Building Geoprocessing Tools

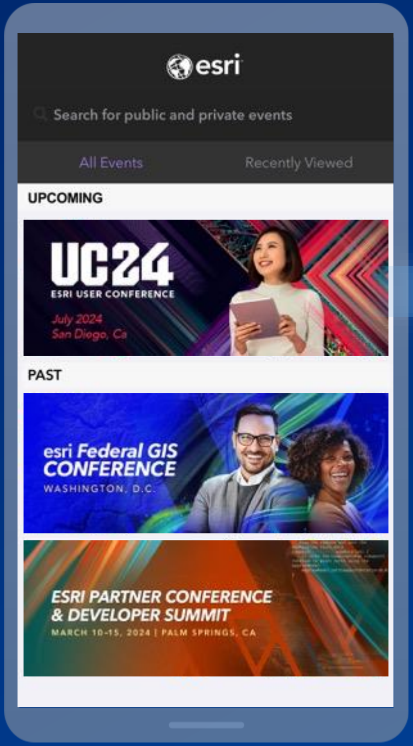
Dave Wynne, Sean Lim

ESRI USER CONFERENCE 2024

Copyright © 2024 Esri.
All rights reserved.

Please share your feedback in the app

Download the Esri Events app and find your event



esri

Search for public and private events

All Events | Recently Viewed

UPCOMING

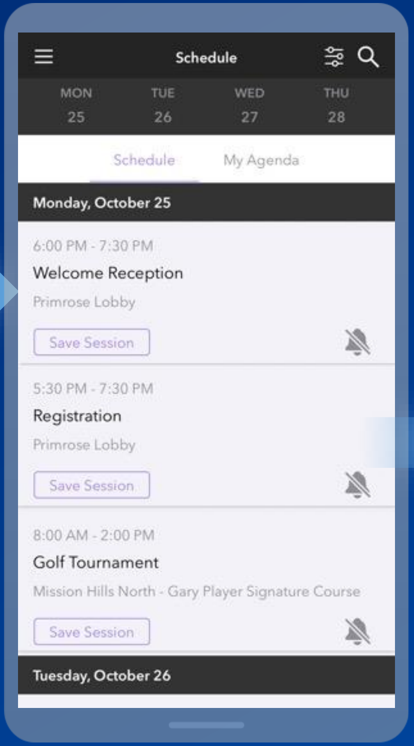
UC24 ESRI USER CONFERENCE July 2024 San Diego, Ca

PAST

esri Federal GIS CONFERENCE WASHINGTON, D.C.

ESRI PARTNER CONFERENCE & DEVELOPER SUMMIT MARCH 10-15, 2024 | PALM SPRINGS, CA

Select the session that you attended



Schedule

MON	TUE	WED	THU
25	26	27	28

Schedule | My Agenda

Monday, October 25

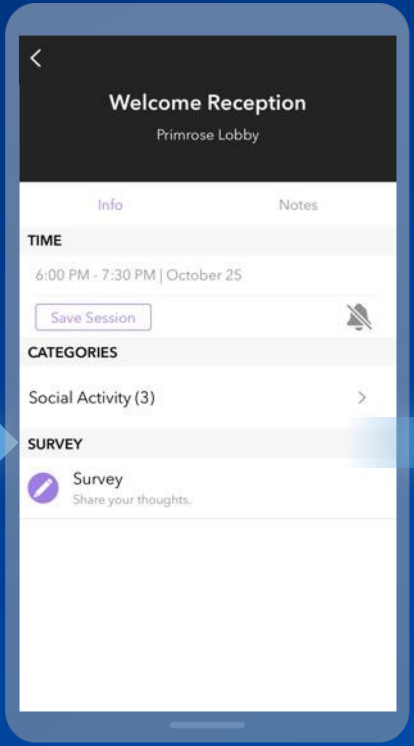
6:00 PM - 7:30 PM
Welcome Reception
Primrose Lobby
Save Session

5:30 PM - 7:30 PM
Registration
Primrose Lobby
Save Session

8:00 AM - 2:00 PM
Golf Tournament
Mission Hills North - Gary Player Signature Course
Save Session

Tuesday, October 26

Scroll down to "Survey"



Welcome Reception
Primrose Lobby

Info | Notes

TIME

6:00 PM - 7:30 PM | October 25

Save Session

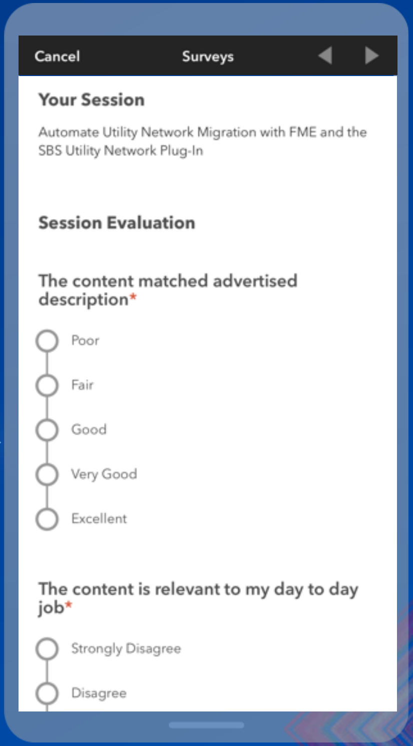
CATEGORIES

Social Activity (3)

SURVEY

Survey
Share your thoughts.

Log in to access the survey



Cancel | Surveys

Your Session

Automate Utility Network Migration with FME and the SBS Utility Network Plug-In

Session Evaluation

The content matched advertised description*

○ Poor
○ Fair
○ Good
○ Very Good
○ Excellent

The content is relevant to my day to day job*

○ Strongly Disagree
○ Disagree

Agenda

- Introduction to tools and toolboxes

- Parameters
- Source code
- Validation
- Sharing your tools

- Questions

Building Geoprocessing Tools

Slides
and demo tools

https://esriurl.com/tool_workshop_2024

Introduction to tools and toolboxes

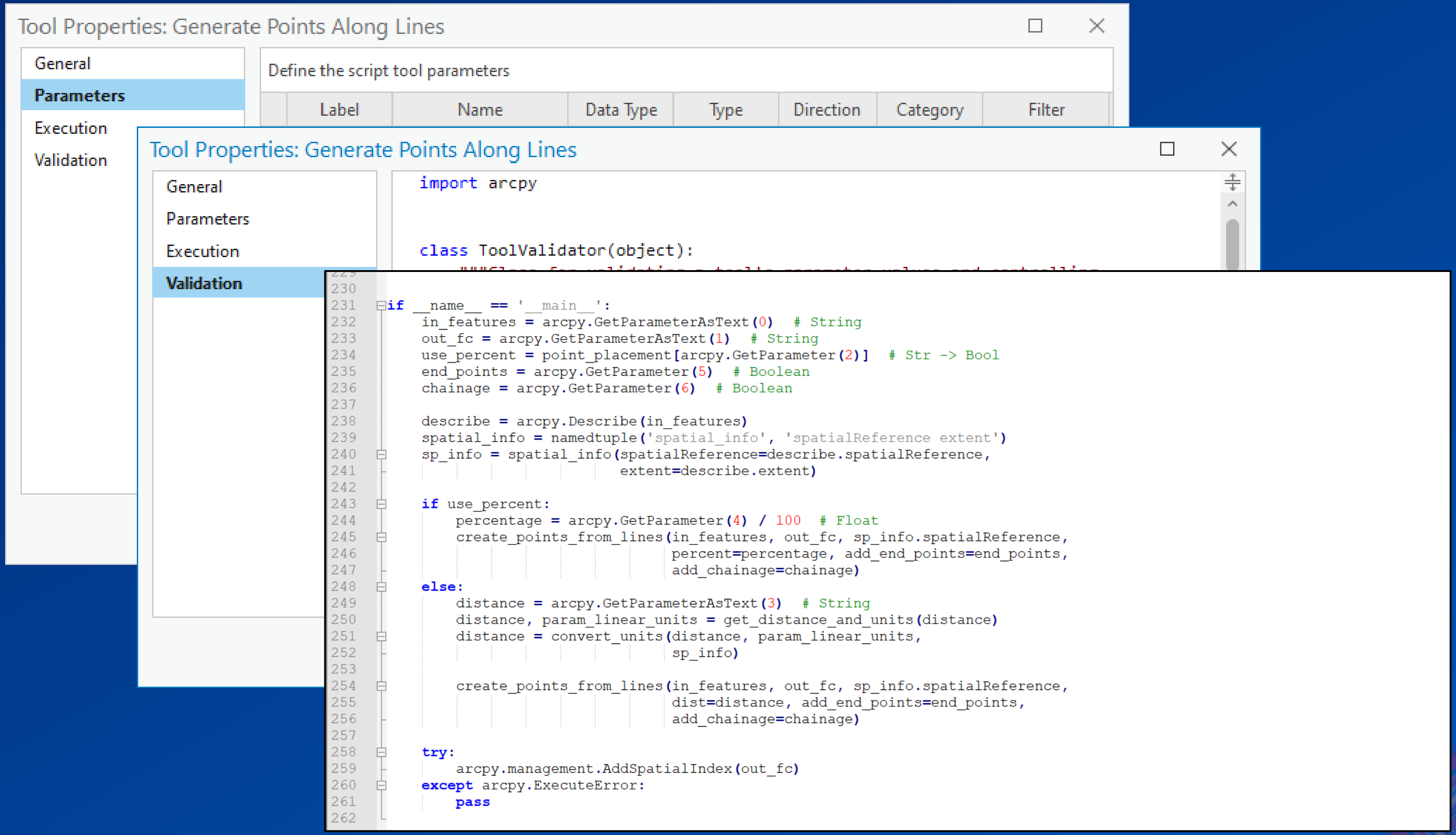
What is a script tool?

