ArcGIS Network Analyst: Automating Workflows with Geoprocessing

Melinda Morang
Patrick Stevens
Introductions

Who are we?
- Network Analyst Product Engineers

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

- ArcGIS Network Analyst extension concepts
- Geoprocessing framework for network analysis
- Models and model tools
- Python scripts and script tools
- Ready-to-Use services
- Support and resources
ArcGIS Network Analyst extension concepts

ArcGIS Help:
What is the ArcGIS Network Analyst extension
ArcGIS Network Analyst Extension does transportation analysis

Coverage
- Service Area

Optimization
- Location-Allocation
- Vehicle Routing Problem

Point-to-point routing
- Route
- Closest Facility
- Origin-Destination Cost Matrix
Where do you get street data?

**Data within your organization**

**Free data**
- Data and maps media
- TIGER (Census data)
- OpenStreetMap
  - OSM to NDS tools
  - ArcGIS Editor for OpenStreetMap

**Pay for data**
- HERE 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
- One layer type for each solver
- Analysis properties
- Inputs
- Outputs
Geoprocessing Framework
What is Geoprocessing?

Computation

Visualization

Storage & Management
What is Geoprocessing?

Computation = Geoprocessing

- Automating workflows
- Modeling & Analysis
Using Geoprocessing – How?

- Performing Network Analysis
- Building networks
- Publishing services
- Managing turns
Using Geoprocessing – How?

- Single tool
- Tool dialog
- Python window
- Chain tools
- Model
- Script
Using Geoprocessing – Where?

- Model tools
- Script tools
- System tools

Web
Mobile
Desktop
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
Network Analyst workflow
Performing an analysis manually
Takeaways
Demo: Network Analyst workflow

• Three steps for network analysis:
  - Make Layer
  - Add Locations
  - Solve
Building Geoprocessing Models

ArcGIS Help: What is ModelBuilder?
Geoprocessing Models

• Author using Model Builder

• Chain tools to perform a workflow

• Use models like ArcToolbox tools

• Use models within other models

• Apply all Model Builder techniques to network analysis models
Example Service Area analysis model
Visualizing analysis results in ArcMap

- When running models as tools…
  - The output network analysis layer should be a **model parameter**
  - This will add the layer to the ArcMap Table of Contents
Post-processing your analysis

- Use your analysis result as an input to another tool
  - The **Select Data** tool accesses individual sublayers
Automating workflows with geoprocessing models

Performing a network analysis in Model Builder

Sharing a model as a tool

Working with inputs and outputs
Takeaways
Demo: Automating workflows with geoprocessing models

• **Automate workflows** with Model Builder
  - Make inputs and outputs **model parameters**
    - Inputs can be selected by the tool’s user
    - Outputs will be added to the ArcMap Table of Contents
  - Use the **Select Data** tool to access NA sublayers
  - Work with **external data** like CSV files

• **Share** your model as a tool
Writing Python Scripts

ArcGIS Help:
What is Python?
Python Scripts

• Conditional logic

• Loops

• Cursors, creating geometry

• Built-in and third party modules

• Cross-platform
Python Scripts

- ArcPy site package
  - Network Analyst module
  - Other geoprocessing tools
  - Other useful functions and classes
    - Describe
Network Analyst Module

- Simplify access to Network Analyst functionality from Python

- Edit the analysis properties of network analysis layers
  - No need to re-create layers
  - Speed up execution
  - Simplify script logic
Python Script - Basic Building Blocks

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

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

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

#Create local variables
netDataset = "Transportation\NetworkModel_008"
outMLayerName = "ClosestWarehouse"
impedanceAttribute = "Time"
accumulatedAttribute = "Distance"
impDistAttribute = "Distance"
impName = "Analysis\Warehouse"
writeMLayerFile = "C:/Network/output/FastestPath\" + outMLayerName + ".shp"

#Create a new closest facility analysis layer. Apart from finding the drive times to the closest warehouse, we also want to find
#the total distance. So we will accumulate the "Time" impedance attribute.
outMLayer = arcpy.sa.MakeClosestFacilityLayer(netDataset, outMLayerName, impedanceAttribute, accumulatedAttribute, outputsName = impDistName, writeMLayer = writeMLayer)
```
Python Script - Basic Building Blocks

Check out the Network Analyst Extension

```python
#Check out the Network Analyst extension license
arcpy.CheckOutExtension("Network")
```
Python Script - Basic Building Blocks

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

# Set local variables
inNetworkDataset = r"Transportation/ParisMultimodal_ND"
outNALayerName = "Closest\nWarehouse"
impedanceAttribute = "Drivetime"
accumulateAttributeName = ["Meters"]
inFacilities = r"Analysis/Warehouses"
inIncidents = r"Analysis/Stores"
outLayerFile = os.path.join(r"C:/data/output", outNALayerName + ".lyr")
```

Set inputs and outputs
Python Script - Basic Building Blocks

Make/edit a network analysis layer

```python
resultObject = arcpy.na.MakeClosestFacilityLayer(inNetworkDataset, outNALayerName, impedanceAttribute, "TRAVEL_TO", ",", 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)
```

Esri UC 2014 | Technical Workshop | ArcGIS Network Analyst: Automating Workflows with Geoprocessing
Add locations to network analysis classes
Python Script - Basic Building Blocks

Solve the network analysis layer

```python
# Solve the closest facility layer
arcpy.na.Solve(outNALayer)
```
Python Script - Basic Building Blocks

```python
# 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"
```

