Network Analyst: Automating Workflows with Geoprocessing

Patrick Stevens
Deelesh Mandloi
Introductions

• Who are we?
  - Network Analyst Product Engineers

• Who are you?
  - Current Network Analyst users?
  - Current geoprocessing users?
  - Have made geoprocessing models?
  - Experience with Python?
  - Have made geoprocessing python scripts?
Topics

- ArcGIS Network Analyst extension concepts
- Geoprocessing framework for network analysis
- Building geoprocessing models
- Writing Python scripts and building script tools
- Support and resources
- Network Analyst at the User’s Conference
- Questions
ArcGIS Network Analyst extension concepts

More Information:

What is the ArcGIS Network Analyst extension in ArcGIS help
ArcGIS
Network Analyst Extension
Solving transportation problems
Network Dataset

Transportation Network

Data Model

Network Dataset

Geodatabase

Shapefile StreetMap
Where do you get street data?

• Data within your organization
• No street data in your organization
  - Free data
    - Data and maps media
    - TIGER (Census data)
    - OpenStreetMap
      - OSM to NDS tools
      - ArcGIS Editor for OpenStreetMap
  - Pay for data
    - NAVTEQ or TomTom
      - Vendor street data processing tools
    - StreetMap Premium for ArcGIS
  - Pay for analysis
    - ArcGIS.com Map Viewer
    - ArcGIS.com Network Services
Network Analysis Layer

- Composite layer configured for a specific solver
- Stores analysis properties, inputs, and outputs from the solver
Geoprocessing Framework

More Information:

The geoprocessing framework in ArcGIS help
What is Geoprocessing?
What is Geoprocessing?

Compute = Geoprocessing

- Automate GIS Tasks
- Modeling & Analysis
Using Geoprocessing – How?

- Accessed through ArcToolbox
- Network Analyst Tools
  - Performing Network Analysis
  - Building networks
  - Publishing services
  - Managing turns
Using Geoprocessing – How?

- Accessed through ArcCatalog
  - Sign in with any organization with credits
- Global, Ready-To-Use Services
  - Closest Facility
  - Service Area
  - Vehicle Routing Problem
Using Geoprocessing – How?

Single tool execution

Chain tools

Model

Script

Tool dialog

Python window

Network Analyst: Automating Workflows with Geoprocessing
Using Geoprocessing – Where?

Web

Mobile

Desktop

Geoprocessing Tools

Model tools

Script tools

System (built-in) tools
Building Geoprocessing Models

More Information:

What is ModelBuilder? in ArcGIS help
Network Analysis Workflow

1. Make or Edit Network Analysis Layer

2. Add locations to one or more Network Analysis Classes

3. Solve

4. Use the results
Geoprocessing Models
Authoring a simple route model
Demo: Geoprocessing models - takeaways

• You can easily share models

• If running models as tools, make the output network analysis layer as **model parameter** so that it is added to the ArcMap Table of Contents

• Network analysis layer is the derived output from most of the tools (Add Locations, Solve)
Geoprocessing Models

- Chain geoprocessing tools to perform a workflow

- Authored using the Model Builder application

- Models behave like any other tools within ArcToolbox
  - Can use a model within another model

- All Model Builder techniques apply when authoring models for network analysis
Example Model to perform Service Area Analysis

- Numbers refer to steps in Network Analysis workflow
Adding analysis results to ArcMap

- If you want to visualize the results in ArcMap, when running models as tools, make the output network analysis layer a **model parameter**. This will add the layer to the ArcMap Table of Contents.
Post-processing your analysis

- If you want to use your analysis results as an input to another geoprocessing tool, use the **Select Data** tool to access individual sublayers
Geoprocessing Models

Authoring a model to determine multiple routes from a text file containing start and end addresses
Demo: Geoprocessing models - takeaways

- Use the **Select Data** tool to access sublayers of a network analysis layer
- Incorporate external data (csv in this example) into your analysis
- Automate your workflows without code
- Model tools can be added as buttons on any toolbar
- If network analysis layer is intermediate data, explicitly delete it as a last step
Writing Python Scripts

More Information:

What is Python? in ArcGIS help
Python Scripts

• Used for
  - Conditional logic
  - Looping
  - Cursors, creating geometry
  - Accessing built-in and third party python modules

• ArcPy site package
  - Network Analyst module
  - Access other geoprocessing tools
  - Other useful functions and classes such as Describe
What is the Network Analyst Module?

