



# ArcGIS Enterprise: Publish Your Own Routing Services

Deelesh Mandloi

Max Zeng

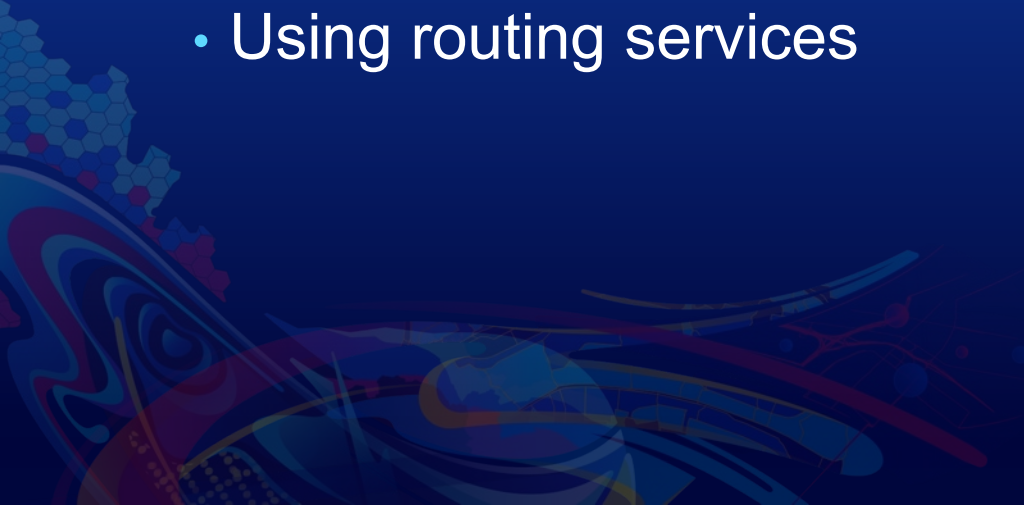
2021 ESRI  
DEVELOPER SUMMIT

Slides and code samples for this workshop on Publishing Your Own  
Routing Services

<http://esriurl.com/ds21rs>



# Agenda

- Routing concepts
  - ArcGIS Enterprise publishing workflow
    - Geoprocessing services
    - Map services with network analysis capability
  - Using routing services
- 

# Routing Concepts





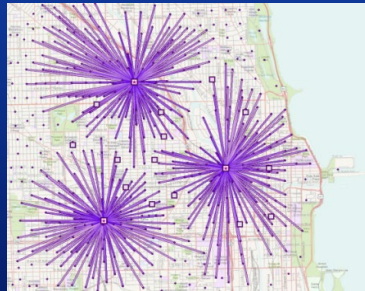
# ArcGIS Network Analyst Extension for transportation analysis

