity of all polygons...
        styles: [{
            polygonOptions: {
                strokeColor: '#00ffff',
                strokeOpacity: .4,
                strokeWeight: .5
            }
        }, {
// Set TYPE 'EXCELLENT' to orange-red R-G-B, 255-69-0
            where: "'TYPE' MATCHES 'EXCELLENT'",
            polygonOptions: {
                fillColor: '#ff4500',
                fillOpacity: 10
            }
        }, {
```

A WebGIS in 20-30 hours....

- Add some JQUERY UI to your HTML file....

The screenshot shows the jQuery website with a jQuery UI tabs widget overlay. The widget has two tabs: "ROOF/PANEL AREA" and "ENERGY TOTALS". The "ROOF/PANEL AREA" tab is active and displays the following information:

- Total Roof Area (sqft) : 40989
- Potential Roof Area (sqft) : 28692
- Pitched or Flat Roof (?) : Flat
- Number of Panels for available Roof Area : 1619

Below the widget, there is a search bar and a list of links. A dashed box highlights a link to "jQuery UI 1.9 API Documentation" with the text "Manually overriding CSS ...my advice....avoid doing this!!!!" pointing to the CSS code on the right.

```
.tabs {  
    padding: 0px !important;  
}  
  
.tabs .ui-tabs-nav {  
    position: relative !important;  
    background: #f5f3e5 !important;  
    scrollbar: false !important;  
    cursor: pointer !important;  
    font-weight: normal !important;  
    border-width: 0px 0px 1px 0px !important;  
    -moz-border-radius: 0px !important;  
    -webkit-border-radius: 0px !important;  
    border-radius: 0px !important;  
    padding-left: 0px !important;  
    width: relative !important;  
    height: 380px;  
}  
  
.tabs .ui-tabs-panel {  
    background: #f5f3e5 !important;  
    margin: 0em 0.2em 0.2em 0.2em !important;  
}  
  
.tabs li {  
    background: lightgrey !important;  
    border-width: 0px !important;  
    padding: 0px;  
}  
  
.tabs li.ui-state-active {  
    background: lightblue !important;  
    color: black !important;  
}  
  
.tabs li a {  
    font-weight: bold !important;  
}
```

A WebGIS in 100-140 hours....

<https://googledrive.com/host/0BxTGtYdXPZ8QNE5tUWxZOWdzU2M>

Map Legend

Buildings are symbolized according to roof size, and the possibility of increased panel placement.

Clicking on the buildings will show you how much roof space (exactly) is available for panel placement.

Low	Good	Very Good	Excellent
Dark Blue	Light Blue	Yellow	Red

Information

The numbers provided are based off of the 'best available' data.

Watts generated = # of Panels * 250kw

Annual KWH = Watts generated / 1.18125
(*includes est. daylight/cloud conditions)

Value of Electricity PER Year = Electricity PER Year / 12

Value of Electricity PER Month = Electricity PER Year / 12

Hatfield Solar Map

Address:

[Click to Find Address](#)

ENTER YOUR ADDRESS TO FIND THE SOLAR POTENTIAL OF NEARBY BUILDINGS.

It is recommended to use this map without minimizing your browser window. Please make sure that Javascript is enabled!

ROOF/PANEL TOTALS:

Total Roof Area (sqft): 40,989
Potential Roof Area (sqft): 28,692
Pitched or Flat Roof (?): Flat
of Panels per Roof Area : 1,619

ENERGY TOTALS:

Watts generated: 404,750
Annual KWH : 478,111
Cost of Elec. PER Year : \$ 56,417.09
Cost of Elec. PER Month : \$ 4,701.42

Available Roof Area for Solar Panels (sq ft)

Category	Value (sq ft)
Building	~28,700
Bldg. Solar Area	~28,692
Bldg. Roof Area	40,989

LINKS & MATERIALS PROVIDED BY: **IDDER** Massachusetts Department of Energy Resources, **PVPC** Pioneer Valley Planning Commission

Google Visualizations...

Javascript

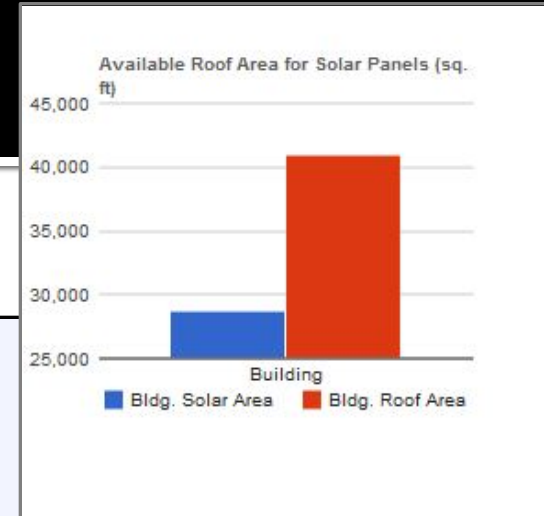
```
// global function to create column chart from fusion table data