- Geoprocessing tool based on your Python code
- Looks and behaves like a built-in tool
- Use in multiple ways
  - Geoprocessing pane
  - ArcPy
  - ModelBuilder
  - A geoprocessing service
- Share with others

Anatomy of a tool

- Parameters
- Validation code
- Source code

- *And*
  - *High-level details*
  - *Documentation*


Tool Properties: Generate Points Along Lines
General
Parameters
Execution
Validation

Define the script tool parameters

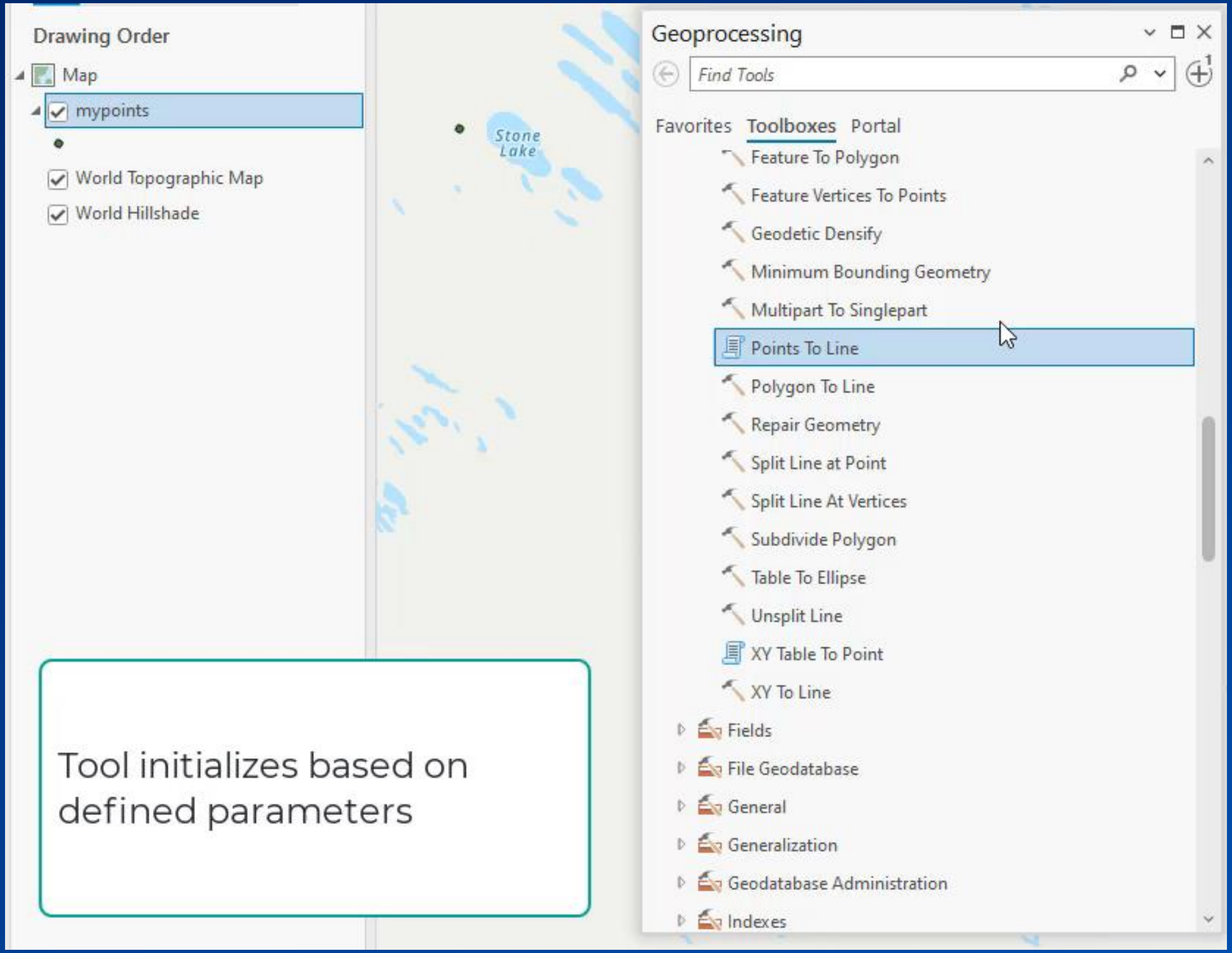
Label	Name	Data Type	Type	Direction	Category	Filter

Tool Properties: Generate Points Along Lines
General
Parameters
Execution
Validation

```python
import arcpy


class ToolValidator(object):
```

```python

if __name__ == '__main__':
    in_features = arcpy.GetParameterAsText(0)  # String
    out_fc = arcpy.GetParameterAsText(1)  # String
    use_percent = point_placement[arcpy.GetParameter(2)]  # Str -> Bool
    end_points = arcpy.GetParameter(5)  # Boolean
    chainage = arcpy.GetParameter(6)  # Boolean

    describe = arcpy.Describe(in_features)
    spatial_info = namedtuple('spatial_info', 'spatialReference extent')
    sp_info = spatial_info(spatialReference=describe.spatialReference,
                           extent=describe.extent)

    if use_percent:
        percentage = arcpy.GetParameter(4) / 100  # Float
        create_points_from_lines(in_features, out_fc, sp_info.spatialReference,
                                 percent=percentage, add_end_points=end_points,
                                 add_chainage=chainage)
    else:
        distance = arcpy.GetParameterAsText(3)  # String
        distance, param_linear_units = get_distance_and_units(distance)
        distance = convert_units(distance, param_linear_units,
                                 sp_info)

        create_points_from_lines(in_features, out_fc, sp_info.spatialReference,
                                 dist=distance, add_end_points=end_points,
                                 add_chainage=chainage)

    try:
        arcpy.management.AddSpatialIndex(out_fc)
    except arcpy.ExecuteError:
        pass

```

How a tool works



Tool initializes based on defined parameters

Toolboxes

- Tools are organized in a toolbox
- We can build Python-based tools in 3 formats:
  - ArcGIS toolbox (.atbx)
  - Legacy toolbox (.tbx)
  - Python toolbox (.pyt)

Pick your toolbox format

	Legacy toolbox	**ArcGIS toolbox**	**Python toolbox**
First release	Desktop 9.0	Pro 2.9	Desktop 10.1
Notable fact	Mature (*but still supported*)	Next generation	All Python
Format	Binary	Compressed JSON-based format	Python in a .pyt file
Why?	Original	Better cross-release compatibility & persistence	All Python *(can create entirely in any editor)*
Supports script tools	Yes	Yes	Yes
Supports model tools	Yes	Yes	No

Parameters

Parameters

https://esriurl.com/gp_define_parameters

- Parameters are how you interact with a tool
- Parameters provide simple rules
  - Does an input exist?
  - Is the input the right type?
  - Is this value an expected keyword?