## Coverage

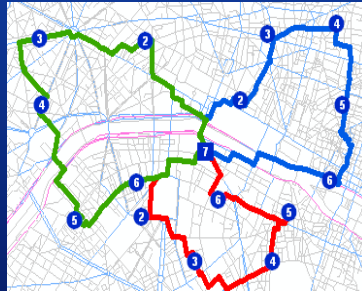


Service Area

## Optimization

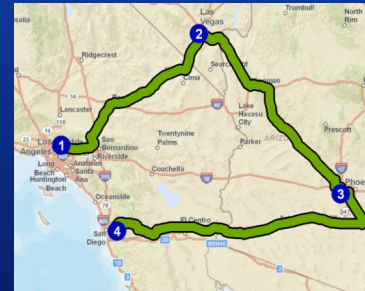


Location-Allocation

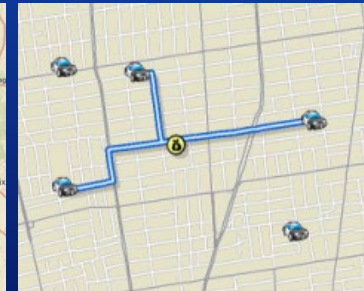


Vehicle Routing  
Problem

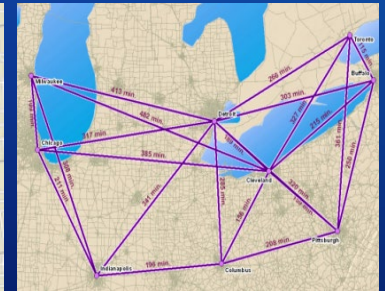
## Point-to-point routing



Route

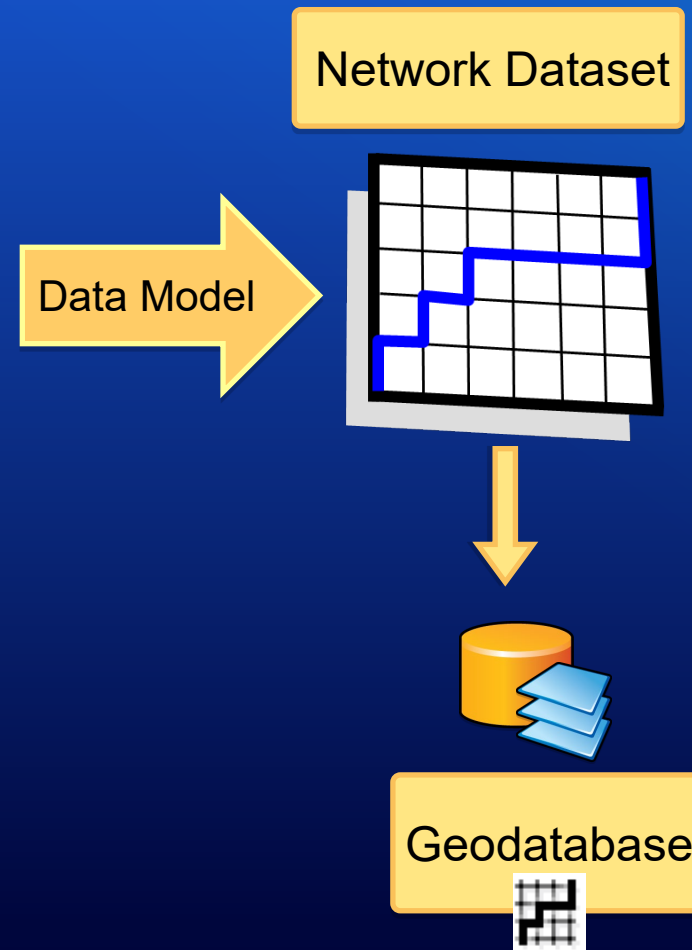
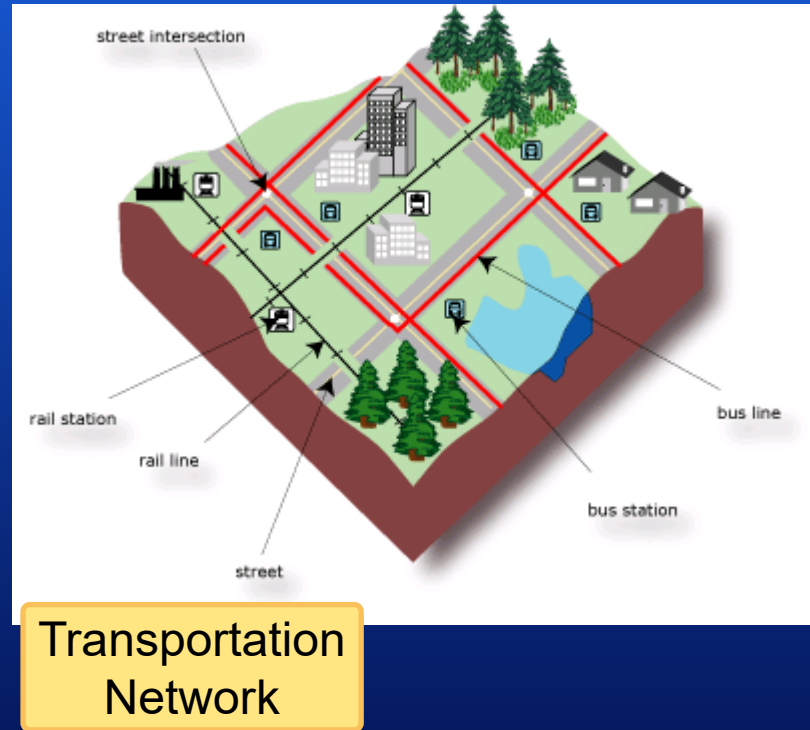


