Esri International User Conference
San Diego, California

Technical Workshops | July 25, 2012

Network Analyst: Automating Workflows with Geoprocessing

Deelesh Mandloi

Patrick Stevens

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?

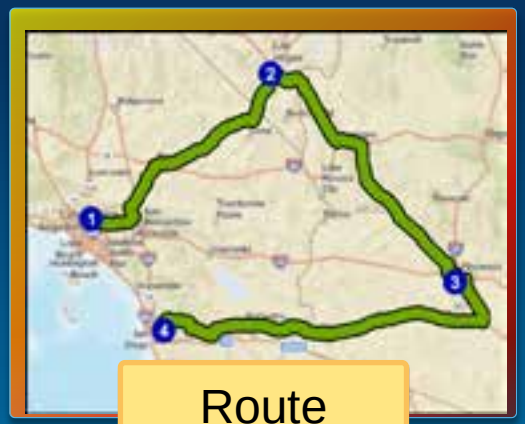
Topics

- 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

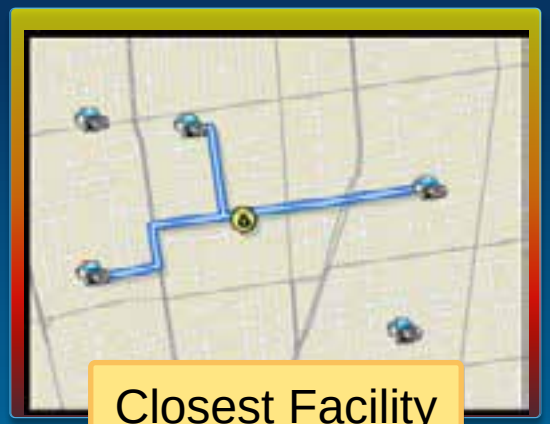
ArcGIS Network Analyst extension concepts

More Information:

What is the ArcGIS Network Analyst extension in ArcGIS help



Route

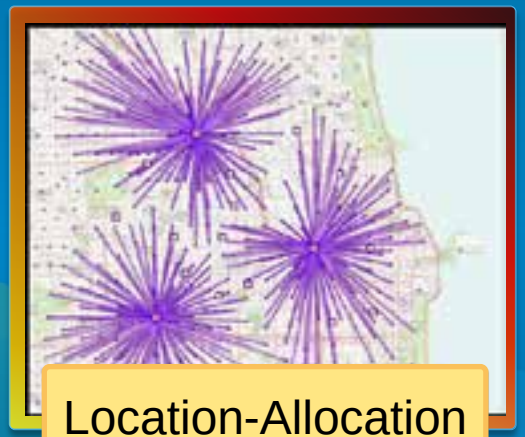


Closest Facility

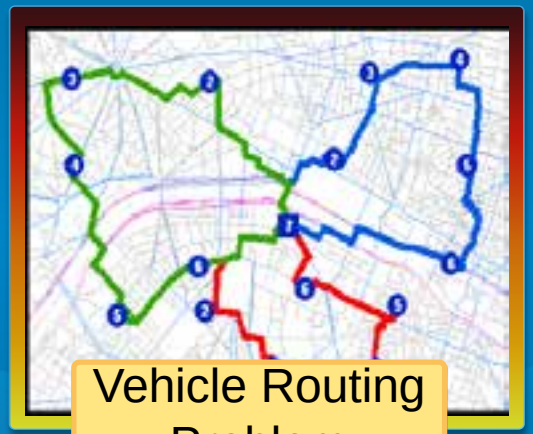
Service Area

ArcGIS Network Analyst Extension

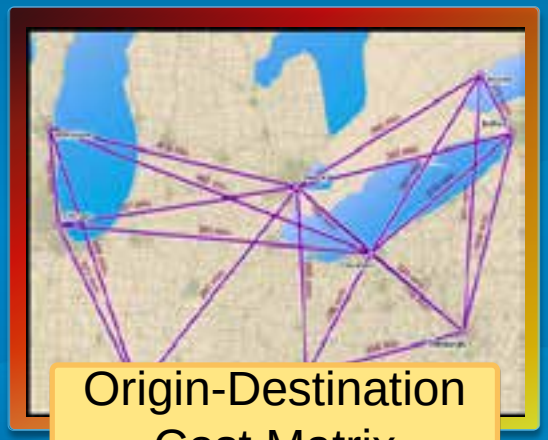
Solving transportation problems



Location-Allocation



Vehicle Routing Problem



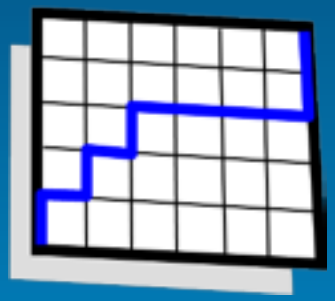
Origin-Destination Cost Matrix

Network Dataset

Transportation Network

Data Model

Network Dataset



Geodatabase

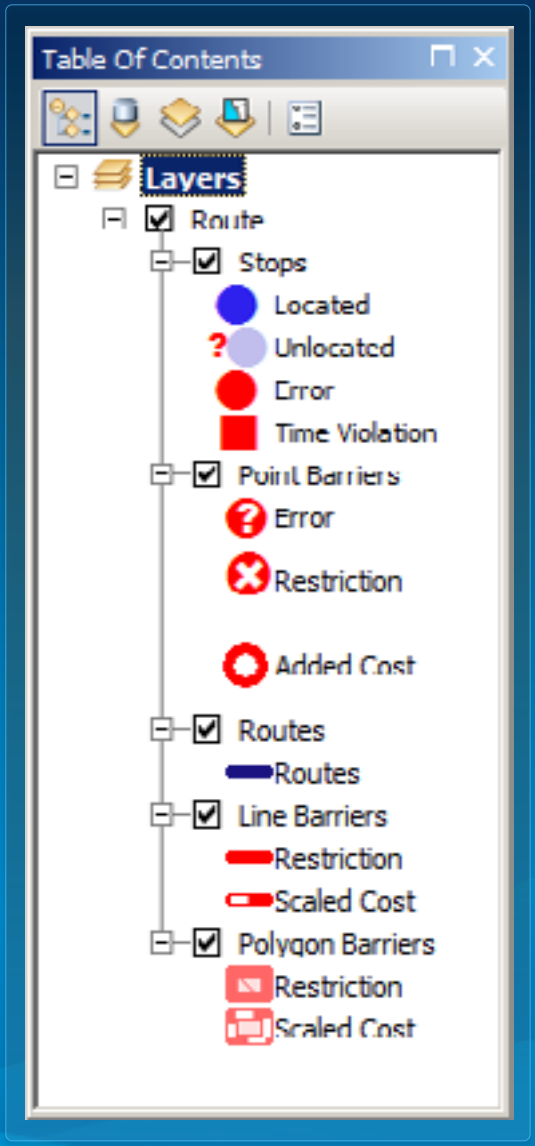
Shapefile
StreetMap

Where do you get street data?

- StreetMap Premium for ArcGIS
- Vendor data
  - NAVTEQ or TomTom
  - Vendor street data processing tools
- Data and Maps media
  - Included with ArcGIS
  - North America ready-to-route dataset
- Your own data
- Government data
  - TIGER
- Community data
  - OpenStreetMap
  - OSM to NDS tools
- No street data at all
  - Network analysis services on ArcGIS online

Network Analysis Layer

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

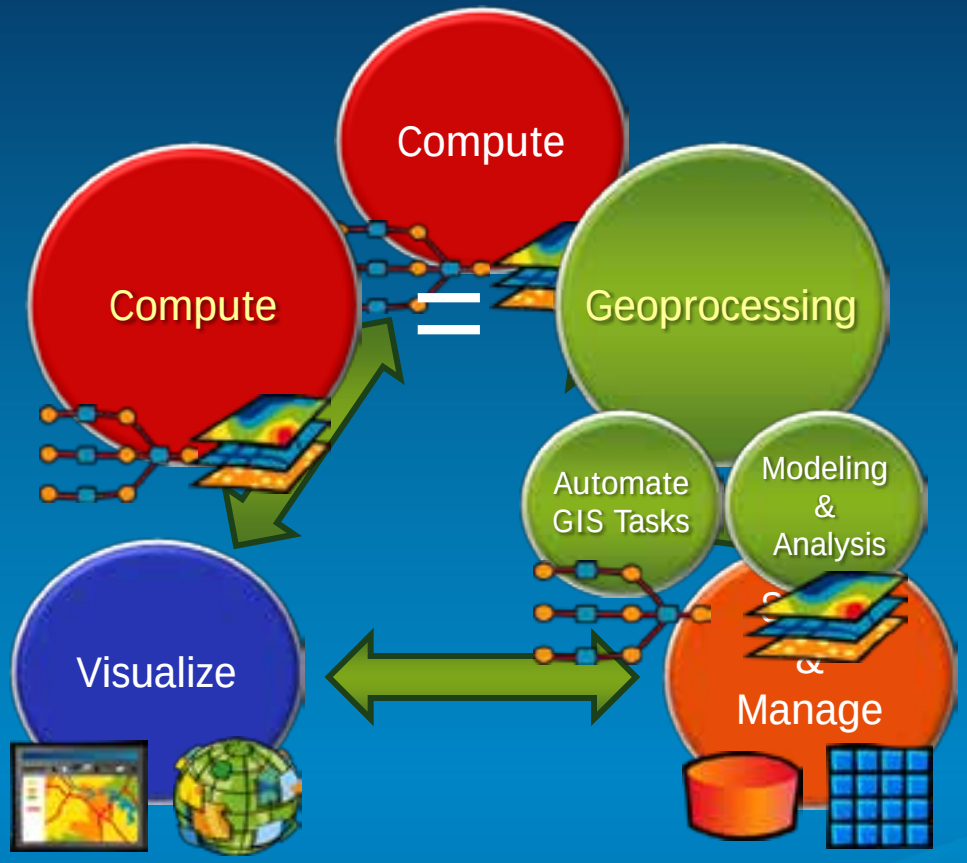


Geoprocessing Framework

More Information:

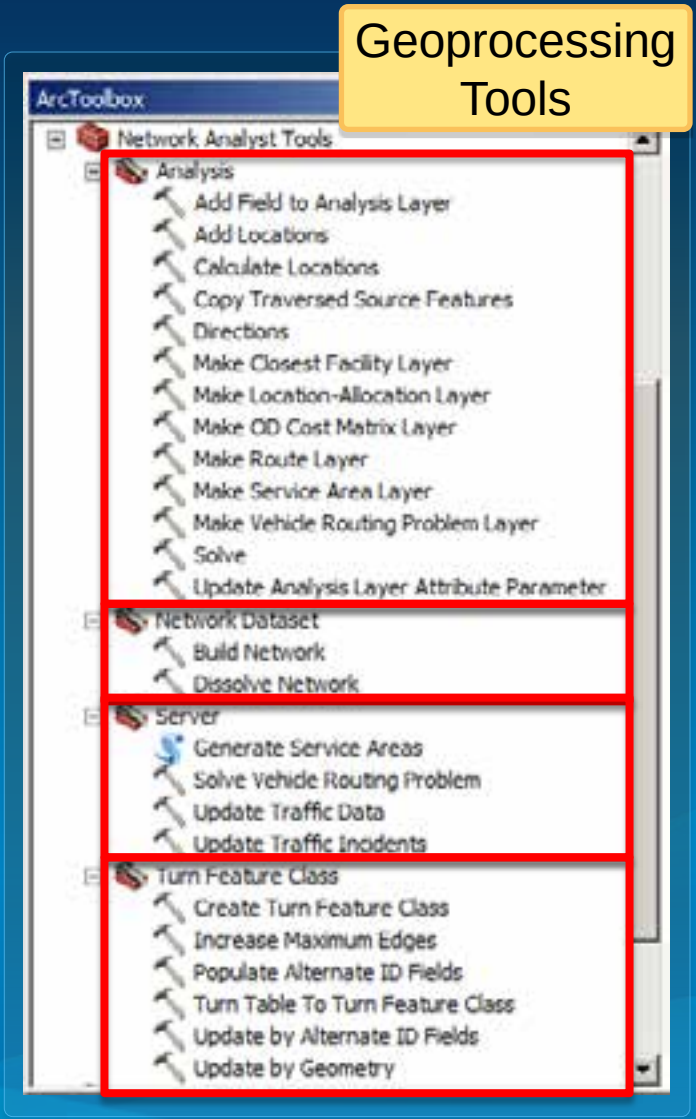
The geoprocessing framework in ArcGIS help

