Batch Geoprocessing using Python to Manage Data Efficiently

Xingwen Chen

ABCD GIS Mapping

4/21/2016

Introduction

- Oil/gas mineral ownership/lease mapping
- Data from different sources, formats, datum, and projections
- Data may not have projection information
- Data usually come in by counties such as Surface Well Locations & Bottom Holes, and some are state level data, such as Texas Original Land Survey
- Process these data sets and save in a geodatabase, re-define projections, project all data in the same DATUM & projection, query for subset of data, calculate fields, and convert all data layers to sequential list of points with latitude/longitude that are to be used in customized applications.
- Python and PyScripter are used and multiple counties of data may be process at once.

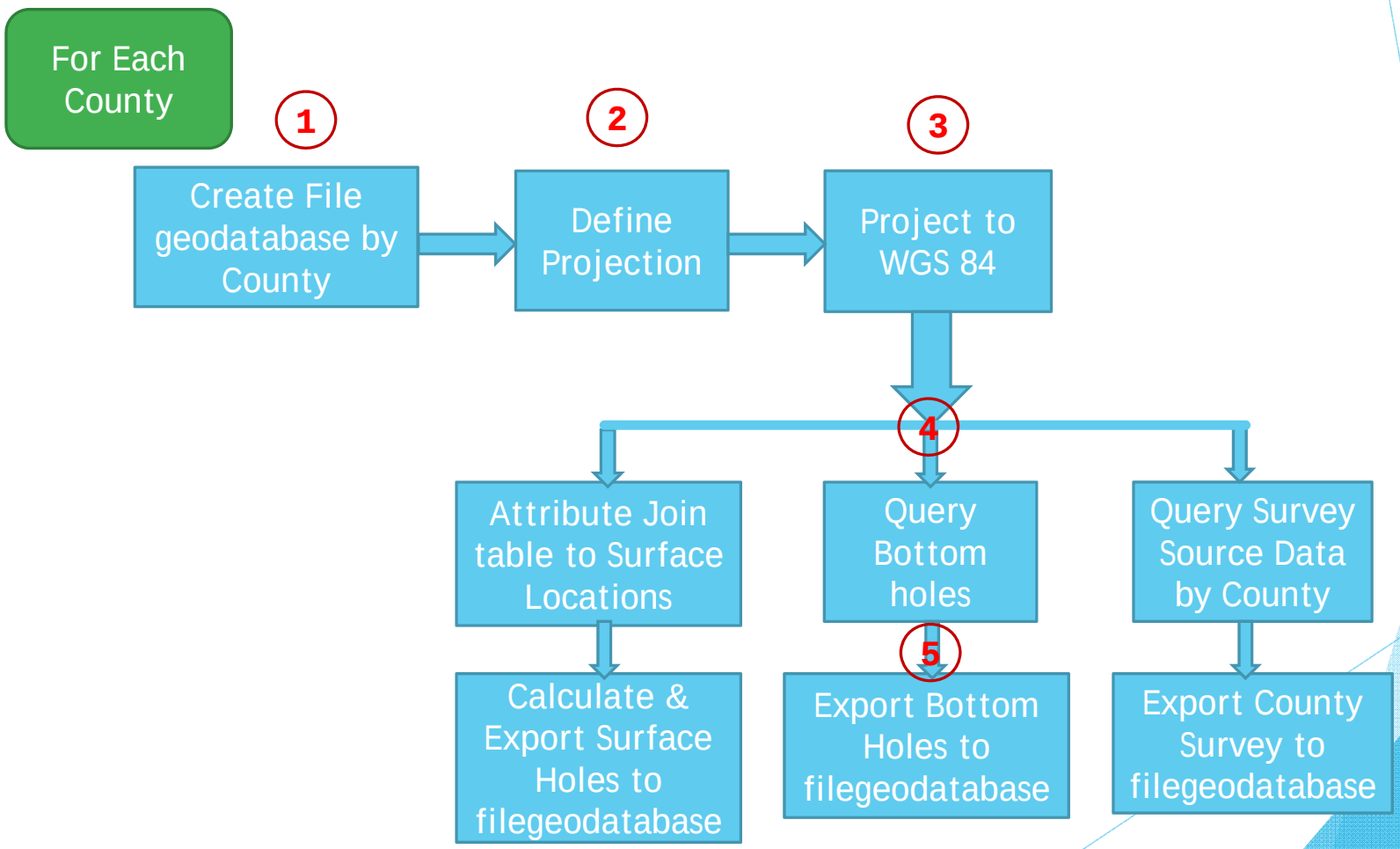
Source Data

- County wide surface well location, projection not defined
- County wide bottom hole location, projection not defined
- State wide Texas Original Land Survey, WGS 84

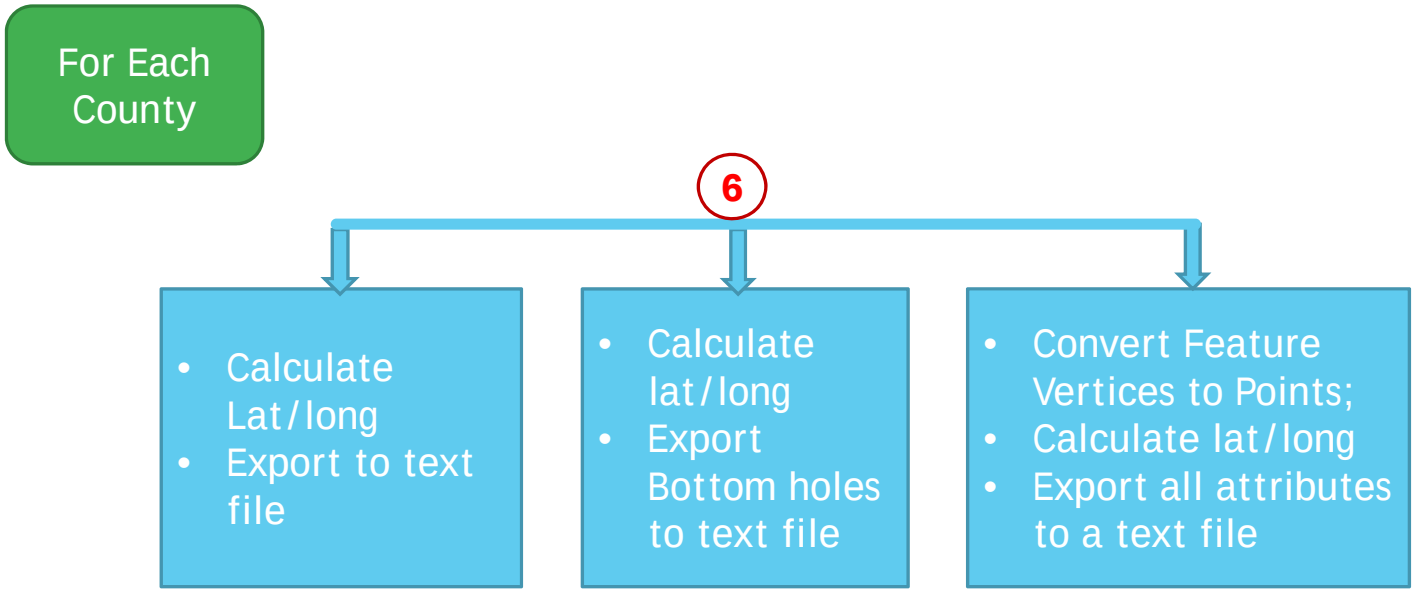
Goal

- For each county:
  - Create a file geodatabase
  - Define projection information for all layers (NAD 27 geographic)
  - Project all layers to WGS 84
  - Query and export the Original Texas Land Survey by county and export to the new database
  - Query and export surface & bottom hole well locations to file geodatabase
  - to create a text file