Parameter properties

Label	Filter
Name	Dependency
Data type	Default
Multiple values	Environment
Type	Symbology
Direction	Enabled
Category	Display order
	Control

Parameters

https://esriurl.com/gp_define_parameters

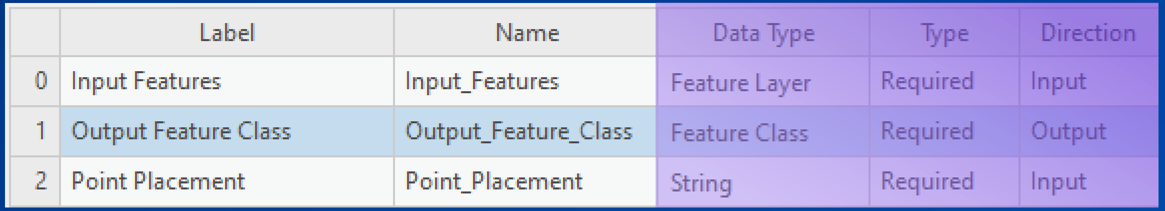
- Parameters are how you interact with a tool
- Parameters provide simple rules
  - Does an input exist?
  - Is the input the right type?
  - Is this value an expected keyword?

Parameter properties

Label	Filter
Name	Dependency
Data type	Default
Multiple values	Environment
Type	Symbology
Direction	Enabled
Category	Display order
	Control

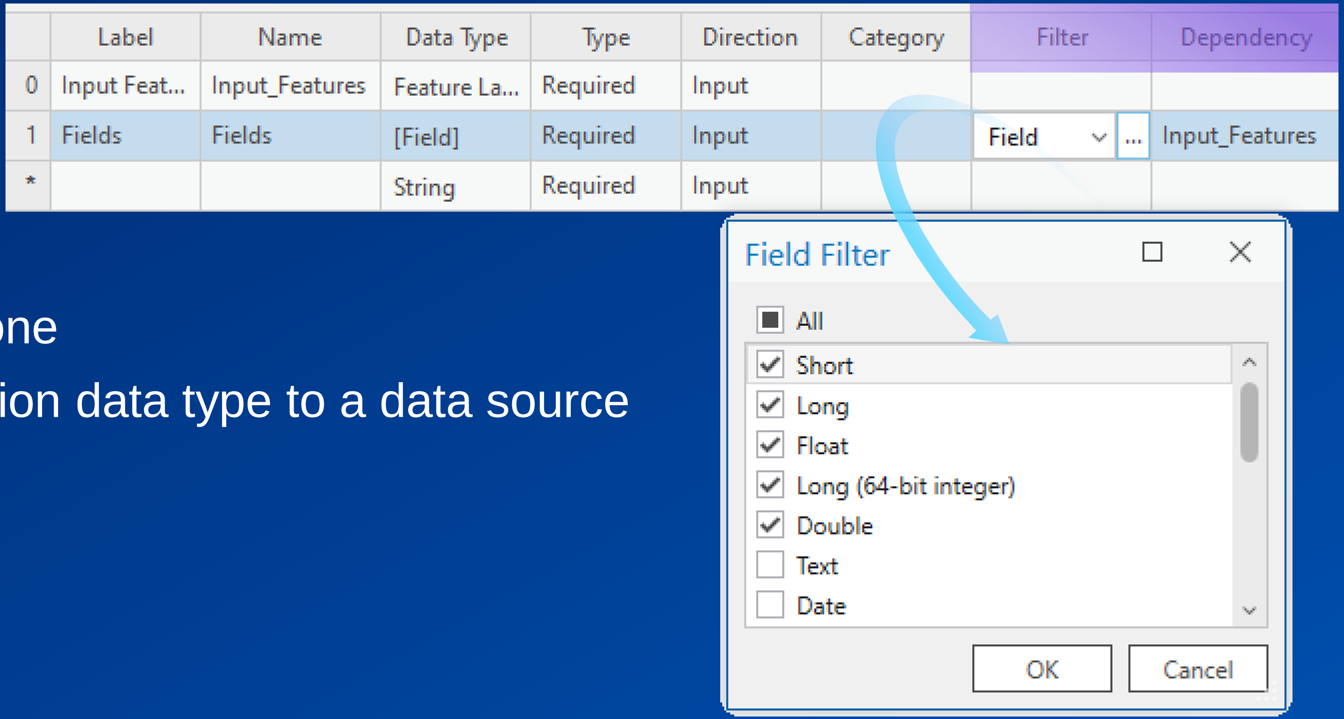
Parameter properties

- Data type
  - The type of value the parameter will accept
  - More than 160 options
- Direction
  - Input or output
- Type
  - Required, Optional, or Derived
  - Derived—A value returned by the tool



	Label	Name	Data Type	Type	Direction
0	Input Features	Input_Features	Feature Layer	Required	Input
1	Output Feature Class	Output_Feature_Class	Feature Class	Required	Output
2	Point Placement	Point_Placement	String	Required	Input

Dependencies and filters



	Label	Name	Data Type	Type	Direction	Category	Filter	Dependency
0	Input Feat...	Input_Features	Feature La...	Required	Input			
1	Fields	Fields	[Field]	Required	Input		Field ...	Input_Features
*			String	Required	Input			

Field Filter

■ All

☑ Short

☑ Long

☑ Float

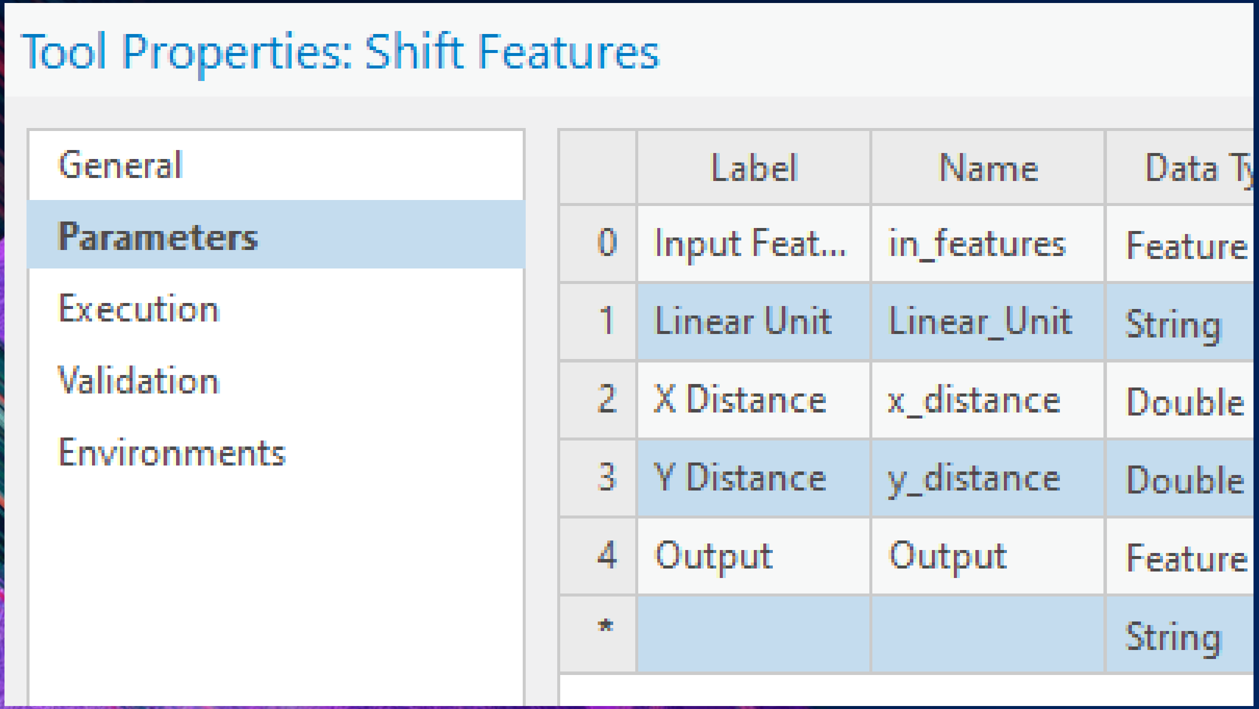
☑ Long (64-bit integer)

☑ Double

□ Text

□ Date

- Dependency
  - Associate a parameter with another one
  - For example, a Field or SQL Expression data type to a data source
- Filter
  - Limit options by value or by type



	Label	Name	Data Ty
0	Input Feat...	in_features	Feature
1	Linear Unit	Linear_Unit	String
2	X Distance	x_distance	Double
3	Y Distance	y_distance	Double
4	Output	Output	Feature
*			String