What is Geoprocessing?

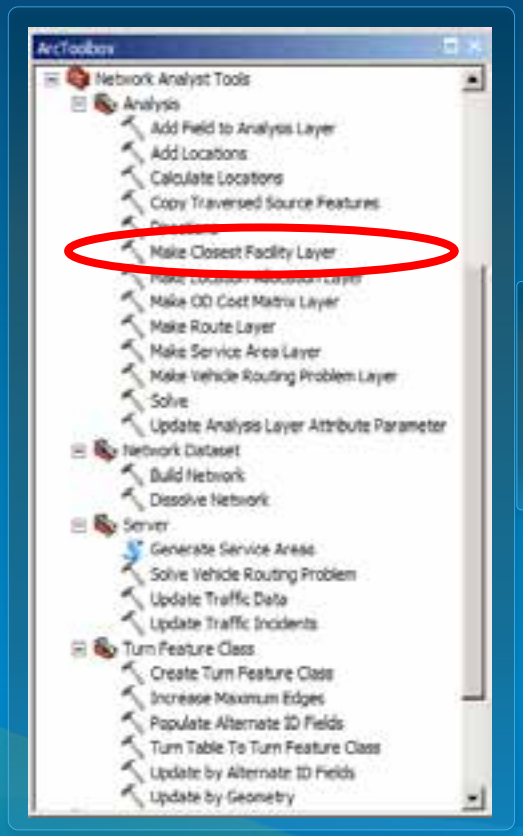


Using Geoprocessing – How?

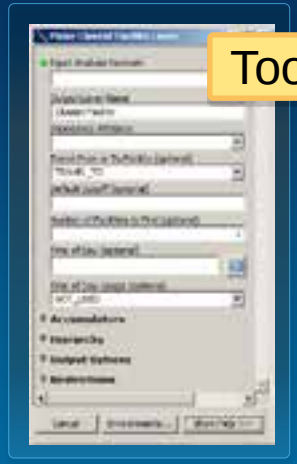
- Accessed through ArcToolbox
- Network Analyst Tools
  - Performing Network Analysis
  - Building networks
  - Publishing services
  - Managing turns



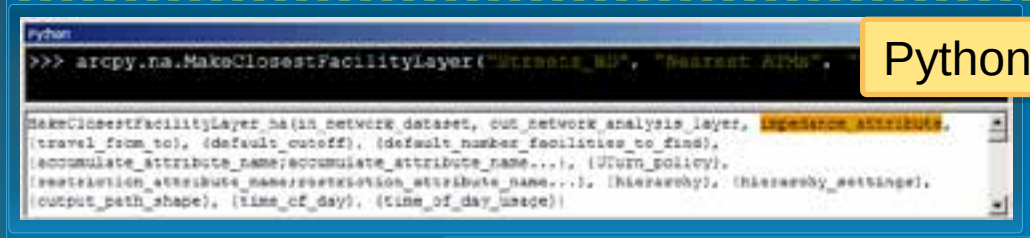
Using Geoprocessing – How?