Simplify access to Network Analyst functionality from Python

arcpy.na
Network Analyst Module

- Support editing analysis properties of network analysis layers
  - No need to re-create layers
  - Speeds up execution
  - Simplifies script logic
  - Automate workflows from Python window

- Provide helper functions and classes to easily use Network Analyst GP tools from Python
Python Script - Basic Building Blocks

```python
# Import system modules
import arcpy
from arcpy import import env

# Set environment settings
env.workspace = "C:\data\workspace.gdb"
env.overwriteOutput = True

# Set local variables
inNetworkDataset = "Transportation\NodeMultiModel.shp"
outMLLayerName = "ClosestWarehouse"
impededAttributeName = "DriveTime"
accumulatedAttributeName = "("Distance")"
inFacilities = "Analysis\Warehouses"
incidents = "Analysis\Injuries"
outMLayerFile = "C:\data\output\*\" + outMLayerName

# Create a new closest facility analysis layer. Apart from finding the drive time to the closest warehouse, we also need to find the total distance. So we will accumulate the "DriveTime" impeded attribute.
outMLayer = arcpy.sa.MakeClosestFacilityLayer(inNetworkDataset, outMLayerName, "impededAttributeName", "accumulatedAttributeName", "by distance")
```
Python Script - Basic Building Blocks

```python
# Check out the Network Analyst extension license
arcpy.CheckOutExtension("Network")

# Set environment settings
env.workspace = "C:/data/Kenai.gdb"
env.overwriteOutput = True

# Set local variables
lnetworkDataset = "Transportation/FuelMultiModel_2013"
layerName = "ClosestWarehouse"
impedanceAttribute = "Distance"
accumulatedAttributeName = "(Network)"
impBegin = "Analysis/StartNodes"
impEnd = "Analysis/EndNodes"
outputFile = "C:/data/outputs/" + outLayerName + ".lyr"

# Create a new closest Facility analysis layer. Apart from finding the drive time
# from the service locations, we also need to find the total distance. So
# we will accumulate the "Distance" impedance attribute.
outLayer = arcpy.sa.NetworkClosestFacilityLayer(lnetworkDataset, outLayerName, 
    impedanceName, [impBegin, impEnd], accumulatedAttributeName, 
    "by network")
```
Python Script - Basic Building Blocks

Set inputs and outputs

```python
#Set environment settings
env.workspace = "C:/data/Paris.gdb"
env.overwriteOutput = True

#Set local variables
inNetworkDataset = "Transportation/ParisMultimodal_ND"
outNALayerName = "ClosestWarehouse"
impedanceAttributeName = "Drivetime"
accumulateAttributeName = ["Meters"]
inFacilities = "Analysis/Warehouses"
inIncidents = "Analysis/Stores"
outLayerFile = "C:/data/output" + "/" + outNALayerName + ".lyr"
```
Python Script - Basic Building Blocks

```python
# Description: Make a Closest Facility analysis to find the closest warehouse.
# From the store locations and save the results to a layer file in D:
# Requirements: Network Analyst Extension

# Import system modules
import arcpy
arcpy.CheckOutExtension("Network")

# Check out the Network Analyst extension license
resultObject = arcpy.na.MakeClosestFacilityLayer(inNetworkDataset, outNALayerName,
impedanceAttribute, "TRAVERSAL", 
"", 1, accumulateAttributeName, 
"NO_UTURNS")

# Get the layer object from the result object. The closest facility layer can
# now be referenced using the layer object.
outNALayer = resultObject.getOutput(0)
```

Make/edit a network analysis layer
Add locations to network analysis classes

```python
# Get the names of all the sublayers within the closest facility layer.
subLayerNames = arcpy.na.GetNAClassNames(outNALayer)

# Stores the layer names that we will use later
faciliticoLayerName = subLayerNames["Facilitics"]
incidentsLayerName = subLayerNames["Incidents"]

# Load the warehouses as Facilities using the default field mappings and search tolerance
arcpy.na.AddLocations(outNALayer, facilitiesLayerName, inFacilities, "", ",")

# Load the Stores as incidents. Map the Name property from the NOM field using field mappings
fieldMappings = arcpy.na.NAClassFieldMappings(outNALayer, incidentsLayerName)
fieldMappings["Name"].mappedFieldName = "NOM"
arcpy.na.AddLocations(outNALayer, incidentsLayerName, inIncidents, fieldMappings,"")
```
Python Script - Basic Building Blocks

```python
# Solve a closest facility analysis to find the closest warehouse
# from the store locations and save the results to a layer file
# for later use.
# Requirements: Network Analyst extension

# Import system modules
import arcpy
# From arcpy: Import env

# Check out the Network Analyst extension license
arcpy.CheckOutExtension("Network")

# Set environment settings
env.workspace = "\\data\facilities.gdb"
env.overwriteOutput = True

# Set local variables
inNetworkDataset = "Transportation/Facilities\facilityNetDataset"
outNALayerName = "ClosestWarehouse"
legendAttribute = "DriveTime"
accumulateAttribute = "Distance"

# Solve the closest facility layer
arcpy.na.Solve(outNALayerName)
```

