Instructions for Python script of Matagorda county\_08142017

1. This script is accompanied with AnnotMatrix.py, thus these two scripts must be in the same folder to run. For the former part of this script, it calls the script of AnnotMatrix.py, converting each row of field 'file\_as\_na','addr\_line1','addr\_line2','addr\_line3' to values in the Excel environment, the rule is as below: null = 0, name = 1, address = 2. Thus, the DBF file need to be converted to the Excel file in advance.
2. Import arcpy module. Calls the script of AnnotMatrix.py.
3. Set the environment. ATTN: The folder name can't begin from number in the path, or it will cause error.
4. Set local variables. Define each variable of which DBF file (dbf) and which Excel file (out\_xls) will work on. (Should be the same attribute table content!) Also, define A = Annotation(out\_xls), A will be used on the following script lines.
5. Add Fields in DBF file. For example, arcpy.AddField\_management(dbf, 'Owner\_Name', 'TEXT'): dbf means which DBF file is worked on, 'Owner\_Name' means the new field name we want to add, 'TEXT' means the data type of the new field. For this step, we need to add 4 new fields of TEXT type ('Owner\_Name', 'Owner\_Add'r, 'Prop\_Addr', 'CountyName') and 4 new fields of float type ('a', 'b', 'c', 'd'). The float type fields are the fields we will place the converted value of fields 'file\_as\_na','addr\_line1','addr\_line2','addr\_line3', they are just for the calculating process and will be deleted in the end of the script.
6. Create update cursor for feature class, convert each row of fields of 'Owner\_name' & 'Owner\_Addr' to: (0 = null, 1 = name, 2 = address) with AnnotMatrix.py.

import arcpy

from AnnotMatrix import Annotation

# Set environment settings

path = r"N:\Projects\Coastal County Parcel Data\Updates\Processed Data\Matagorda\Shapefiles"

arcpy.env.workspace = path

arcpy.env.overwriteOutput = True

# Set local variables

dbf = "NewMaskIntersect.dbf"

in\_table = dbf

out\_xls = "NewMaskIntersect.xls"

A = Annotation(out\_xls)

# Add Fields in dbf file

arcpy.AddField\_management(dbf, "Owner\_Name", "TEXT")

arcpy.AddField\_management(dbf, "Owner\_Addr", "TEXT")

arcpy.AddField\_management(dbf, "Prop\_Addr", "TEXT")

arcpy.AddField\_management(dbf, "CountyName", "TEXT")

arcpy.AddField\_management(dbf, "a", "float")

arcpy.AddField\_management(dbf, "b", "float")

arcpy.AddField\_management(dbf, "c", "float")

arcpy.AddField\_management(dbf, "d", "float")

fc = dbf

fields = ['a', 'b', 'c','d']

# Create update cursor for feature class, convert each row of fields of "Owner\_name" & "Owner\_Addr" to: (0 = null, 1 = name, 2 = address) with AnnotMatrix.py

with arcpy.da.UpdateCursor(fc, fields) as cursor:

index = 0

for row in cursor:

index = index + 1

row = A[index]

cursor.updateRow(row)

# Use update cursor to calculate each row of fields of "Owner\_name" & "Owner\_Addr"

fields2 = ['file\_as\_na','addr\_line1','addr\_line2','addr\_line3','Owner\_name','Owner\_Addr', 'a', 'b', 'c','d', 'addr\_city', 'addr\_state', 'zip']

#row 0 1 2 3 4 5 6 7 8 9 10 11 12

with arcpy.da.UpdateCursor(fc, fields2) as cursor:

for row in cursor:

if row[6] == 1: # 1 = name

row[4]= row[0]

if row[7] == 1:

if row[4] == '':

row[4]= row[1]

else:

row[4]= row[4]+' '+row[1]

if row[8] == 1:

if row[4] == '':

row[4]= row[2]

else:

row[4]= row[4]+' '+row[2]

if row[9] == 1:

if row[4] == '':

row[4]= row[3]

else:

row[4]= row[4]+' '+row[3]

if row[7] == 2: # 2 = address

row[5]= row[1]

if row[8] == 2:

if row[5] == '':

row[5]= row[2]

else:

row[5]= row[5]+', '+row[2]

if row[9] == 2:

if row[5] == '':

row[5]= row[3]

else:

row[5]= row[5]+', '+row[3]

if row[10] != ' ':

if row[5] == '':

row[5]= row[10]

else:

row[5]= row[5]+', '+row[10]

if row[11] != ' ':

if row[5] == '':

row[5]= row[11]

else:

row[5]= row[5]+', '+row[11]

if row[12] != ' ':

if row[5] == '':

row[5]= row[12]

else:

row[5]= row[5]+', '+row[12]

cursor.updateRow(row)

# Calculate each row of field of "Prop\_Addr"

fields3 = ['situs\_num','situs\_stre','situs\_st\_1','situs\_st\_2','situs\_city','situs\_stat','situs\_zip','Prop\_Addr']

#row 0 1 2 3 4 5 6 7

with arcpy.da.UpdateCursor(fc, fields3) as cursor:

for row in cursor:

if row[0] != ' ':

if row[0] != '0':

row[7]= row[0]

if row[1] != ' ':

if row[7] == '':

row[7]= row[1]

else:

row[7]= row[7]+' '+row[1]

if row[2] != ' ':

if row[7] == '':

row[7]= row[2]

else:

row[7]= row[7]+' '+row[2]

if row[3] != ' ':

if row[7] == '':

row[7]= row[3]

else:

row[7]= row[7]+' '+row[3]

if row[4] != ' ':

if row[7] == '':

row[7]= row[4]

else:

row[7]= row[7]+', '+row[4]

if row[5] != ' ':

if row[7] == '':

row[7]= row[5]

else:

row[7]= row[7]+', '+row[5]

if row[6] != ' ':

if row[7] == '':

row[7]= row[6]

else:

row[7]= row[7]+', '+row[6]

cursor.updateRow(row)

# Calculate each row of field of "CountyName"

field4 = ['CountyName']

#row 0

with arcpy.da.UpdateCursor(fc, field4) as cursor:

for row in cursor:

row[0] = "MATAGORDA"

cursor.updateRow(row)

# Execute DeleteField to delete fields of a,b,c,d

dropFields = ['a','b','c','d']

arcpy.DeleteField\_management(fc, dropFields)

# Tell the operator the process is finished

print 'The result is done.'

# The whole process may cost 3 mins

# Owner\_Addr Data no.9 & no.739 will manually revise because of the typo error (ex. 0(number:0)F should be O(letter:o)F)