Single tool execution



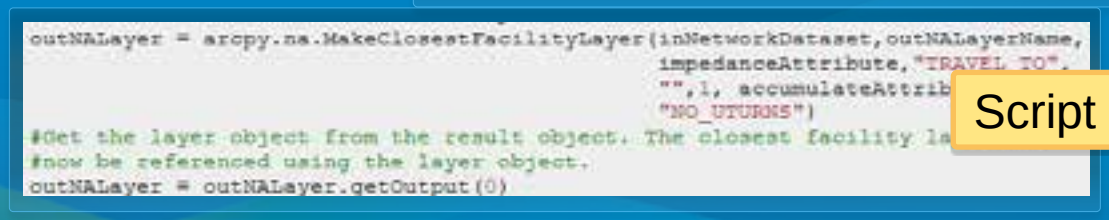
Tool dialog



Python window

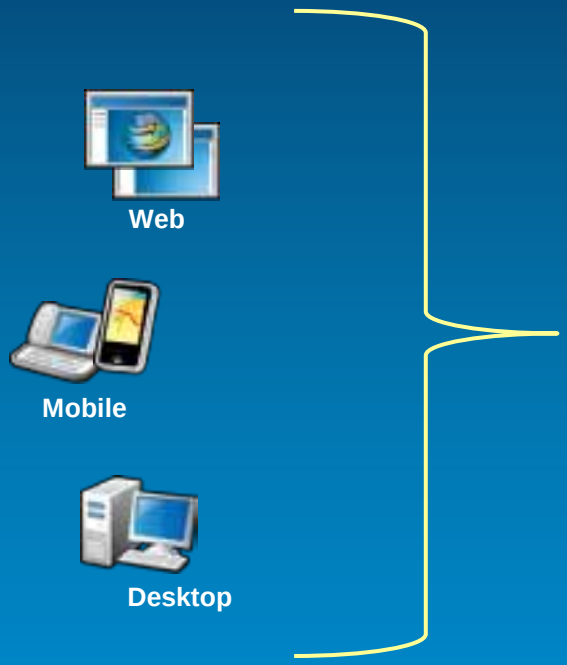
Chain tools

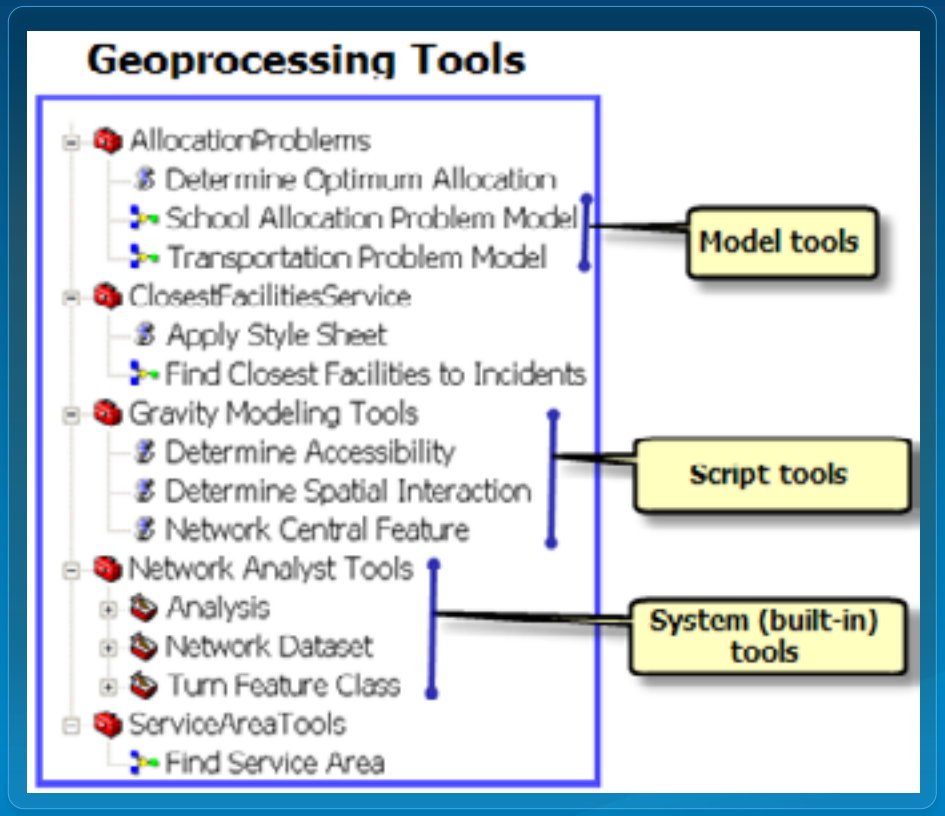
Model



Script

Using Geoprocessing – Where?





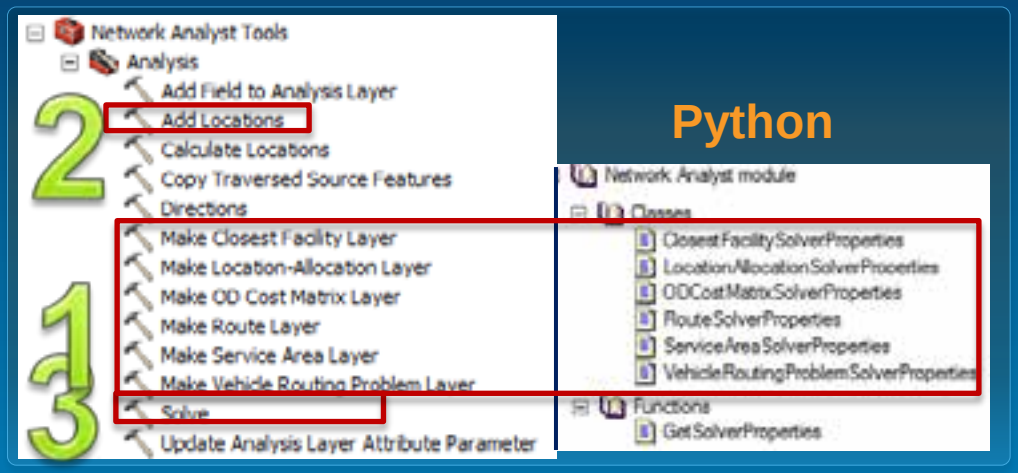
Building Geoprocessing Models

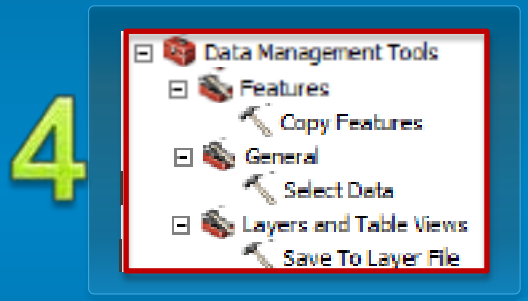
More Information:

What is ModelBuilder? in ArcGIS help

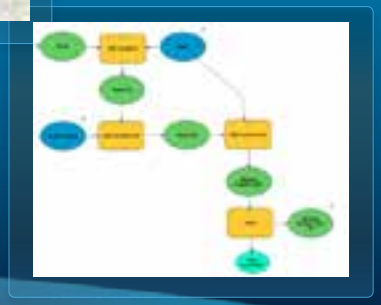
Network Analysis Workflow

1. Make or Edit Network Analysis Layer
2. Add locations to one or more Network Analysis Classes
3. Solve
4. Use the results