Solve the network analysis layer
Use the results

```
# Save the solved closest facility layer as a layer file on disk with
# relative paths
arcpy.management.SaveToLayerFile(outNALayer, outLayerFile, "RELATIVE")
print "Script completed successfully"
```
Working with analysis layers within scripts

- The network layer can be accessed as a layer object via the result object of a Make<solver>Layer function.
Working with analysis layers within scripts

- The network analysis layer can be edited via the solver properties of an existing layer object

```python
# Get the service area layer as an input parameter
salayer = arcpy.GetParameter(0)

# Get the solver properties object from the service area layer
solverProps = arcpy.da.GetSolverProperties(salayer)

# Update the properties for the service area layer using the solver properties
solverProps.defaultBreaks = [5, 10, 15]
solverProps.useHierarchy = "USE_HIERARCHY"
```
Accessing sublayers in scripts

• To access sublayers in python scripts, use the arcpy.na.GetNAClassNames function
  - The Select Data tool is not meant for python scripting
  - Write scripts that work across ArcGIS language versions
  - Avoid using localized strings in scripts such as sublayer names

```python
#Get the names of all the sublayers within the closest facility layer.
subLayerNames = arcpy.na.GetNAClassNames(outNALayer)

#Store the layer names that we will use later
facilitiesLayerName = subLayerNames["Facilities"]

#Load the warehouses as Facilities using the default field mappings and search tolerance
arcpy.na.AddLocations(outNALayer, facilitiesLayerName, inFacilities, "", ")
```
Working with analysis layers within scripts

- Helper classes for complex parameter types
  - Easily specify field mappings in Add Locations tool by using `arcpy.na.NAClassFieldMappings`
Saving analysis results

• The in-memory network analysis layer can be persisted using **SaveToLayerFile** geoprocessing tool in the management module

```python
arcpy.management.SaveToLayerFile(outNALayer, outLayerFile, "RELATIVE")
```

• Layer files can then be dragged from disk into ArcMap manually
Demo: Python Script

Authoring a Python script that finds the best sequenced route for given stops
Demo: Python Script - takeaways

- The network analysis layer can be referenced within the script using its name.
- The in-memory network analysis layer can be persisted using `SaveToLayerFile` geoprocessing tool.
- The sublayers within a network analysis layer are feature layers that can be used with many other tools.
- Scripts can be run at the operating system command prompt.
Building Script Tools

More Information:

[What is a script tool?] in ArcGIS help
Script Tools

- Script tools allow you to work with your scripts through a user interface, instead of a command line

- Script tools behave like any other tool within ArcToolbox
  - Can use script tools in models and vice versa
Add outputs from script tool to ArcMap

• If network analysis layer is the output, make an additional derived output parameter of type Network Analyst Layer and use `arcpy.SetParameterAsText(…)`

```python
# Do your analysis workflow
outNALayer = arcpy.na.MakeClosestFacilityLayer(inNetworkDataset, outNALayerName)
```

```python
# Set your analysis layer as an output parameter for the script tool
arcpy.SetParameterAsText(1, outNALayerName)
```
Script Tool

1. Creating a script tool to provide a UI for a Python script
2. Solve an allocation problem assigning students to schools with capacity constraints
Determine Optimum Allocation Script Tool

- Scripts can take advantage of all the capabilities provided by the python language
- Call third party applications that support python interface to have a “tightly coupled” approach

- For example, calling linear programming (LP) solvers using PuLP
  - PuLP is a public domain Python module for modeling LP problems
  - PuLP can work with a variety of LP solvers such as COIN-OR, GLPK, XPRESS, CPLEX.
Demo: Script Tool - takeaways

• If network analysis layer is the output, make an additional derived output parameter of type Network Analyst Layer and use `arcpy.SetParameterAsText()`.