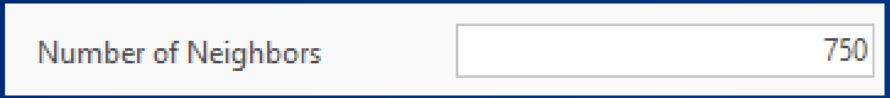
Demo: Parameters

https://esriurl.com/shift_features_tool

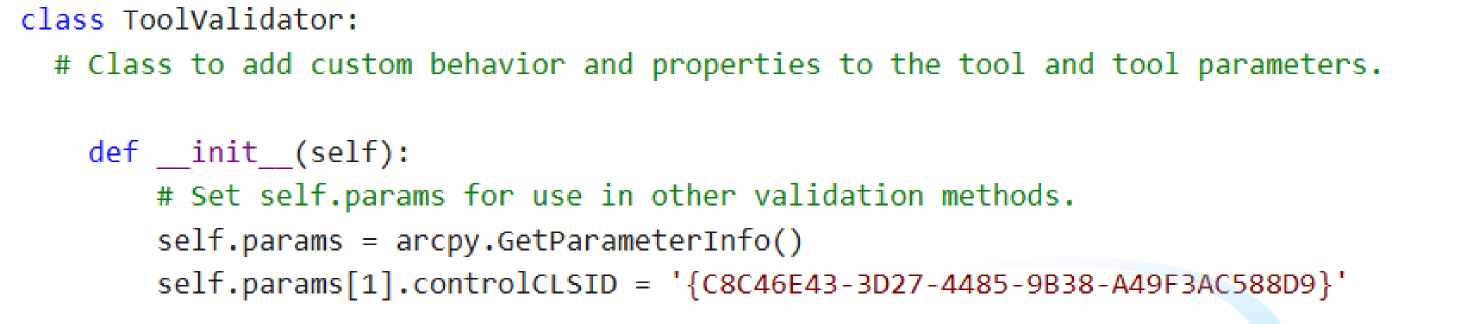
Changing a parameter’s control

https://esriurl.com/gp_controls

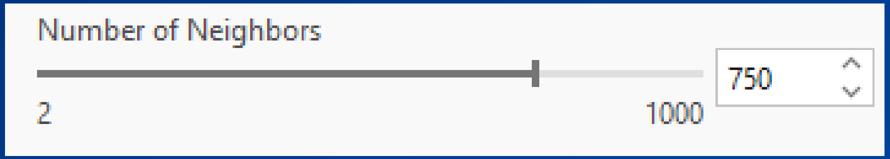
- For some parameters, you can change the default appearance
- Use the controlCLSID property with a GUID-like string



Number of Neighbors 750



```
class ToolValidator:
  # Class to add custom behavior and properties to the tool and tool parameters.

    def __init__(self):
        # Set self.params for use in other validation methods.
        self.params = arcpy.GetParameterInfo()
        self.params[1].controlCLSID = '{C8C46E43-3D27-4485-9B38-A49F3AC588D9}'
```



Number of Neighbors 750

Tool source code

Accessing parameter values

https://esriurl.com/gp_access_parameters

- In a script tool, access parameter values using one of:

- GetParameterAsText – values returned as a string
- GetParameter – values returned as a dynamic type
  - GetParameter is best for Boolean and numeric types

- For derived parameters, returns values back to the tool using:
  - SetParameterAsText or SetParameter

Tool messages

https://esriurl.com/gp_messages

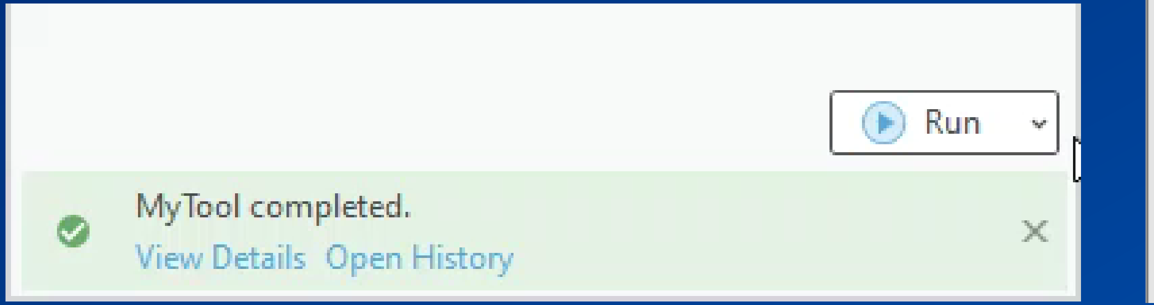
- Relay information using arcpy message functions
    - AddMessage
    - AddWarning
    - AddError
    - AddIDMessage – use Esri standard ID codes

- Note: Error messages are just messages, they will not end the script
    - Best to exit your code soon after, such as Python’s sys.exit()