Demo: Geoprocessing Models



Authoring a simple route model

Demo: Geoprocessing models - takeaways

- You can easily share models as tools
- 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)

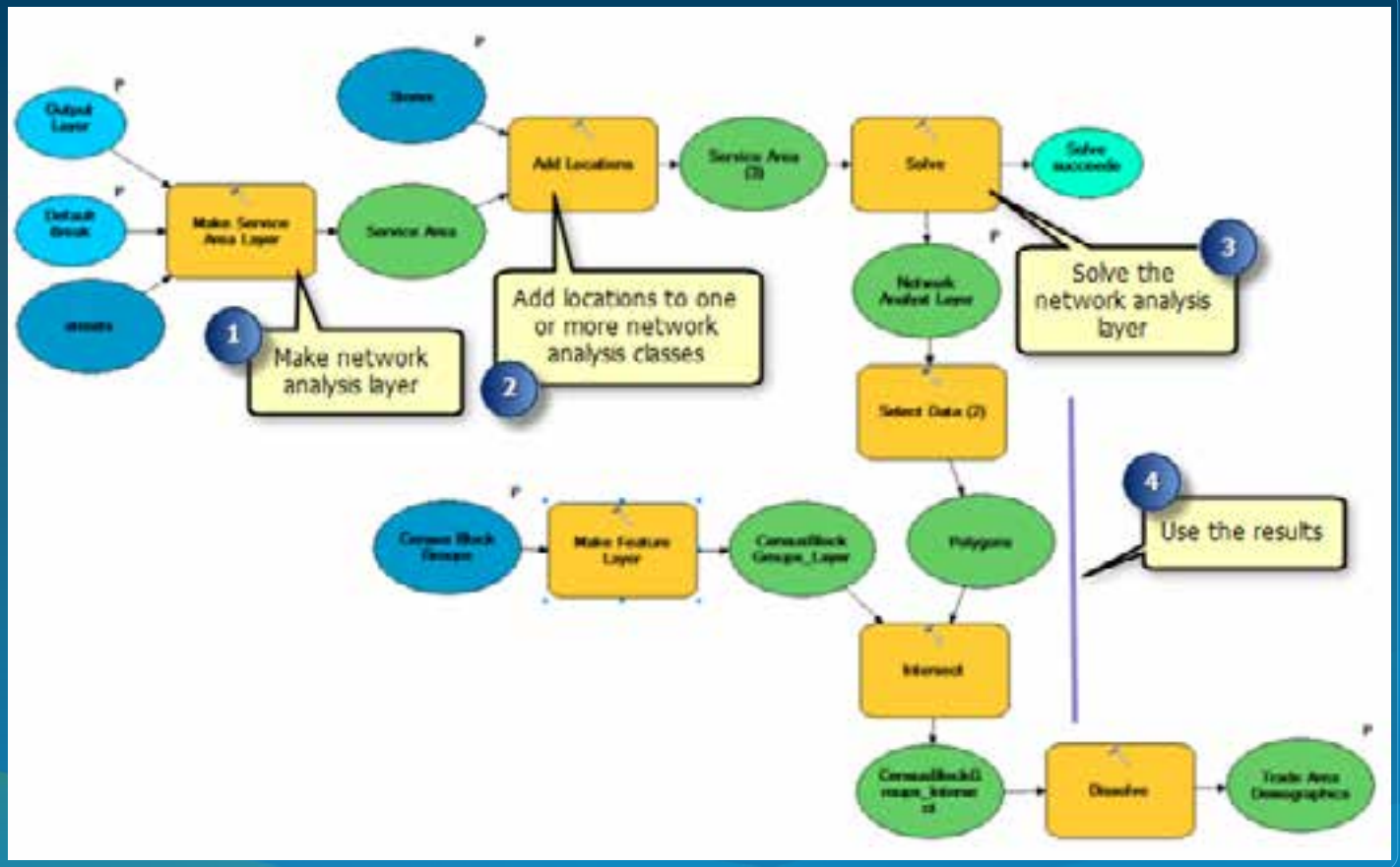
Geoprocessing Models

- 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

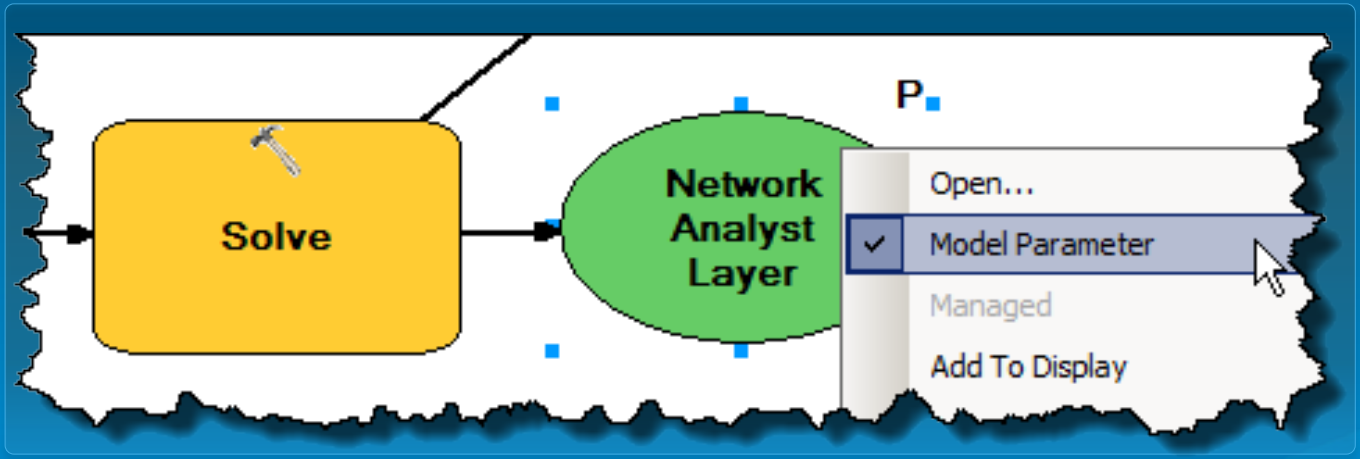
Example Model to perform Service Area Analysis