• Custom validation logic can be programmed for the script tool user interface by programming the Tool Validator class.

• The output network analysis layer supports pre-defined symbology using layer files.
What’s new in 10.2

• Global, Ready-To-Use services
Summary
Summary

• Geoprocessing framework for network analyses
  - Network Analyst Tools (system tools)
  - Models and Model tools (no programming)
  - Script and Script tools (python code)

• Automate repetitive tasks

• Easier than writing ArcObjects code

• Incorporate network analysis in larger process
Resources
Support and Resources

• Tutorials
  - [Network Analyst tutorial](#)
  - [Network Analyst geoprocessing service examples](#)

• Code samples in Network Analyst tools toolbox

• [ArcGIS Network Analyst Extension Discussion Forum](#)

• [ArcGIS for Transportation Analytics Group on arcgis.com](#)

• [Python for ArcGIS resource center](#)
Network Analyst at the User’s Conference
<table>
<thead>
<tr>
<th>Time</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30 am</td>
<td>Network Analyst: An Introduction</td>
<td>Network Analyst: Network Analysis with ArcGIS Online and On-premise Services</td>
<td>Designing your Network Analyst Workflow</td>
</tr>
<tr>
<td>9 am</td>
<td>Room 32 B</td>
<td>Room 32 A</td>
<td>Hall F: Room 1</td>
</tr>
<tr>
<td>10 am</td>
<td>Network Analyst: Performing Network Analysis</td>
<td>Network Analyst: Automating Workflows with Geoprocessing</td>
<td></td>
</tr>
<tr>
<td>11 am</td>
<td>Room 32 B</td>
<td>Room 32 A</td>
<td></td>
</tr>
<tr>
<td>12 pm</td>
<td></td>
<td>Network Analyst: Routing and Directions using Data and Services on ArcGIS Online</td>
<td></td>
</tr>
<tr>
<td>2 pm</td>
<td>Room 32 B</td>
<td>Room 32 B</td>
<td></td>
</tr>
<tr>
<td>3 pm</td>
<td>Network Analyst: Creating Network Datasets</td>
<td>Real-time Traffic and Other New Capabilities of Network Analysis</td>
<td></td>
</tr>
<tr>
<td>4 pm</td>
<td>Room 32 B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 pm</td>
<td></td>
<td>Designing your Network Analyst Workflow</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Tuesday</td>
<td>Wednesday</td>
<td>Thursday</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>8:30 am</td>
<td><strong>Transportation Network Analysis and Planning</strong></td>
<td><strong>Multi-Modal Transportation and Logistics: Leading Examples</strong></td>
<td><strong>Esri &amp; OpenStreetMap: Tools, Apps, Maps!</strong></td>
</tr>
<tr>
<td>9 am</td>
<td>Room 26 B</td>
<td>Room 28 E</td>
<td>Hall G: Room 2</td>
</tr>
<tr>
<td>10 am</td>
<td>Electric Vehicles: GIS for EV Infrastructure</td>
<td>Transportation Planning for Rural Areas</td>
<td></td>
</tr>
<tr>
<td>11 am</td>
<td>Room 29 A/B</td>
<td>Room 26 B</td>
<td>Using Streetmap Premium</td>
</tr>
<tr>
<td>12 pm</td>
<td></td>
<td></td>
<td>Online GIS Exhibit Hall C</td>
</tr>
<tr>
<td>1 pm</td>
<td>Customer and Route Optimization in Public Works</td>
<td>Indoor Location, Tracking, and Routing</td>
<td></td>
</tr>
<tr>
<td>2 pm</td>
<td>Room 30 B</td>
<td>Room 31 B</td>
<td></td>
</tr>
<tr>
<td>3 pm</td>
<td>GIS for Non-Motorized Transport</td>
<td>Public Transit: Accessibility and Land Use</td>
<td></td>
</tr>
<tr>
<td>4 pm</td>
<td>Room 26 B</td>
<td>Facilities and Real Property Management</td>
<td></td>
</tr>
<tr>
<td>5 pm</td>
<td>Using Streetmap Premium</td>
<td>Pedestrian Routing, Transit, and Tolls</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Online GIS Exhibit Hall C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Thank you…

Please fill out the session evaluation

First Offering ID: 1364
Second Offering ID: 1449

Online – www.esri.com/ucsessionsurveys
Paper – pick up and put in drop box