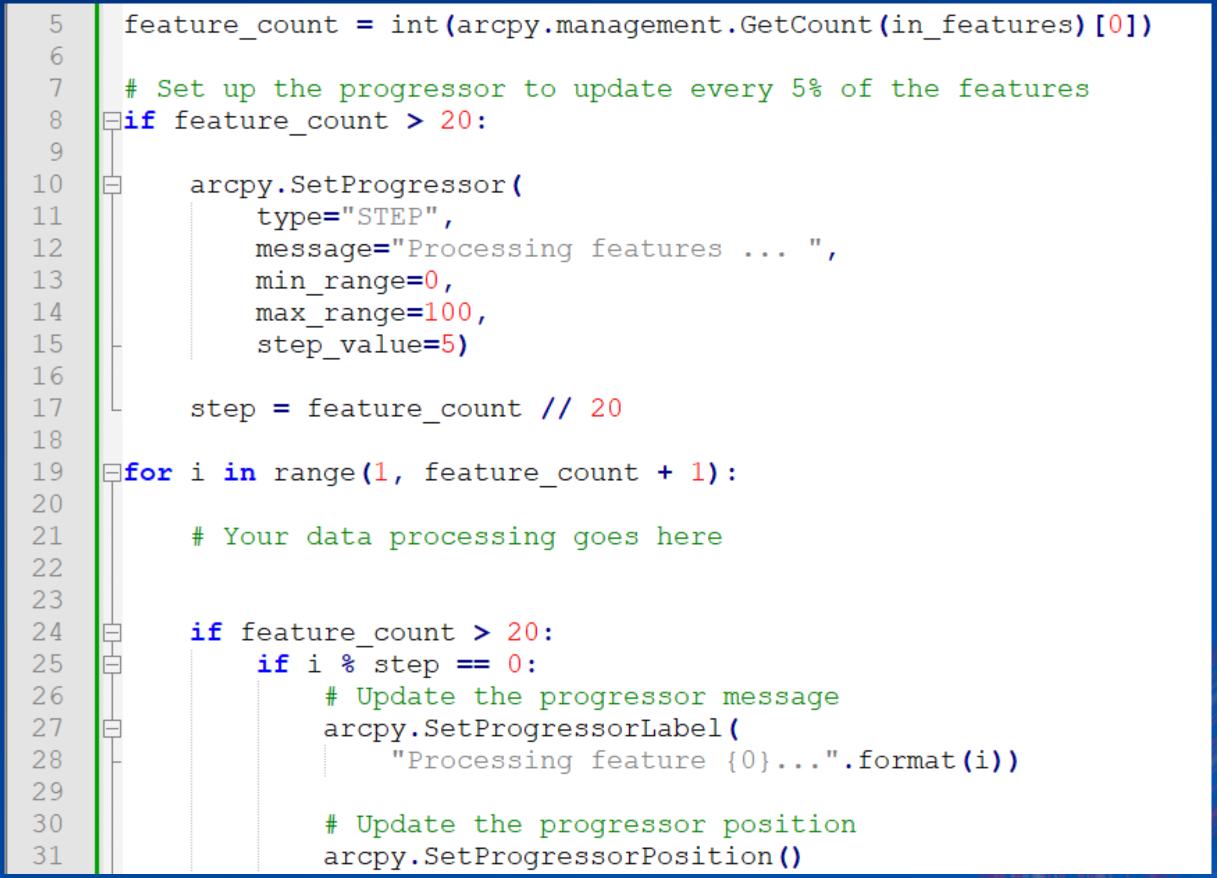
Tool progressor

https://esriurl.com/gp_progressor

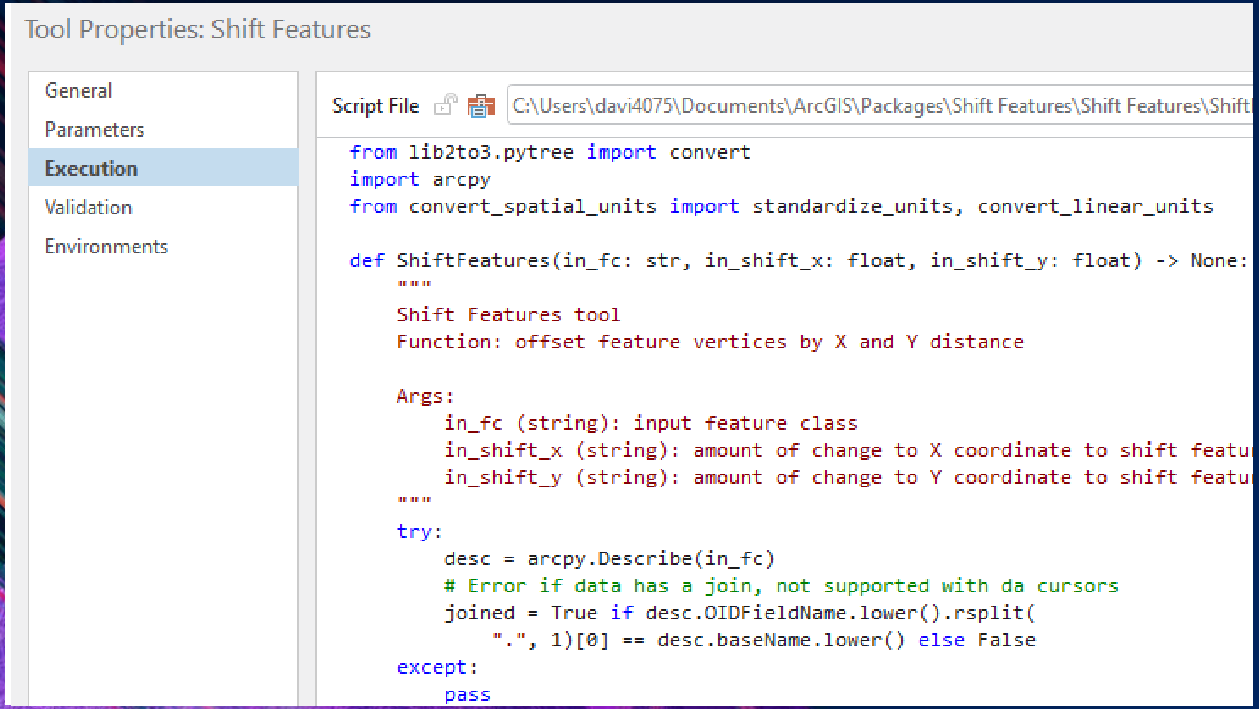
- Relay simple information to the Geoprocessing pane
- Provide messages and increments using:
  - SetProgressor
  - SetProgressorPosition
  - SetProgressorLabel
  - ResetProgressor



MyTool completed.



```
feature_count = int(arcpy.management.GetCount(in_features)[0])

# Set up the progressor to update every 5% of the features
if feature_count > 20:

    arcpy.SetProgressor(
        type="STEP",
        message="Processing features ... ",
        min_range=0,
        max_range=100,
        step_value=5)

    step = feature_count // 20

for i in range(1, feature_count + 1):

    # Your data processing goes here


    if feature_count > 20:
        if i % step == 0:
            # Update the progressor message
            arcpy.SetProgressorLabel(
                "Processing feature {0}...".format(i))

            # Update the progressor position
            arcpy.SetProgressorPosition()
```



Script File C:\Users\davi4075\Documents\ArcGIS\Packages\Shift Features\Shift Features\Shift

```
from lib2to3.pytree import convert
import arcpy
from convert_spatial_units import standardize_units, convert_linear_units

def ShiftFeatures(in_fc: str, in_shift_x: float, in_shift_y: float) -> None:
    """
    Shift Features tool
    Function: offset feature vertices by X and Y distance

    Args:
        in_fc (string): input feature class
        in_shift_x (string): amount of change to X coordinate to shift featu
        in_shift_y (string): amount of change to Y coordinate to shift featu
    """
    try:
        desc = arcpy.Describe(in_fc)
        # Error if data has a join, not supported with da cursors
        joined = True if desc.OIDFieldName.lower().rsplit(
            ".", 1)[0] == desc.baseName.lower() else False
    except:
        pass
```

Demo:
Source code

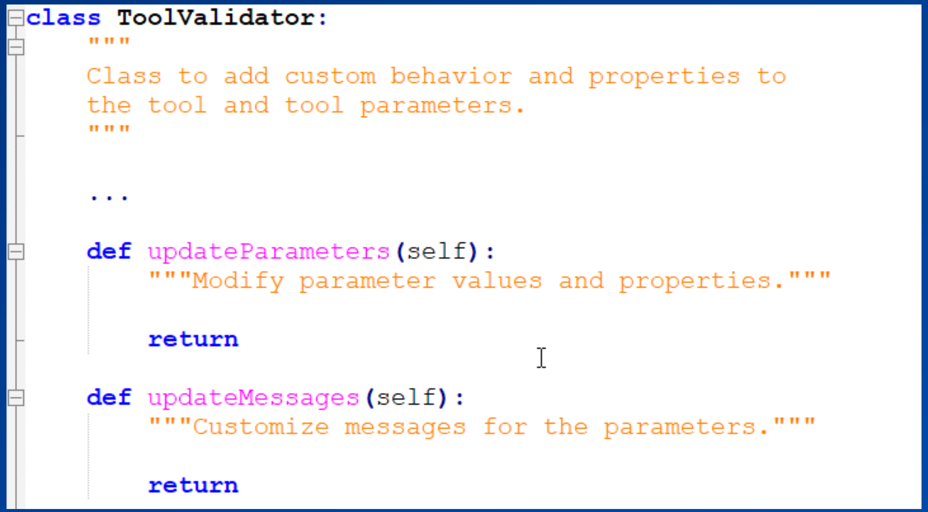
https://esriurl.com/shift_features_tool

Validation

Validation

https://esriurl.com/gp_validation

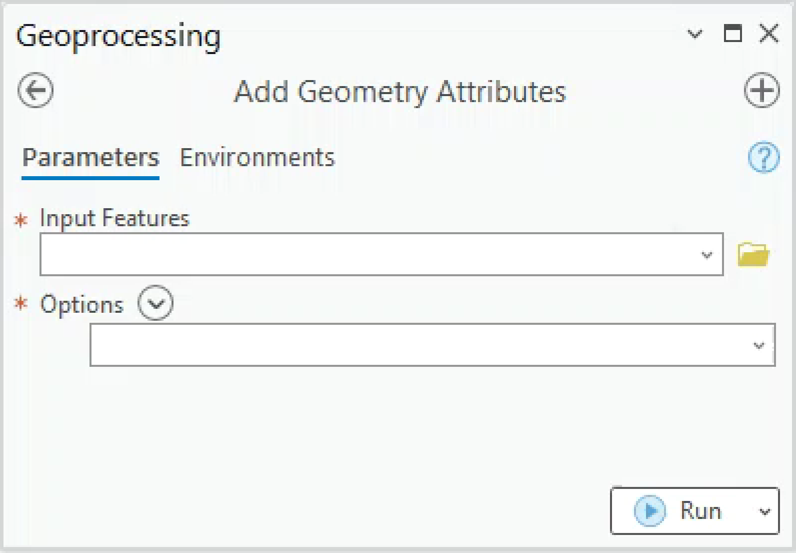
- Validation is interactive—runs every time a parameter is modified
- Parameters provide some simple validation by default
- Refine tool behavior with additional validation
  - Parameter interaction
  - Calculate defaults
  - Enable or disable parameters
  - Set parameter errors and messages
  - Define characteristics of your output (for ModelBuilder)