- **Numbers refer to steps in Network Analysis workflow**



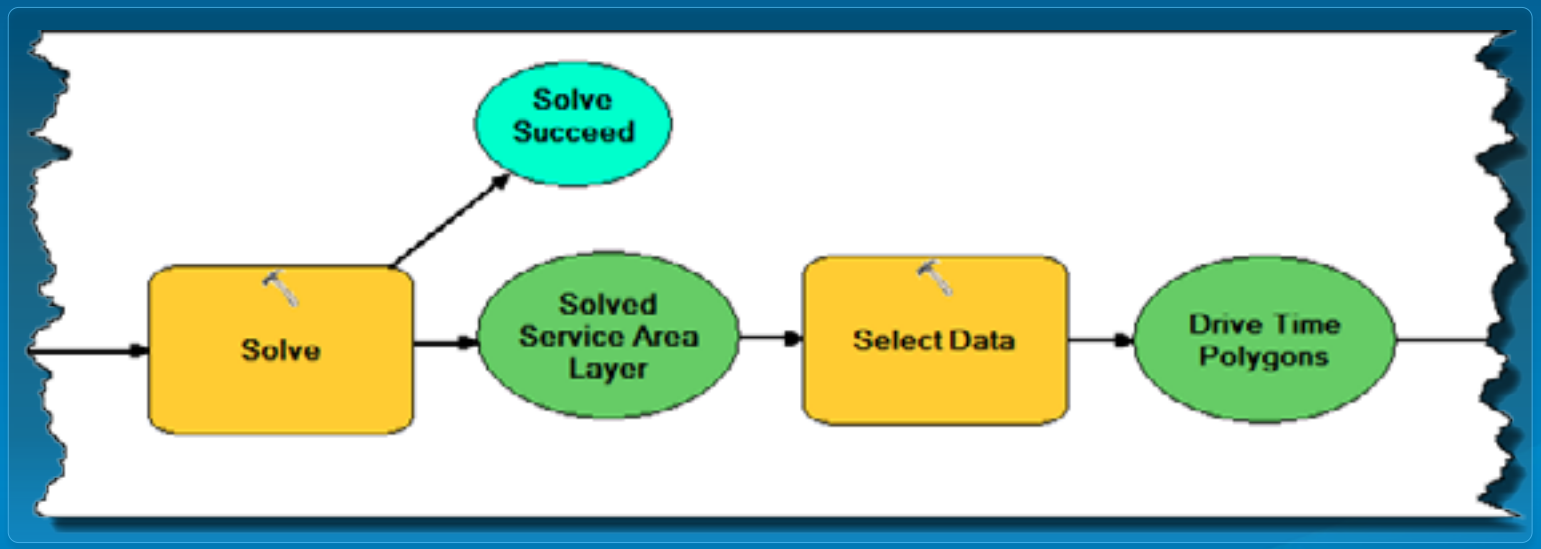
Adding analysis results to ArcMap

- 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



Demo: Geoprocessing Models

Authoring a model to determine multiple routes from a text file containing start and end addresses

Demo: Geoprocessing models - takeaways

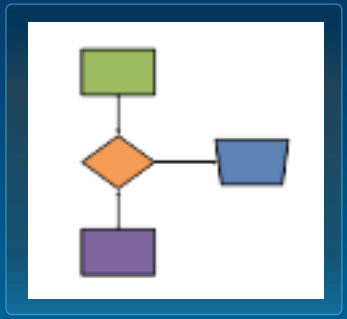
- Use the **Select Data** tool to access sublayers of a network analysis layer
- Incorporate external data (csv in this example) into your analysis
- Automate your workflows without code
- Model tools can be added as buttons on any toolbar
- If network analysis layer is intermediate data, explicitly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

Writing Python Scripts

More Information:

What is Python? in ArcGIS help

Python Scripts

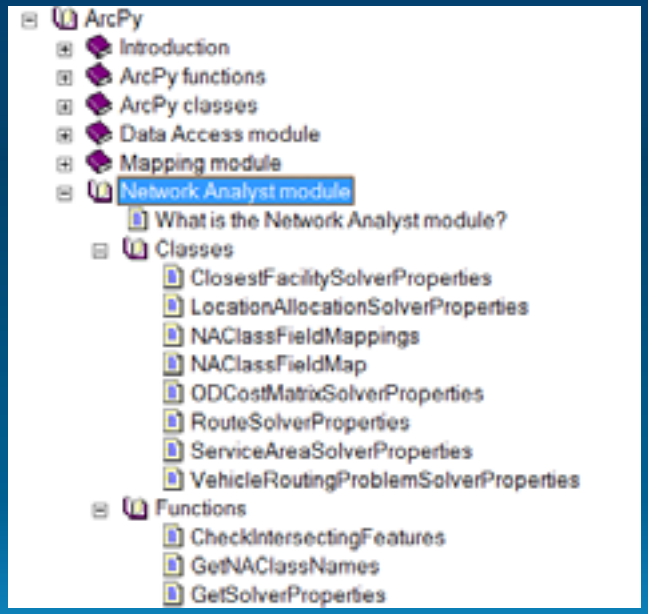


- Used for
  - Conditional logic
  - Looping
  - Cursors, creating geometry
  - Accessing built-in and third party python modules
- ArcPy site package
  - Network Analyst module
  - Access other geoprocessing tools
  - Other useful functions and classes such as Describe
- Python scripts can be run cross platform

What is the Network Analyst Module?