The workflow - I



The workflow - II



Coding in PyScripter

```
# Define_ReProject84_allTXwells.py
# ---------------------------------------------------------------------------
# Define_ReProject84_allTXwells.py
# Created on: 2015-03-13 16:02:06.00000
# Description:  Loop through all the folders in the current workspace, define all
# wells layers to NAD 27, AND project to WGS 84.
# ---------------------------------------------------------------------------
# Import arcpy module
import arcpy
import os
import shutil


current_folder = os.getcwd()
arcpy.env.overwriteOutput = True


Texas_Counties = { 'Dallam': '111','Sherman': '421','Hansford': '195','Ochiltree': '357','Lipscomb': '295','Hartley':
'205','Moore': '341','Hutchinson': '233','Roberts': '393','Hemphill': '211','Oldham': '359" }
```

```
# ############   Loop through the folders in the current workspace   ###################
for sub_folder in os.listdir(current_folder):
    if os.path.isdir(os.path.join(current_folder,sub_folder)):
        # get the fips #
        cnty_fips = Texas_Counties[sub_folder]

        # ###################  Beginning of file geodatabase & Survey layer   ############
        # 1. Create a geodatabase based in each folder
        # 2. Query Texas survey layer for each county (folder) and export to the database
        DB_path = os.path.join(current_folder,sub_folder)
        DB_name = sub_folder + "DB.gdb"   ##"HarrisDB.gdb"
        DB_absolute_path = os.path.join(DB_path, DB_name)

1       arcpy.CreateFileGDB_management(DB_path, DB_name,"CURRENT")
```

```
        # Process Texas Survey layer query and export
        Texas_SurveyLyr = "X:\Texas\TexasNew.gdb\\Layers\\SurveyNew"
        where_clause = "\"FIPS\" = ' " + cnty_fips + "'"
        arcpy.FeatureClassToFeatureClass_conversion(Texas_SurveyLyr, DB_absolute_path, "survey",where_clause)
        # ##################  End of file geodatabase & Survey layer   ########################################


        for shp in os.listdir(sub_folder):
            if shp.endswith('shp'):
                ## Define the shapefile as NAD27
                in_shp = os.path.join(current_folder,sub_folder,shp)
2               arcpy.DefineProjection_management(in_shp,
"GEOGCS['GCS_North_American_1927',DATUM['D_North_American_1927',SPHEROID['Clarke_1866',6378206.4,294.9786982]],PR
IMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]]")


                ## Process: Project the shapefile from NAD 27 to WGS 84
                shp_wgs84 = os.path.join(current_folder, sub_folder, os.path.splitext(shp)[0]+'_84.shp')

3               arcpy.Project_management(in_shp, shp_wgs84,
"GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],
UNIT['Degree',0.0174532925199433]]", "'NAD_1927_To_NAD_1983_NADCON + WGS_1984_(ITRF00)_To_NAD_1983'",
"GEOGCS['GCS_North_American_1927',DATUM['D_North_American_1927',SPHEROID['Clarke_1866',6378206.4,294.9786982]],PR
IMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]]")
```

```
# Import surface & bottom hole layers to the database
    # Create bottomhole84 featureclass
4   for shp in os.listdir(sub_folder):
        if shp.endswith('b_84.shp'):
            in_shp = os.path.join(current_folder,sub_folder,shp)
            where_clause = "\"APINUM\" like '%D%' OR " + "\"APINUM\" like '%H%'"

5           arcpy.FeatureClassToFeatureClass_conversion(in_shp,DB_absolute_path,"bottomholes84",
where_clause)
        # Surface well location
        if shp.endswith('s_84.shp'):
            in_shp = os.path.join(current_folder,sub_folder,shp)
            arcpy.FeatureClassToFeatureClass_conversion(in_shp,DB_absolute_path,"surface84")
```

6

```
for cnty in countyFolderNames:
    arcpy.env.workspace = "X:\\ABCDGISMapping\\Projects\\Data\\Texas\\" + cnty + "\\" + cnty + "DB" + ".gdb"
    print arcpy.env.workspace

    ## ########################################   CHANGE VALUE  ##########################
    mineralTracts = "bottomholes84"
    mineral_Tracts_pts = "Bottomholes84_pts"

    ## Convert features to points
    arcpy.FeatureVerticesToPoints_management(mineralTracts, mineral_Tracts_pts, "ALL")

    ## Defind field variables: orig_fid_fld field can be "ORIG_FID", or the_id -- internal id. Both INTEGER
    orig_fid_fld = "ORIG_FID"
    point_id_fld = "OBJECTID"
    mineral_id = "4"
    ##############  Other Wells layer fields  ########################################
    api_fld = "APINUM"
    status_fld = "SYMNUM"
```

