Appendix 4-F

Python Programs to Create Annual Land Use

Python Programs to Create Annual Land Use

Appendix 5.2-B includes two Python programs that were used to expidite extraction of land use by model cell. The first Phython program performed overlay functions with groundwater and surface water irrigation layers dryland and rangeland layers, and then extracted and calculated the portions of specific land uses by county. These data became the GIS potential curve shown in the Appendix 5.2-D charts. The second python program used the county data and a vector based model cell layer to further partition the land use into 1 mile by 1 mile cells. These data were exported to a table where adjustment factors based on all data sources (GIS potential, NASS Census of Agriculture, remotely sensed data) were applied to calculate estimated actual irrigation at the cell scale.

```
# ---------------------------------------------------------------------------
# COHYSTluseCounty1950_2010.py
# Created: July 2nd, 2011
# Author: Amy Wright, Integrated Water Management Analyst, Nebraska
Department of Natural Resources (NDNR)

# #  What it does:
#    1. Creates yearly land use feature classes by updating a base
        layer (pre1950 SW fields, potential
#       dryland,grassland) with a transient GW irrigated fields and
        post 1950 SW fields layer
#    2. Creates a new attribute in yearly land use feature classes
        called "AcresYEAR" and calculates acres of
#       of each land cover by multiplying 0.000023 by the ESRI created
        Shape_area field.  This will convert
#       square feet (projection is state plane, feet) to acres.
#    3. Divides the Land use into zones (Counties) by using the
        Identity tool.
#    4. Utilizes Summary statistics to create a compacted table of
        land use by County, by year.
#    5. The Summary Statistics table is then appended to a
        comprehensive file geodatabase output table called
        CountyLuse1950_2011.


# Import system modules
import sys, string, os, arcgisscripting

# Create the Geoprocessor object
gp = arcgisscripting.create(9.3)

# Set the necessary product code
gp.SetProduct("ArcInfo")

# Load required toolboxes...
```

```
gp.AddToolbox("C:/Program Files (x86)/ArcGIS/ArcToolbox/Toolboxes/Data
Management Tools.tbx")
gp.AddToolbox("C:/Program Files (x86)/ArcGIS/ArcToolbox/Toolboxes/Analysis
Tools.tbx")

### Set working space, change working directory here

gp.Workspace =
"C:\\azoller\\Projects\\Cohyst\\LandUse\\IrrigatedAcres\\AllNRDs\\COHYSTacres
Processing.gdb"
wkspc = gp.Workspace
# Allow overwrite an existing file
gp.OverwriteOutput = True

# Set variables
inFiles = wkspc + "\\Inputs"
outFiles = wkspc + "\\CountyOutputs"
baseLayer = inFiles + "\\CountyBasePre1950"
timeSeries = inFiles + "\\CountyTimeSeries"
timeSeriesComingled = "\\CountyTimeSeriesComingled"
CountyZones = inFiles + "\\CountyZones"

# Create a loop that creates new feature classes for each year between # 1950
and 2010.  The shapefiles contain polygon features that
# represent multiple irrigated fields per county (i.e. multi-part #
features), along with attributes to explain the summed number of irrigated
acres for
# that county.

yr = 1950
while yr < 2012:

    # Time series variables
    LuseByYr = outFiles + "\\COHYSTdryswgw_" + str(yr)
    LuseByYrComingled = outFiles + "\\COHYSTlc_" + str(yr)
    acresField = "Acres" + str(yr)
    StatsByYr = wkspc  + "\\CountyLC" + str(yr) + "stats"
    StatsByYrQA = wkspc  + "\\CountyLC" + str(yr) + "statsQA"
    timeOut = outFiles + "\\COHYSTswgw" + str(yr)
    timeOutComingled = outFiles + "\\COHYSTcomingled" + str(yr)
    CountyTimeOut = outFiles + "\\CountyLC" + str(yr)
    CountyOutputAll = wkspc + "\\CountyLC1950_2011stats"
    CountyOutputAllQA = wkspc + "\\CountyLC1950_2011statsQA"

    # Make a layer of the County Time Series feature class
    # Then, Select Layer By Attributes using the year, and make a copy of the
    selected features
    timeLayer = inFiles + "\\select" + str(yr)
    gp.MakeFeatureLayer(timeSeries, timeLayer)
    selCondition = "IrrYr <= " + str(yr)
    gp.SelectLayerByAttribute_management(timeLayer, "NEW_SELECTION",
    selCondition)
    gp.CopyFeatures(timeLayer, timeOut)

    timeLayerComingled = inFiles + "\\selectComingled" + str(yr)
    gp.MakeFeatureLayer(timeSeriesComingled, timeLayerComingled)
    selCondition = "IrrYr <= " + str(yr)
    gp.SelectLayerByAttribute_management(timeLayerComingled, "NEW_SELECTION",
    selCondition)
    gp.CopyFeatures(timeLayerComingled, timeOutComingled)