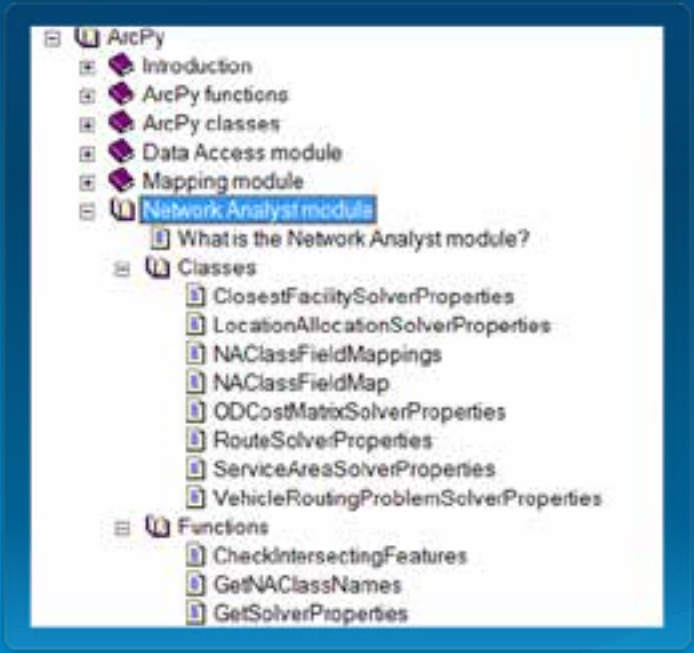
- **Simplify access to Network Analyst functionality from Python**

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

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

Import arcpy module

Python Script - Basic Building Blocks

Check out the Network Analyst Extension

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

Python Script - Basic Building Blocks

Set inputs and outputs

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

#Set local variables
inNetworkDataset = "Transportation/ParisMultimodal_ND"
outNALayerName = "ClosestWarehouse"
impedanceAttribute = "Drivetime"
accumulateAttributeName = ["Meters"]
inFacilities = "Analysis/Warehouses"
inIncidents = "Analysis/Stores"
outLayerFile = "C:/data/output" + "/" + outNALayerName + ".lyr"
```

Python Script - Basic Building Blocks

Make/edit a network analysis layer

```
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)
```

Python Script - Basic Building Blocks

Add locations to network analysis classes

```
#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
facilitiesLayerName = subLayerNames["Facilities"]
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

Solve the network analysis layer

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

Python Script - Basic Building Blocks

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**

```
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)
```

Working with analysis layers within scripts

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

```
# 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

- 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

```
#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`**

10.0

```
barrierFieldMappings = "Name # Precipitation; BarrierType # 1; " + "Attr_%s %s #" % (impedance, scaleFactorField)

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

10.1

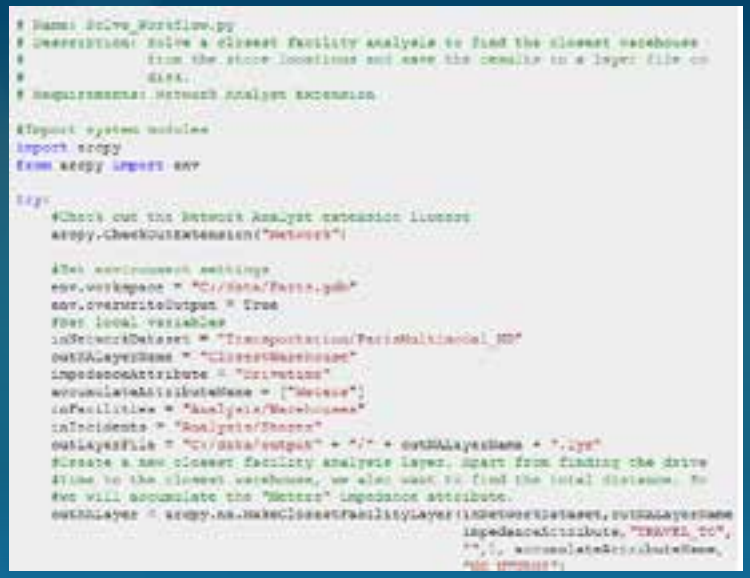
```
naClasses = arcpy.na.GetNAClassNames(routeLayer)
polygonBarriersNAClass = naClasses['PolygonBarriers']
barrierFieldMappings = arcpy.na.NAClassFieldMappings(routeLayer, polygonBarriersNAClass, False,
                                                     arcpy.ListFields(weatherPolygonLayer))
barrierFieldMappings['Name'].defaultValue = "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 SaveToLayerFile geoprocessing tool in the management module

```
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

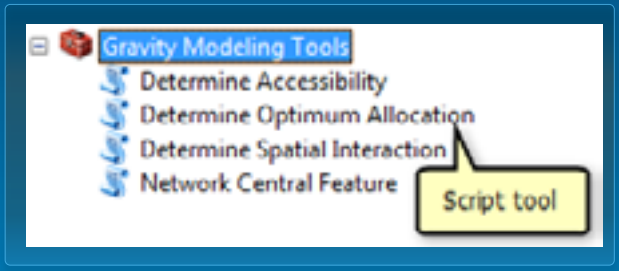
Building Script Tools

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



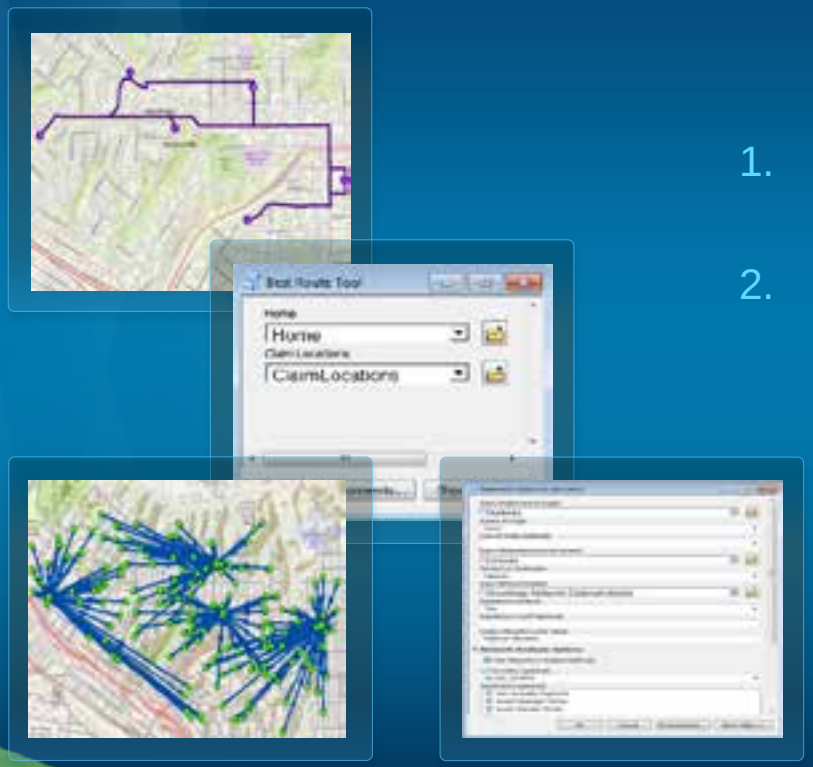
- Script tools behave like any other tool within ArcToolbox
  - Can use script tools in models and vice versa

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(…)**