Use the results
Working with analysis layers within scripts

- The network layer is retrieved as a layer object from the result object returned by `arcpy.na.MakeClosestFacilityLayer`.
Working with analysis layers within scripts

- Edit 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_na.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

• The Select Data tool is not meant for python scripting

• arcpy.na.GetNAClassNames should be used
  - Renamed or localized sublayer names will work in the script

```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

- Easily specify field mappings in Add Locations with `arcpy.na.NAClassFieldMappings`

```python
barrierFieldMappings = "Name # Precipitation; BarrierType # 1; " + "Attr_#s #s #" & (impedance, scaleFactorField)
arcpy.na.AddLocations(routeLayer, polygonBarriersNAClass, weatherPolygonLayer, barrierFieldMappings)
```

```python
naClasses = arcpy.na.GetNAClassNames(routeLayer)
polygonBarriersNAClass = naClasses['PolygonBarriers']
barrierFieldMappings = arcpy.na.NAClassFieldMappings(routeLayer, polygonBarriersNAClass, False, arcpy.ListFields(weatherPolygonLayer))
barrierFieldMappings['Name'].defaultValue = "Precipitation"
barrierFieldMappings['BarrierType'].defaultValue = 1
barrierFieldMappings['Attr_' + defaultImpedance].mappedFieldName = scaleFactorField

arcpy.na.AddLocations(routeLayer, polygonBarriersNAClass, weatherPolygonLayer, barrierFieldMappings)
```
Saving analysis results

- The in-memory network analysis layer can be persisted using the `SaveToLayerFile` tool in the management module.

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

- Layer files can be dragged from disk into ArcMap.
Working with sublayers in python

• Access individual sublayers using `ListLayers` on the NA layer.

```python
# Get the output Routes sublayer and save it to a feature class
routesSubLayer = arcpy.mapping.ListLayers(outNALayer, sublayerNames["Routes"])[0]

arcpy.management.CopyFeatures(routesSubLayer, outRoutesFC)
```

• Use sublayers as input to other tools (CopyFeatures, Join, Buffer, etc.).

• Use a SearchCursor to access the rows within a sublayer.
Automating workflows with python scripts

Performing a network analysis with a python script
Takeaways
Demo: Automating workflows with python scripts

- Run simple python commands in ArcMap’s python window
- Export models to python scripts
- Persist the in-memory network analysis layer with SaveToLayerFile
- Run scripts from a command prompt outside of ArcMap
ArcGIS Help:
What is a script tool?

Building Script Tools
Script Tools

- Work with your scripts through a user interface

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

• If a network analysis layer is the output...
  - Make an additional derived output parameter of type Network Analyst Layer
  - Use `arcpy.SetParameterAsText(…)`

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

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

Create a script tool to provide a UI for a Python script.

Use tool validation to customize the UI.

Write a script tool to extend the capabilities of ArcGIS.
Takeaways
Demo: Creating a script tool

• Provide a user interface for python scripts by making a **script tool**

• Use derived output and `arcpy.SetParameterAsText()` to add results to the map

• Use **tool validation** to customize your script tool’s UI

• Use python modules to **extend** the capabilities of ArcGIS
Ready-To-Use Services

- Published by Esri
- Run in an Esri-administrated cloud infrastructure
Benefits of Ready-To-Use Services

- **Global data**
- **Live traffic**
- **No Network Analyst specific software required**
- **No maintenance of servers, services, or data**
- **Excellent uptime and reliability**
- **Create Layer, Add Locations, and Solve all in one tool**
Where can you use Ready-To-Use Services

• Anywhere geoprocessing tools can be used
  - Model builder
  - Python scripts
  - ArcMap

• Like any other service that uses credits

• Supported from 10.0 SP5 and up
Ready-To-Use Services Requirements

- ArcGIS Online subscription with credits
- Your own analysis inputs
How much do Ready-To-Use Services cost?

- Check [credit page](#)
- Varies per type of analysis

<table>
<thead>
<tr>
<th>Network Analysis</th>
<th>Service Credits Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Routes</td>
<td>0.04 credits per route</td>
</tr>
<tr>
<td>Optimized Routes</td>
<td>0.5 credits per optimized route</td>
</tr>
<tr>
<td>Drive Time (Service Areas)</td>
<td>0.5 credits per drive time</td>
</tr>
<tr>
<td>Closest Facilities</td>
<td>0.5 credits per closest facility route</td>
</tr>
<tr>
<td>Multi-Vehicle Routes( VRP)</td>
<td>1 credits per route</td>
</tr>
<tr>
<td>Traffic</td>
<td>0 credits</td>
</tr>
</tbody>
</table>
Ready-To-Use services

Connecting to Ready-To-Use services

Using a Ready-To-Use service in a model
Takeaways

Demo: Ready-To-Use services

• Use **Ready-To-Use services** like other geoprocessing tools

• You **don’t need** your own street data
Publishing geoprocessing services

- More details about authoring, publishing and using on-premise network analysis services are available in technical workshop titled **Performing Network Analysis with ArcGIS for Server** from 2012 user conference
  - Workshop presentation
  - Workshop video

- Applicable for ArcGIS for Server 10.1 and 10.2
Summary
Summary

• Use the geoprocessing framework for network analyses
  - Network Analyst Tools
  - Models and Model tools
  - Script and Script tools

• Automate workflows

• 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
Support and Resources

• Python for ArcGIS resource center

• Books
  - GIS Tutorial for Python Scripting
  - Python Scripting for ArcGIS