    # Process: Update...
    gp.repairgeometry_management(timeOut)
    gp.Update_analysis(baseLayer, timeOut, LuseByYr, "BORDERS", "")
    gp.Update_analysis(LuseByYr, timeOutComingled, LuseByYrComingled,
    "BORDERS", "")

    # Process: Calculate Field...
    gp.CalculateField_management(LuseByYrComingled, "GISAcres", "[Shape_Area]
    * 0.000023", "VB", "")

    # Use Identity to create zones of counties to tabulate areas for
    gp.Identity_analysis(LuseByYrComingled, CountyZones, CountyTimeOut)
    gp.CalculateField_management(CountyTimeOut, "GISacres", "[Shape_Area] *
    0.000023", "VB", "")

    # Process: Summary Statistics...
    gp.Statistics(CountyTimeOut, StatsByYr, "GISacres SUM",
    "County;LCclass;LCcode")

    # Add a year and acres field to output stats table, and populate with
    year of analysis, and associated acres
    gp.AddField_management (StatsByYr,"Year", "short")
    gp.CalculateField_management (StatsByYr, "Year", yr, "VB")

    # Process: Summary Statistics for QA...
    gp.Statistics(StatsByYr, StatsByYrQA, "SUM_GISacres SUM", "County;Year")

    # Append individual tables to a table that contains all data
    gp.Append_management(StatsByYr, CountyOutputAll)
    gp.Append_management(StatsByYrQA, CountyOutputAllQA)

    # go on to next year
    print "moving on to next year"
    yr = yr + 1

del gp

# ----------------------------------------------------------------------


##    COHYSTluseGrid_1950_2010
##    Created July 10, 2011
##    Author:  Amy Wright, NDNR
##    What it does:  This script takes land use outputs from
COHYSTluseCounty1950_2010.py, and
##    splits the vector features into 160 acre grid cells.  From this, acres
of SW, GW, comingled
##    irrigated lands, as well as dryland and grassland acres are calculated
for each grid cell, for
##    each year, and the output is a table for each year.  Since there may be
more than one
##    polygon for each land cover, unique land covers are summarized for each
grid cell, so there is one
##    and only one land cover row for each grid cell.  The yearly output
tables are merged into one
##    combined output called GridLC1950_2011all.

# Import system modules
import sys, string, os, arcgisscripting

# Create the Geoprocessor object
gp = arcgisscripting.create(9.3)

# Set the necessary product code
gp.SetProduct("ArcInfo")

# Load required toolboxes...
```

```
gp.AddToolbox("C:/Program Files (x86)/ArcGIS/ArcToolbox/Toolboxes/Data
Management Tools.tbx")
gp.AddToolbox("C:/Program Files (x86)/ArcGIS/ArcToolbox/Toolboxes/Analysis
Tools.tbx")

### Set working space************************Change working directory here,
if necessary***************************************
gp.Workspace =
"C:\\azoller\\Projects\\Cohyst\\LandUse\\IrrigatedAcres\\AllNRDs\\COHYSTacres
_GISprocessing.gdb"
wkspc = gp.Workspace
# Allow overwrite an existing file
gp.OverwriteOutput = True

# Set variables
inFiles = wkspc + "\\Inputs"
outFiles = wkspc + "\\GridOutputs"
outFiles2 =
"C:\\azoller\\Projects\\Cohyst\\COHYSTacres_GISprocessing\\GridOutputs.gdb"
baseLayer = inFiles + "\\GridBasePre1950"
baseLayers = wkspc + "\\TestMultiPart\\GridBasePre1950s"
timeSeries = inFiles + "\\GridTimeSeries"
timeSeriesComingled = "\\GridTimeSeriesComingled"
GridZones = inFiles + "\\GridZones"