```
#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)
```



Demo: 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.

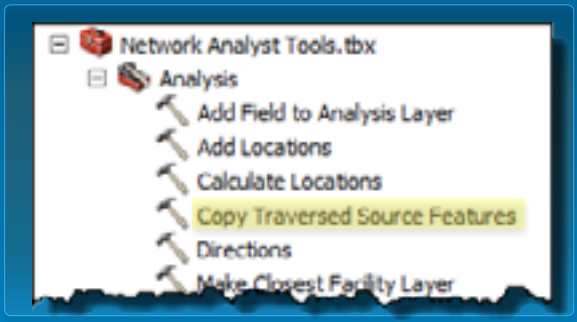
Demo: Script Tool - takeaways

- 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

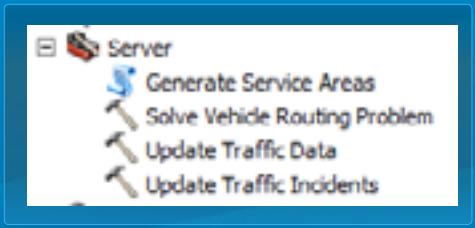
What's new in 10.1

- Network Analyst Python module (arcpy.na)
  - Easy access to Network Analyst functionality from Python, along with helper functions and classes
  - Ability to edit a Network Analysis layer without having to create a new one

- New tools
  - Working with traversal results



  - Easy publishing of GP 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
- Getting to know ArcGIS ModelBuilder book

Network Analyst at the User’s Conference

	Tuesday	Wednesday	Thursday
8:30	Network Analyst: An Introduction Room 3	Network Analyst: Automating Workflows with Geoprocessing Room 10	
9 am	Modeling Real-World Problems with the VRP Solver **		
10 am	Routing with Open Source Data (OSM) ***		
	Network Analyst: Performing Network Analysis Room 3	Network Analyst: Creating Network Datasets Room 9	Network Analyst: Performing Network Analysis Room 4
11 am			Routing in buildings with 3D Networks **
12 pm			
1 pm		Routing with Open Source Data (OSM) ***	
2 pm	Network Analysis with ArcGIS for Server Room 3	Network Analyst: An Introduction Room 4	Using Streetmap Premium *
			Network Analyst: Creating Network Datasets Room 9
3 pm			
4 pm			
5 pm	What's New in Network Analyst 10.1 **		

* Demo Theater – Online GIS – Exhibit Hall C

** Demo Theater – Analysis and Geoprocessing – Exhibit Hall B2

*** Demo Theater – Esri Labs – Exhibit Hall B1

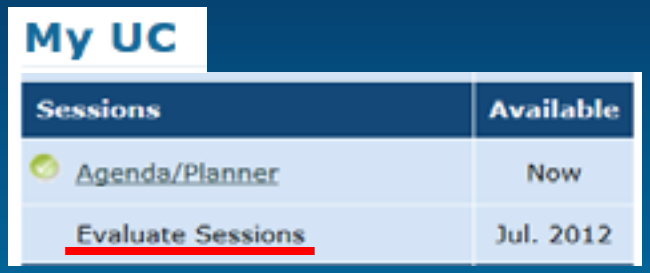
Related Tech Workshops - Geoprocessing

- Geoprocessing Models
  - Getting Started with ModelBuilder
    - Wednesday 1:30 – Ballroom 6B
  - Building Tools with ModelBuilder
    - Wednesday 10:15 – Ballroom 6D
    - Thursday 3:15 – Ballroom 6D

- Python Scripts and Script Tools
  - Python – Getting Started
    - Wednesday 1:30 – Ballroom 6A
  - Building Tools with Python
    - Wednesday 1:30 – Room 28E
    - Thursday 10:15 – Room 9

Steps to evaluate UC sessions

- My UC Homepage >
  "Evaluate Sessions"



- Choose session from planner

OR

- Search for session

by offering ID:

755

www.esri.com/ucsessionsurveys

- Thank you for attending
- Have fun at UC2012
- Open for Questions

- Please fill out the evaluation:

www.esri.com/ucsessionsurveys

Offering ID: 755