Closest Facility



Origin-Destination  
Cost Matrix

# Analysis is performed on a network dataset



# Where do I get a network dataset?

- Purchase StreetMap Premium for ArcGIS
  - High quality ready-to-use network dataset
  - Can add your own street data
- Build your own
  - Your organization's data
    - Try the ArcGIS Pro Tasks to Create a Local Government Network Dataset
  - TIGER
  - OpenStreetMap
- Use the ArcGIS Online services
  - You don't need a network. You just call the services.



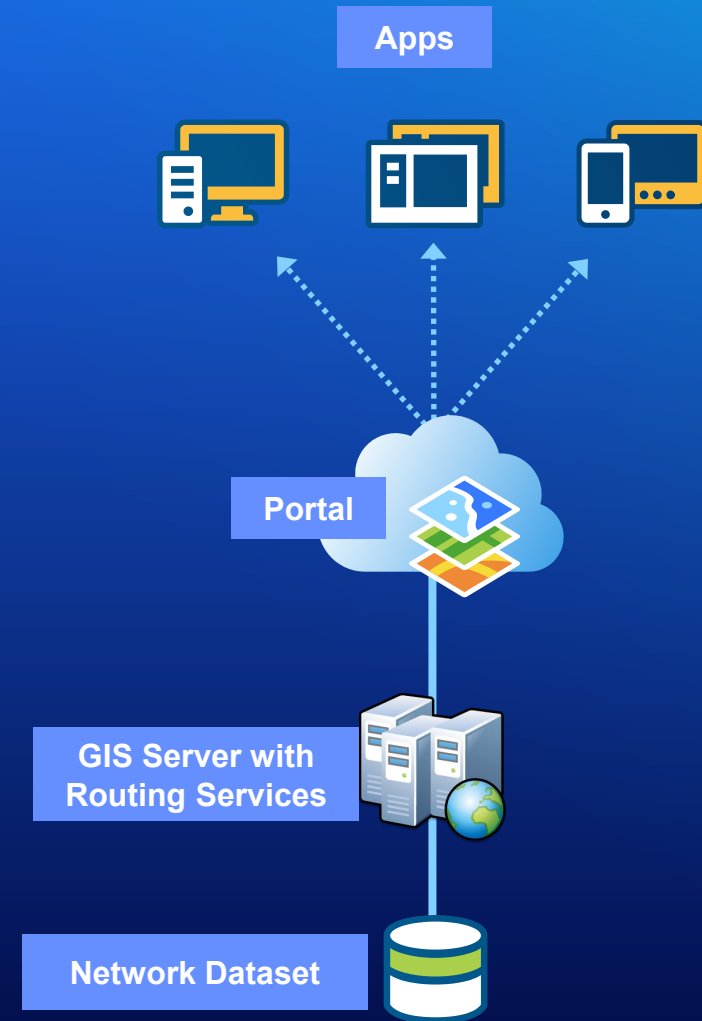
# ArcGIS Enterprise





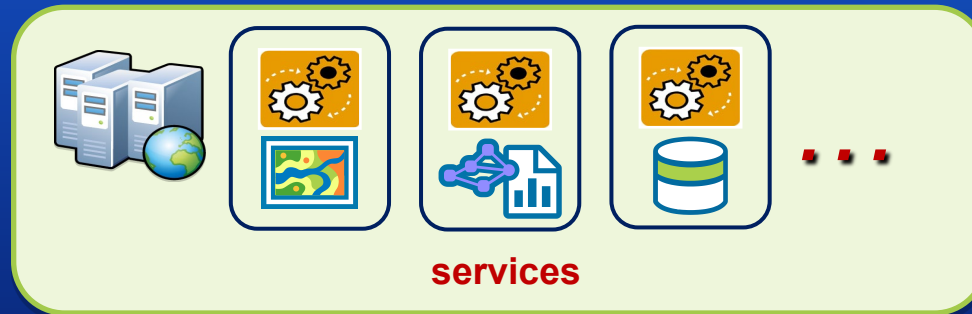
# What is ArcGIS Enterprise?

- Enables “Web GIS” in your infrastructure
  - On-premises or in your own cloud