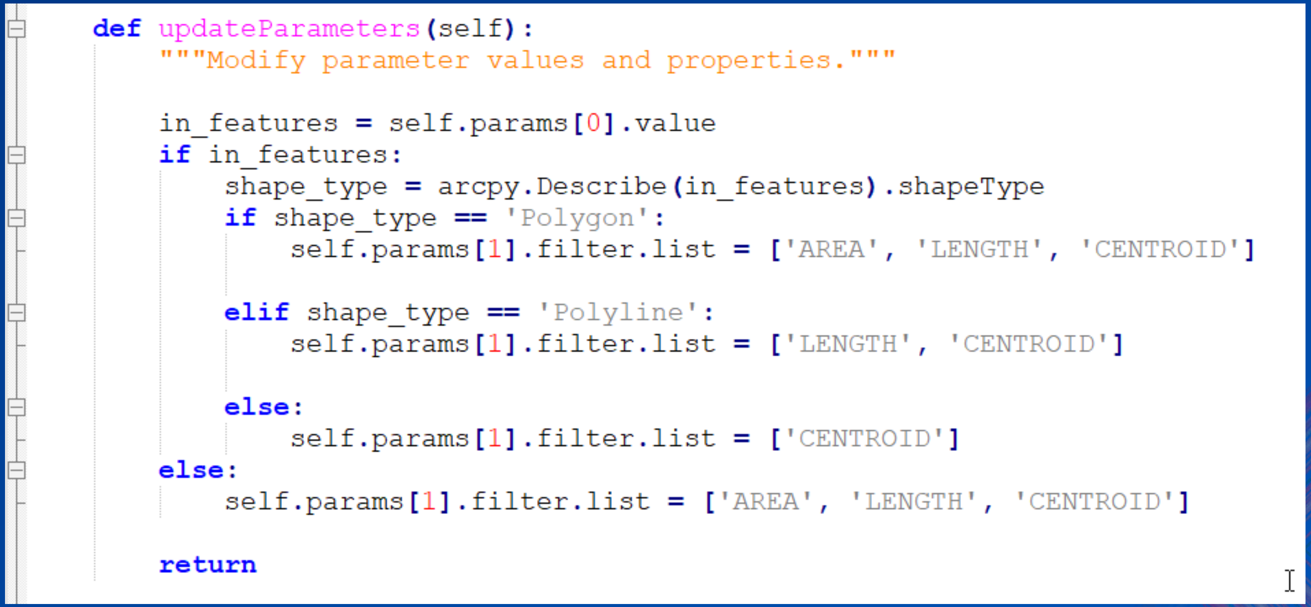


```
class ToolValidator:
    """
    Class to add custom behavior and properties to
    the tool and tool parameters.
    """


    ...

    def updateParameters(self):
        """Modify parameter values and properties."""

        return

    def updateMessages(self):
        """Customize messages for the parameters."""

        return
```

updateParameters

- updateParameters allows you to change certain parameter characteristics
  - Values
  - Filters
  - Enabled
  - Etc.

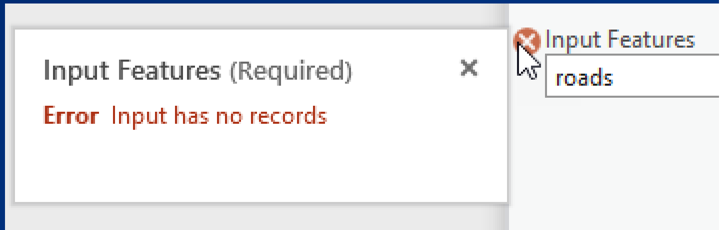




```
def updateParameters(self):
    """Modify parameter values and properties."""

    in_features = self.params[0].value
    if in_features:
        shape_type = arcpy.Describe(in_features).shapeType
        if shape_type == 'Polygon':
            self.params[1].filter.list = ['AREA', 'LENGTH', 'CENTROID']

        elif shape_type == 'Polyline':
            self.params[1].filter.list = ['LENGTH', 'CENTROID']

        else:
            self.params[1].filter.list = ['CENTROID']
    else:
        self.params[1].filter.list = ['AREA', 'LENGTH', 'CENTROID']

    return
```

updateMessages

- Provide warnings or errors before running the tool
- Provide information in Geoprocessing pane in real time

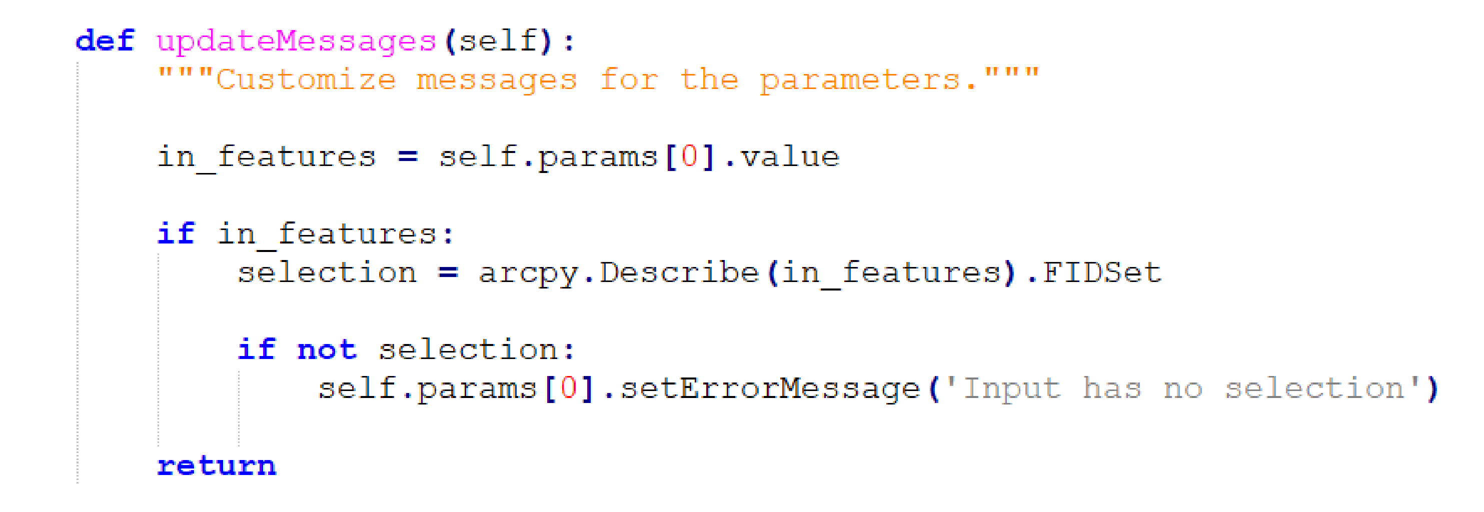


Input Features (Required)

