

Exercise 2: Batch Processing using ArcPy Listing

Often, the first step when starting a new project, after acquiring data, is to clip that data (that may have widely varying extents) to a common area – a study area. Instead of repeating the clip process for each feature class, you can use ArcPy listing functions to do the same batch processing work in only a few lines of code.

Examine data

- Navigate to folder ...Python_ArcGIS\exercises\exercise2_listing
- Double-click Habitat_Analysis.mxd to open it in ArcMap, and examine the data layers that are available in the table of contents
- This map is similar to the starting map for Exercise 1, but with a new layer, **Study Area**, which represents a special area that we want to limit our analysis to
- The feature classes needed to complete this exercise are stored at ...Python_ArcGIS\exercises\exercise2_listing\data\Habitat_Analysis.gdb
- Close ArcMap

Setting up the script

- Open PythonWin
- Click the New button and create a new Python script



- The first thing to do is to write some comments; summary, author, date, etc

```
# Script to prepare for project by clipping all data to study area
# Created by San Diego Gnatcatcher Conservation Fund
# On March 15, 2010
```

- Import arcpy and os modules

```
import arcpy
import os
```

- Set the geoprocessing environments workspace and overwriteOutput

```
arcpy.env.workspace = os.path.join(os.path.dirname(os.getcwd()), "data", "Habitat_Analysis.gdb")
arcpy.env.overwriteOutput = True
```

Iterate through feature classes

In this case it was **necessary** to set the geoprocessing workspace, because the ArcPy listing functions will return the data stored in that directory or geodatabase.

- Use the ListFeatureClasses() function to create a list of all feature classes in the workspace

```
fclist = arcpy.ListFeatureClasses()
```

- Optionally, create a new Feature Dataset to store the new copies of the feature classes

```
arcpy.CreateFeatureDataset_management(arcpy.env.workspace, "Clipped")
```

- Iterate through each feature class in that list and clip it to the "StudyArea" feature class using the Clip_analysis tool

- The Clip_analysis tool has three required parameters: Input, Clip Features, and Output
- Use logic to skip clipping the study area by itself

```
for fc in fclist:
    if fc != "StudyArea":
        arcpy.Clip_analysis(fc, "StudyArea", "Clipped/%s_clip" % fc)
```

- Click the Save button and save this script to the directory ...Python_ArcGIS\exercises\exercise1_gp\scripts
- Run the script

Results

With only a few short lines of code you can perform batch processing on any number of datasets.

