Create your own geoprocessing tools!

Trevor Laureys
Director – GIS & Data Analysis
Indiana State Dept. of Agriculture
Automated Report Generation using ArcGIS Pro, ArcPy MapSeries Module and Python

LAST YEAR WE COVERED ARCPY.MP

Meg Leader
Director of Soil Health
INFA State Coordinator

Trevor Laureys
Program Manager Data Analytics
Indiana has a unique situation

Tracking ALL ICP cost-shared practices since 2013

ICP Conservation Database includes

- NRCS Applied Conservation Practices/Agronomic
- SWCD Applied Conservation Practices
- ISDA-DSC & CWI Applied Conservation Practices
- DNR-LARE Applied Conservation Practices
- IDEM-319 Applied Conservation Practices

ONLY state with a comprehensive data sharing agreement with our federal partners

Over 73,000 Region 5 Modeled practices since 2013!
Tracking progress -

Key Regulatory Challenge:
Water Quality as it relates to Nutrients and Non-point Source Runoff
Tracking allows the ICP to monitor the success of cost-share programs in various geographic regions.

All conservation data is stored in points geodatabase.

Some parameters tracked include:
- Practice Name
- Load Reductions
- UTM XY
- Applied Date
- Program
- Size

We can defend the work we do as conservationists as well as the work of private landowners.
A WHOLE LOT OF DATA
“Geoprocessing is a framework and set of tools for processing geographic and related data.”

-Sir Esri Websitatingon
Geoprocessing Tools

I’ll cover how we can create custom geoprocessing tools using:

> Model Builder Method
> Python Method

May touch on:
> Command Line
> ArcToolBox Basics

DISCLAIMER: This is geared towards novice users who have some GIS experience, but may not have ventured outside the normal toolboxes.

Some of you may be experts, and may not benefit from this presentation.
Why create your own tools?

Create toolboxes for workflows that one must do repetitively

Creating tools for users who aren’t familiar with the software

Skip UI for certain tools and save time!

Insert persuasion.txt
Model Builder Approach

Simple example workflow from Trev’s life: Cropping & land histories for different watersheds/political boundaries

**Arctoolbox workflow**

- Export feature
- Locate CDL Raster
- Raster Clip + UI
- Add Field + UI
- Calculate Field + UI
- Table to table + UI

Multiplied by # of Years of interest going back ~10 years
Model Builder Approach cont.
Model Builder Approach cont.

[Image of Geoprocessing Options dialog box with options for General, Background Processing, Script Tool Editor/Debugger, ModelBuilder, Results Management, and Display / Temporary Data.]
Model Builder Approach cont.

Geoprocessing > ModelBuilder

Analysis Tab > ModelBuilder
Model Builder Approach cont.

Data management > raster > raster processing > clip
Data management > fields > add field
Data management > fields > calculate field
Conversion Tools > to geodatabase > table to table
Model Builder Approach cont.
Model Builder Approach cont.
This tool has no parameters.
Model Builder Approach cont.
Model Builder Approach cont.
Model Builder Approach cont.

iterate_raster_clip:

Hey guys, happy GIS day
Model Builder Approach cont.
Model Builder Approach cont.
Model Builder Approach cont.
Python Script Approach

Arctoolbox workflow

- Export feature
- Locate NHD Stream Data
- Clip
- Buffer
- Clip
Python Script Approach
```python
import arcpy

# set parameters
Clip_Boundary = arcpy.GetParameterAsText(0)
CY_Practices = arcpy.GetParameterAsText(1)
Distance_value_or_field = arcpy.GetParameterAsText(2)
Stream_Clip_Name = arcpy.GetParameterAsText(3)
Stream_Buffer_Name = arcpy.GetParameterAsText(4)

# variables:
NHDFlowline_shp = "D:\\NED\NHDFlowline.shp"
RiparianPractices = "D:\\Presentations\\Geoprocessing\\Geoprocessing.gdb\\RiparianPractices"

# Process: Clip, buffer, clip
arcpy.Clip_analysis(NHDFlowline_shp, Clip_Boundary, Stream_Clip_Name, "")

arcpy.AddField(""Clippping by watershed"")

arcpy.Buffer_analysis(Stream_Clip_Name, Stream_Buffer_Name, Distance_value_or_field, "FULL", "ROUND", "NONE", ",", "PLANAR")

arcpy.AddField("Buffering streams")

arcpy.Clip_analysis(CY_Practices, Stream_Buffer_Name, RiparianPractices, ")

arcpy.AddField("Clippping practices")
```
Python Script Approach
Python Script Approach
import arcpy

# set parameters
Clip_Boundary = arcpy.GetParameterAsText(0)
CY_Practices = arcpy.GetParameterAsText(1)
Distance_value_or_field = arcpy.GetParameterAsText(2)
Stream_Clip_Name = arcpy.GetParameterAsText(3)
Stream_Buffer_Name = arcpy.GetParameterAsText(4)

# variables:
NHDFlowline_shp = "D:\\NHM\NHDFlowline.shp"
RiparianPractices = "D:\\Presentations\\Geoprocessing\\Geoprocessing.gdb\\RiparianPractices"

# Process: Clip, buffer, clip
arcpy.Clip_analysis(NHDFlowline_shp, Clip_Boundary, Stream_Clip_Name, "")
arcpy.Buffer_analysis(Stream_Clip_Name, Stream_Buffer_Name, Distance_value_or_field, "FULL", "ROUND", "NONE", "")
arcpy.AddMessage("Clipping by watershed")
arcpy.Clip_analysis(CY_Practices, Stream_Buffer_Name, RiparianPractices, "")
arcpy.AddMessage("Clipping practices")
Python Script Approach

```python
import arcpy

class ToolValidator(object):
    """Class for validating a tool’s parameter values the behavior of the tool’s dialog."""

    def __init__(self):
        """Setup arcpy and the list of tool parameters. self.params = arcpy.GetParameterInfo()"""

    def InitializeParameters(self):
        """Define the properties of a tool’s parameters called when the tool is opened."""
        return

    def UpdateParameters(self):
        """Modify the values and properties of parameter validation is performed. This method is called after internal return"""
        return

    def UpdateMessages(self):
        """Modify the messages created by internal validation is performed. This method is called after internal return"""
        return
```
Python Script Approach
GIS

What my friends think I do

What my mom thinks I do

What society thinks I do

What my clients think I do

What I think I do

What I really do
Possibilities. Promise. Purpose.

CONTACT ME

Trevor Laureys
EMAIL
Tlaureys@isda.in.gov

PHONE NUMBER
317.476.1106