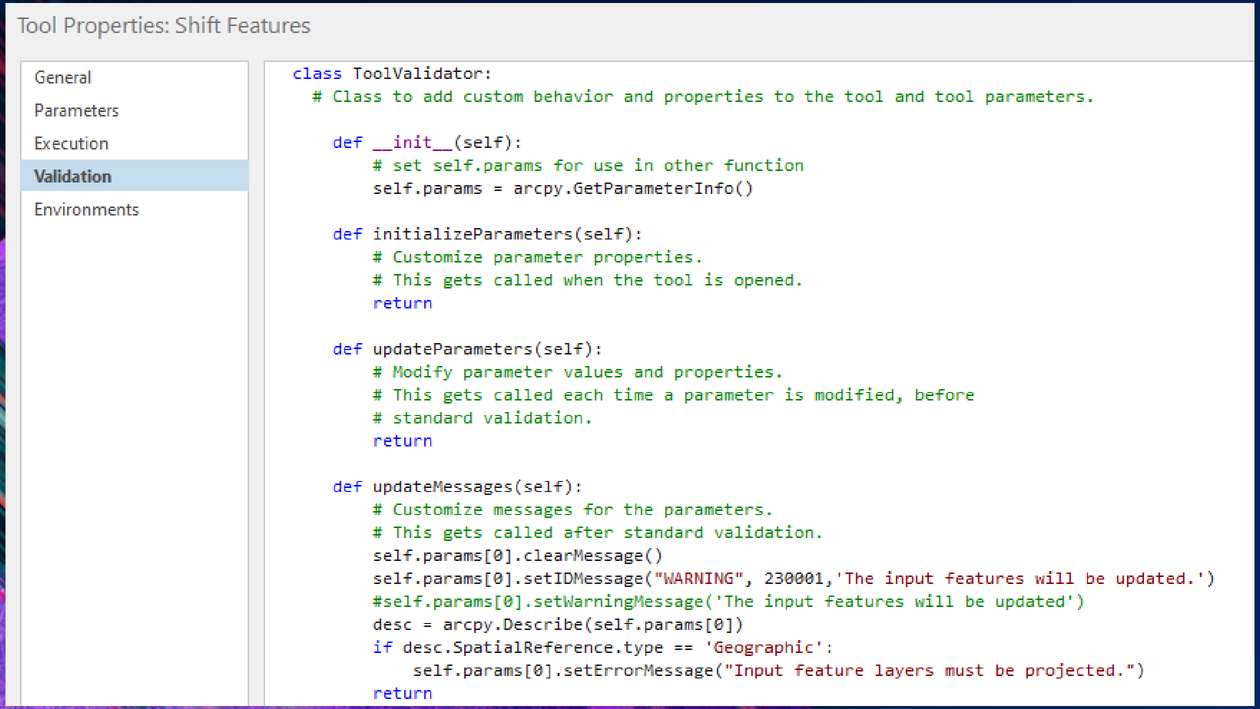
Error Input has no records

Input Features

roads



```
def updateMessages(self):
    """Customize messages for the parameters."""

    in_features = self.params[0].value

    if in_features:
        selection = arcpy.Describe(in_features).FIDSet

        if not selection:
            self.params[0].setErrorMessage('Input has no selection')

    return
```



```
class ToolValidator:
  # Class to add custom behavior and properties to the tool and tool parameters.

    def __init__(self):
        # set self.params for use in other function
        self.params = arcpy.GetParameterInfo()

    def initializeParameters(self):
        # Customize parameter properties.
        # This gets called when the tool is opened.
        return

    def updateParameters(self):
        # Modify parameter values and properties.
        # This gets called each time a parameter is modified, before
        # standard validation.
        return

    def updateMessages(self):
        # Customize messages for the parameters.
        # This gets called after standard validation.
        self.params[0].clearMessage()
        self.params[0].setIDMessage("WARNING", 230001,'The input features will be updated.')
        #self.params[0].setWarningMessage('The input features will be updated')
        desc = arcpy.Describe(self.params[0])
        if desc.SpatialReference.type == 'Geographic':
            self.params[0].setErrorMessage("Input feature layers must be projected.")
        return
```

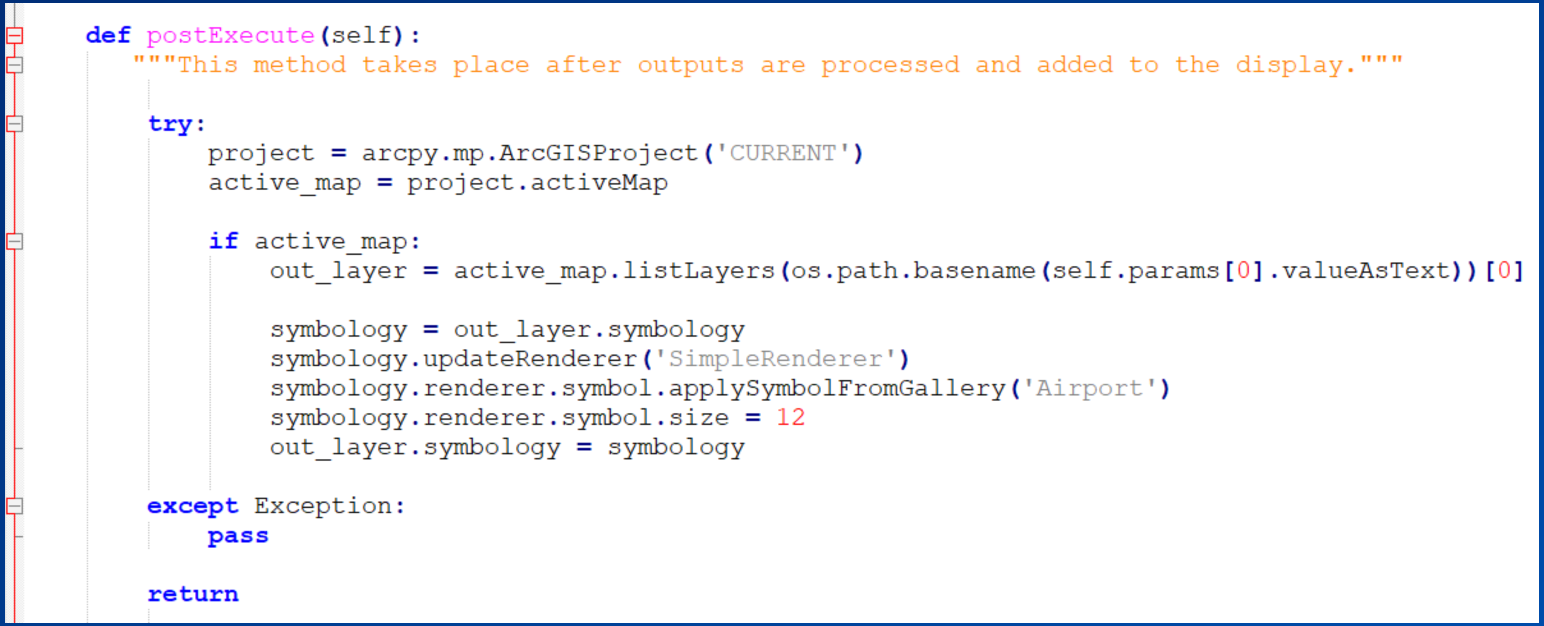
Demo: Validation

https://esriurl.com/shift_features_tool
https://esriurl.com/validation_game

postExecute

https://esriurl.com/gp_postprocessing

- Validation method that is called when a tool finishes
- Use for interacting with the map and modifying symbology



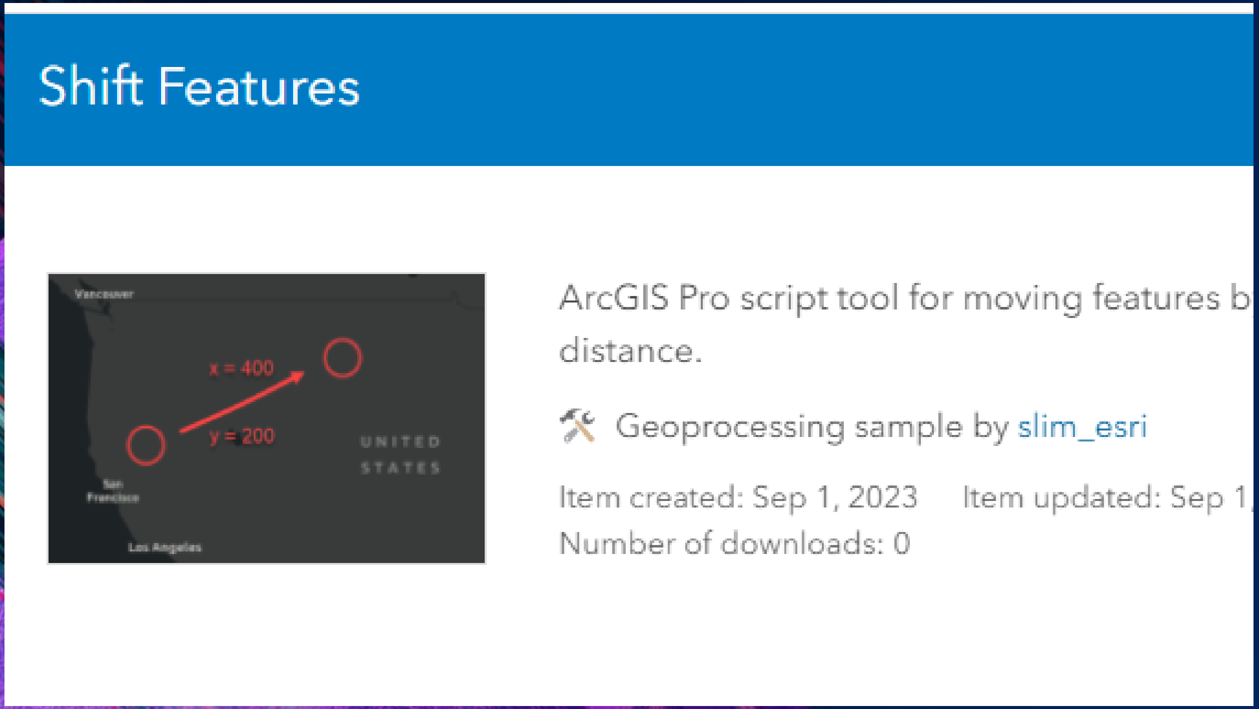
```python
def postExecute(self):
    """This method takes place after outputs are processed and added to the display."""

    try:
        project = arcpy.mp.ArcGISProject('CURRENT')
        active_map = project.activeMap

        if active_map:
            out_layer = active_map.listLayers(os.path.basename(self.params[0].valueAsText))[0]

            symbology = out_layer.symbology
            symbology.updateRenderer('SimpleRenderer')
            symbology.renderer.symbol.applySymbolFromGallery('Airport')
            symbology.renderer.symbol.size = 12
            out_layer.symbology = symbology

    except Exception:
        pass

    return
```

Sharing

Sharing

https://esriurl.com/gp_sharetools

- In many cases, you will be creating a tool for others
- There are several options for sharing tools:
  - Geoprocessing sample
  - Geoprocessing package
  - Web tool
  - Geoprocessing service
- In every Pro release, tools are added and enhanced
  - Use Analyze Toolbox For Version / Save Toolbox To Version tools to analyze your tools



Shift Features

ArcGIS Pro script tool for moving features b
distance.