function drawVisualization(e) {

    var data = google.visualization.arrayToDataTable([
        ['Area', 'Bldg. Solar Area', 'Bldg. Roof Area'],
        ['Building', Math.round(e.row['POTENTIAL_'].value), Math.round(e.row['TOTAL_ROOF'].value)],
    ]);

    var options = {
        title: 'Available Roof Area for Solar Panels (sq. ft)',
        titleTextStyle: {color: '#5c5c5c'},
        chartArea: {width:"75%", height:"70%"},
        legend: 'bottom',
    };

    var chart = new google.visualization.ColumnChart(document.getElementById('visualization_div'));
    chart.draw(data, options);
};
```



Google Visualizations...

Javascript

```
    },  
    // Function call: Click Listener on layer using jquery  
    google.maps.event.addListener(layer, 'click', function(e) {  
        $("#roof-panel-area").html(  
            '<p><strong>Total Roof Area (sqft)</strong>: ' + '&nbsp;&nbsp;&nbsp;' + Math.round(e.row['TOTAL_ROOF'].value) +  
            '<br><strong> Potential Roof Area (sqft)</strong>:' + '&nbsp;&nbsp;&nbsp;' + Math.round(e.row['POTENTIAL_'].value) +  
            '<br><strong> Pitched or Flat Roof (?)</strong>:' + '&nbsp;&nbsp;&nbsp;' + e.row['PITCHED_OR'].value +  
            '<br><strong> # of Panels per Roof Area :</strong>' + '&nbsp;&nbsp;&nbsp;' + Math.round(e.row['NUMBER_OF_'].value) +  
            '</p>');  
        drawVisualization(e);  
    });  
    layer.setMap(map);  
}
```

HTML