6

```
## Add fields: long1 & lat1
    long_fld = "long1"
    lat_fld = "lat1"
    fieldPrecision = 12
    fieldScale = 8

    arcpy.AddField_management(mineral_Tracts_pts,long_fld,"DOUBLE", fieldPrecision, fieldScale)
    arcpy.AddField_management(mineral_Tracts_pts,lat_fld,"DOUBLE", fieldPrecision, fieldScale)

    ## Calculate long1, lat1
    arcpy.CalculateField_management(mineral_Tracts_pts, long_fld,"!SHAPE.EXTENT.XMAX!","PYTHON_9.3")
    arcpy.CalculateField_management(mineral_Tracts_pts, lat_fld,"!SHAPE.EXTENT.YMAX!", "PYTHON_9.3")

    ## Open tab delimited text file for writing
    outFileName = "X:\\ABCDGISMapping\\Projects\\Data\\Texas\\" + cnty + "\\Wells_b" + ".txt"
    outFile = open(outFileName,"w")
```

```
## Get unique values for field "ORIG_FID"
    uValues = [row[0] for row in arcpy.da.SearchCursor(mineral_Tracts_pts,(orig_fid_fld))]
    uniqueValues = set(uValues)

    writeStr = ""
    for a in uniqueValues:
        # Create an expression for SQL
        expression = "" + orig_fid_fld + "'= ' + str(a)
        with arcpy.da.SearchCursor(mineral_Tracts_pts,(point_id_fld,orig_fid_fld, api_fld, status_fld, long_fld, lat_fld), where_clause=expression) as rows:
            count = 0
            outStr = ""
            for row in rows:
                # write 6 columns of data to text file
                outStr += str(row[0]) + '\t' + str(mineral_id) + '\t' + str(row[1]) + '\t' + str(row[2]) + '\t' + str(row[3]) + '\t' + str(row[4]) + '\t' + str(row[5]) + '\t'+ str(count) + '\n'
                count = count + 1
        writeStr = writeStr + outStr
    outFile.write(writeStr)
    outFile.close()
print "Script completed successfully!“
```

25	4	25	4216134640H1	5	-96.1754505749	31.434026474	0	
26	4	26	4228931248D1	3	-96.1327490842	31.4322023324	0	
27	4	27	4228931882H1	2	-96.1553597986	31.4324618084	0	
28	4	28	4228931476HW	5	-96.1612455916	31.431364318	0	
29	4	29	4228931861D1	5	-96.1697175138	31.4290840528	0	
30	4	30	4228931476H1	8	-96.1639538658	31.428892544	0	
31	4	31	4228931000DW	5	-96.1920601142	31.4284391031	0	
32	4	32	4228931000D2	2	-96.1919457105	31.4284011634	0	
33	4	33	4228931000D1	5	-96.1919483833	31.4283561055	0	
303	8	1037	MC CORMACK, D W		-96.272723974	31.27208177	2	
304	8	1037	MC CORMACK, D W		-96.268696274	31.275645517	3	
305	8	1037	MC CORMACK, D W		-96.263072509	31.270935856	4	
306	8	1039	MC CARTY, J		-95.794815337	31.324137031	0	
307	8	1039	MC CARTY, J		-95.801135351	31.321042359	1	
308	8	1039	MC CARTY, J		-95.80205247	31.322407132	2	
309	8	1039	MC CARTY, J		-95.806245347	31.328564374	3	
310	8	1039	MC CARTY, J		-95.803214495	31.330107449	4	
311	8	1039	MC CARTY, J		-95.798735858	31.332429302	5	

THANK YOU