🛠 Geoprocessing sample by slim_esri

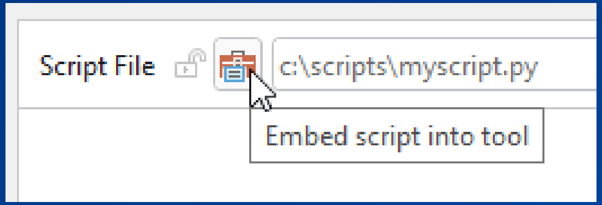
Item created: Sep 1, 2023   Item updated: Sep 1,
Number of downloads: 0

Demo: Sharing a geoprocessing package

https://esriurl.com/shift_features_tool

Embedding and encryption

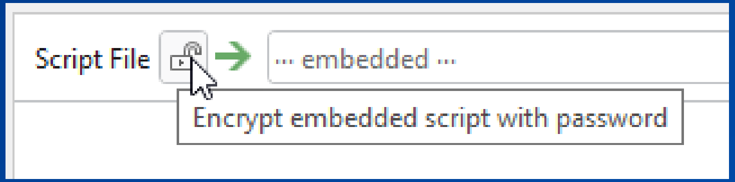
- You can embed code in .atbx and .tbx toolboxes
  - The source file is embedded by default
  - One less file to manage



Script File c:\scripts\myscript.py

Embed script into tool

- When embedded the file can be encrypted



Script File ··· embedded ···

Encrypt embedded script with password

- Python toolboxes also support encryption
  - The entire .pyt file is encrypted

Questions

https://www.linkedin.com/in/dave-wynne/
https://www.linkedin.com/in/seanlimhy/

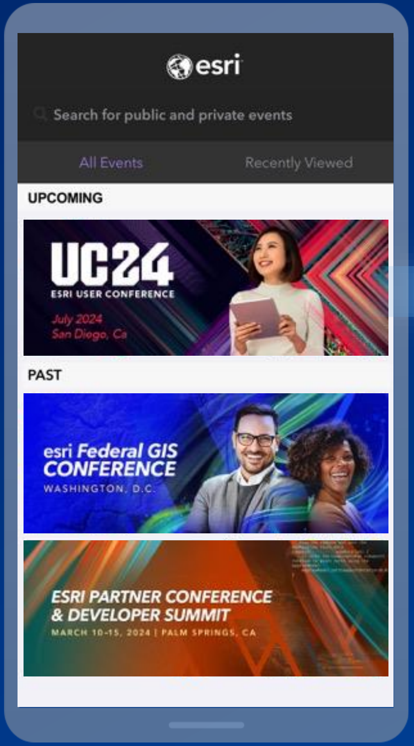
Building Geoprocessing Tools

Slides
and demo tools

https://esriurl.com/tool_workshop_2024

Please share your feedback in the app

Download the Esri Events app and find your event



esri

Search for public and private events

All Events | Recently Viewed

UPCOMING

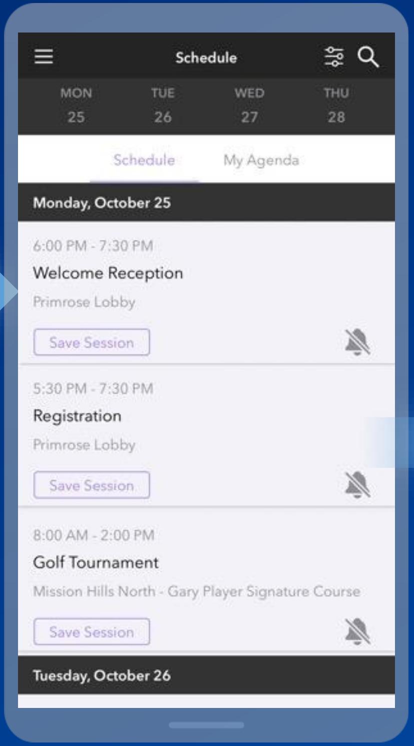
UC24 ESRI USER CONFERENCE July 2024 San Diego, Ca

PAST

esri Federal GIS CONFERENCE WASHINGTON, D.C.

ESRI PARTNER CONFERENCE & DEVELOPER SUMMIT MARCH 10-15, 2024 | PALM SPRINGS, CA

Select the session that you attended



Schedule

MON 25 | TUE 26 | WED 27 | THU 28

Schedule | My Agenda

Monday, October 25

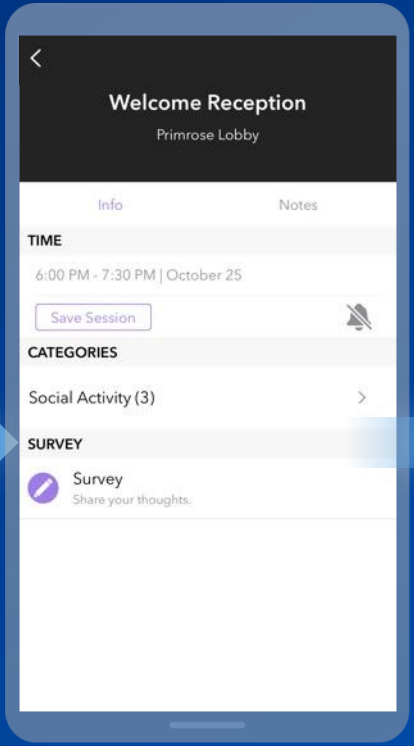
6:00 PM - 7:30 PM
Welcome Reception
Primrose Lobby
Save Session

5:30 PM - 7:30 PM
Registration
Primrose Lobby
Save Session

8:00 AM - 2:00 PM
Golf Tournament
Mission Hills North - Gary Player Signature Course
Save Session

Tuesday, October 26

Scroll down to “Survey”



Welcome Reception
Primrose Lobby

Info | Notes

TIME

6:00 PM - 7:30 PM | October 25

Save Session

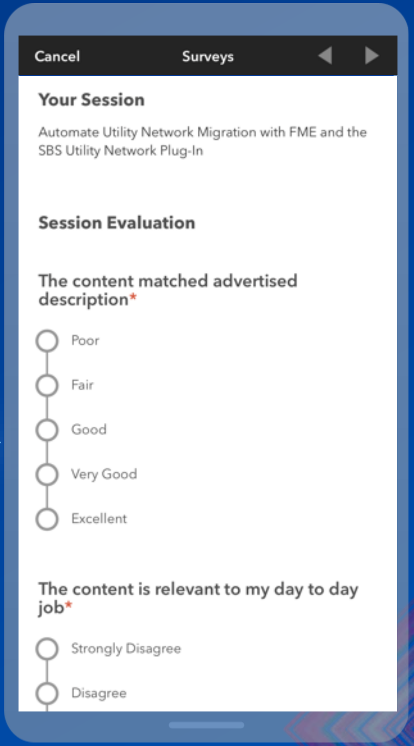
CATEGORIES

Social Activity (3)

SURVEY

Survey
Share your thoughts.

Log in to access the survey



Cancel | Surveys

Your Session

Automate Utility Network Migration with FME and the SBS Utility Network Plug-In

Session Evaluation

The content matched advertised description*

○ Poor
○ Fair
○ Good
○ Very Good
○ Excellent

The content is relevant to my day to day job*

○ Strongly Disagree
○ Disagree

esri®

THE
SCIENCE
OF
WHERE®

Copyright © 2024 Esri. All rights reserved.