# Create a loop that creates new feature classes w/ divided land cover by
grid cell
# for each year between 1950 and 2010.

yr = 1950
while yr < 2011:

    # Time series variables
    timeOut = outFiles2 + "\\COHYSTGRIDswgw" + str(yr)
    # timeOuts = outFiles + "\\COHYSTGRIDswgws" + str(yr)
    timeOutComingled = outFiles2 + "\\COHYSTGRIDcomingled" + str(yr)
    LuseByYr = outFiles2 + "\\COHYSTGRIDdryswgw_" + str(yr)
    LuseByYrComingled = outFiles2 + "\\COHYSTGRIDallLC_" + str(yr)
    acresField = "Acres" + str(yr)


    GridLCbyYr = outFiles2 + "\\GridLC_" + str(yr)
    GridStatsByYr = wkspc  + "\\GridLC" + str(yr) + "stats"
    GridStatsByYrQA = wkspc  + "\\GridLC" + str(yr) + "statsQA"
    GridOutputAll = wkspc + "\\GridLC1950_2011stats"
    GridOutputAllQA = wkspc + "\\GridLC1950_2011statsQA"


    # Make a layer of the County Time Series feature class
    # Then, Select Layer By Attributes using the year, and make a copy
    of the selected features
    timeLayer = inFiles + "\\select" + str(yr)
    gp.MakeFeatureLayer(timeSeries, timeLayer)
    selCondition = "IrrYr <=" + str(yr)
    gp.SelectLayerByAttribute_management(timeLayer, "NEW_SELECTION",
    selCondition)
    gp.CopyFeatures(timeLayer, timeOut)

    timeLayerComingled = inFiles + "\\selectComingled" + str(yr)
    gp.MakeFeatureLayer(timeSeriesComingled, timeLayerComingled)
    selCondition = "IrrYr <= " + str(yr)
    gp.SelectLayerByAttribute_management(timeLayerComingled,
    "NEW_SELECTION", selCondition)
    gp.CopyFeatures(timeLayerComingled, timeOutComingled)
    # Process: Update...
    # gp.MultipartToSinglePart_management(timeOut, timeOuts)
    gp.Update_analysis(baseLayers, timeOut, LuseByYr, "BORDERS", "")
    gp.Update_analysis(LuseByYr, timeOutComingled, LuseByYrComingled,
    "BORDERS", "")

    # Process: Calculate Field...
    gp.CalculateField_management(LuseByYrComingled, "GISAcres",
    "[Shape_Area] * 0.000023", "VB", "")


    # Use Identity to create zones of counties to tabulate areas for
    gp.Identity_analysis(LuseByYrComingled, GridZones, GridLCbyYr)
    gp.CalculateField_management(GridLCbyYr, "GISacres", "[Shape_Area]
    * 0.000023", "VB", "")

    # Process: Summary Statistics...
    gp.Statistics(GridLCbyYr, GridStatsByYr, "GISacres SUM",
    "RowCol_ID;County;LCclass;LCcode")

    # Add a year and acres field to output stats table, and populate
    with year of analysis, and     # associated acres
    gp.AddField_management (GridStatsByYr,"Year", "short")
    gp.CalculateField_management (GridStatsByYr, "Year", yr, "VB")

    # Process: Summary Statistics for QA...
    gp.Statistics(GridStatsByYr, GridStatsByYrQA, "SUM_GISacres SUM",
    "RowCol_ID;Year")

    # Append individual tables to a table that contains all data
    gp.Append_management(GridStatsByYr, GridOutputAll)
    gp.Append_management(GridStatsByYrQA, GridOutputAllQA)

    # go on to next year
    print "moving on to next year"
    yr = yr + 1

del gp
```