```
<div id = "visualization_div" style="width: 100%; height: 266%; align=center;"></div>
```

“Reading the Instructions” GOOGLE vs. OGR/GDAL....

- **Problem** : It takes time to learn a new API/Libraries – and the legibility of the instructions vary....



VS



- ü Provides clear & frequent code samples...
- ü Documentation is clean, direct, and not techno-centric
- ü Easy Menu to Move through

As a python developer...before I went here I usually tried to explore the documentation using Python's built-in `help()`, or `pydoc -g osgeo` (for command line).

- q Very few examples... and not all methods are documented so you might have to refer to the C++ docs on the GDAL website...
- q Documentation is... technically precise, but does require a higher degree of familiarity with code than the Google API.
- q Not an Easy Menu to Move through

Resources to seek out ...

More information on how to modify the System

Variables and Path variables (for GDAL): <http://tinyurl.com/bmIncmg>

GDAL library (includes links to wiki, downloads, API
reference documentation, and the Python
Library) :

<http://www.gdal.org>

Information about the Python package and
interfacing it with the GDAL library :

<https://pypi.python.org/pypi/GDAL/>

FWTools (Open Source GIS Binary Kit for Windows
AND Linux) :

<http://fwtools.maptools.org/>

Resources to seek out....

...the *very CRAZY* intimidating webpage which has the most recent build versions of GDAL : <http://www.gisinternals.com/sdk/>

This page contains the links to the most recent versions of the build SDKs and the related binary packages

This build environment compiles GDAL and MapServer from the subversion and provide cutting edge binary packages containing the latest fixes until the time when build has started (refer to build log for the details). Only the binaries of the latest successful builds are available to download. The possible breaking changes in the subversion causes the compilation to terminate immediately before the packaging is started.

Note: The different compilers or platforms involve different CRT dependencies, therefore the binaries of the different packages are not interchangeable.

The contents of the packages are provided under the terms of [this license](#). It is intended to give you permission to do whatever you want with the files: download, modify, redistribute as you please, including building proprietary commercial software, no permission from Tamas Szekeres is required. Some external libraries which can be optionally used by GDAL and MapServer (provided as plugins) are under radically different licenses, you **MUST obtain valid licenses** for each of these dependent libraries.

Reporting issues or enhancement requests related to the build system can be posted to the [github issue tracker](#).

The following binary only packages are compiled daily based on the MapServer and GDAL SVN (development and stable branches):

Compiler (Platform)	Downloads	Package Info	Build log	Build status	Latest revision
MSVC2003 (Win32) -development	release-1310-gdal-mapserver	information	buildlog msautotest	Packaging sources	772dcf2d65, r25991
MSVC2005 (Win32) -development	release-1400-gdal-mapserver	information	buildlog msautotest	Packaging headers and libs	772dcf2d65, r25991
MSVC2005 (Win64) -development	release-1400-x64-gdal-mapserver	information	buildlog msautotest	MapServer python build	32c839b9b9, r25989
MSVC2008 (Win32) -development	release-1500-gdal-mapserver	information	buildlog msautotest	GDAL autotest	32c839b9b9, r25989
MSVC2008 (Win64) -development	release-1500-x64-gdal-mapserver	information	buildlog msautotest	GDAL autotest	32c839b9b9, r25989
MSVC2010 (Win32) -development	release-1600-gdal-mapserver	information	buildlog msautotest	GDAL autotest	32c839b9b9, r25989
MSVC2010 (Win64) -development	release-1600-x64-gdal-mapserver	information	buildlog msautotest	GDAL autotest	32c839b9b9, r25989
MSVC2003 (Win32) -stable	release-1310-gdal-1-10-mapserver-6-2	information	buildlog msautotest	finished at 2013.05.06. 7:05:12,06	fe843bf95c, r25989
MSVC2005 (Win32) -stable	release-1400-gdal-1-10-mapserver-6-2	information	buildlog msautotest	finished at 2013.05.06. 7:19:26,05	fe843bf95c, r25989
MSVC2005 (Win64) -stable	release-1400-x64-gdal-1-10-mapserver-6-2	information	buildlog msautotest	finished at 2013.05.06. 7:39:40,42	fe843bf95c, r25989
MSVC2008 (Win32) -stable	release-1500-gdal-1-10-mapserver-6-2	information	buildlog msautotest	finished at 2013.05.06. 8:01:12,30	fe843bf95c, r25989
MSVC2008 (Win64) -stable	release-1500-x64-gdal-1-10-mapserver-6-2	information	buildlog msautotest	finished at 2013.05.06. 8:17:37,21	fe843bf95c, r25989
MSVC2010 (Win32) -stable	release-1600-gdal-1-10-mapserver-6-2	information	buildlog msautotest	finished at 2013.05.06. 8:17:17,05	fe843bf95c, r25989
MSVC2010 (Win64) -stable	release-1600-x64-gdal-1-10-mapserver-6-2	information	buildlog msautotest	finished at 2013.05.06. 8:20:08,14	fe843bf95c, r25989

Contact Information

- Ari Simmons
 - 413-781-6045 (w)
 - 508-723-2496 (c)
 - Asimmons@PVPC.org or ari.ucb.fire@gmail.com

Feel free to ask me questions!