- Formerly known as ArcGIS for Server



# What are GIS services?

- **GIS service** → GIS resource running on a server
  - vs. GIS application on your local computer



- Share GIS resources to the web as following service types



Map



Feature



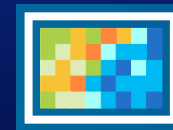
Network  
Analysis



Geodata



Geoprocessing



Image



Locator



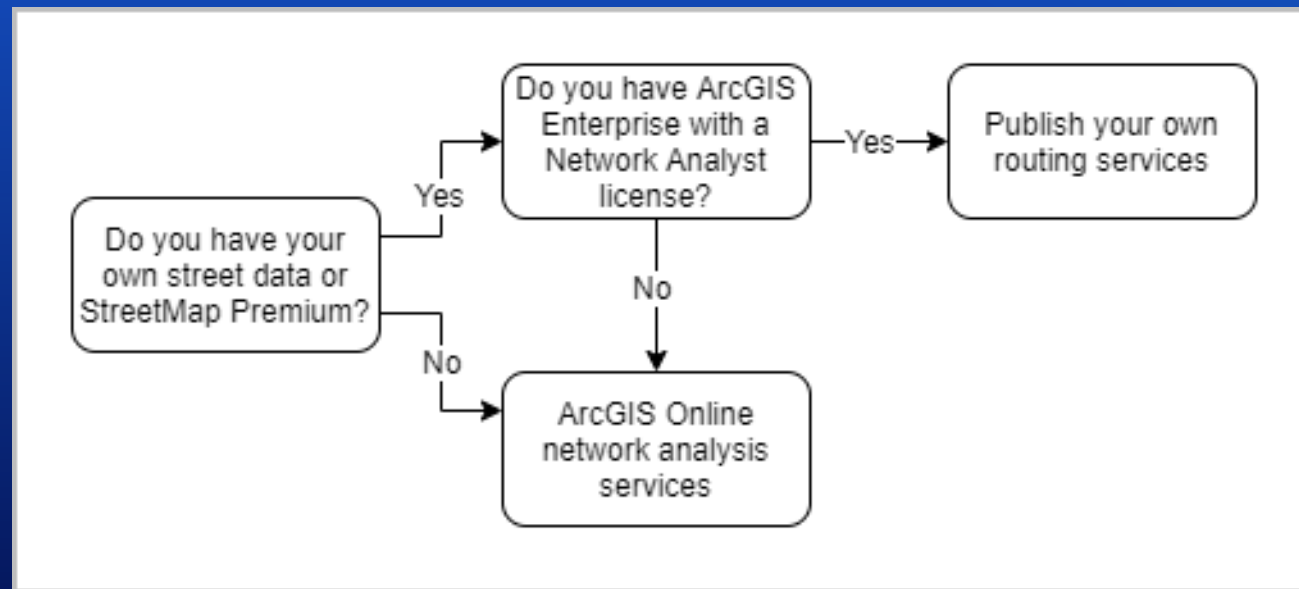
Schematics

# Publishing Routing Services



# ArcGIS Online or ArcGIS Enterprise

- When should I use ArcGIS Online routing services or publish my own routing services ?





# Publishing workflow for GIS services



## Publishing routing services

- Can be published as two different service types

Geoprocessing  
Service

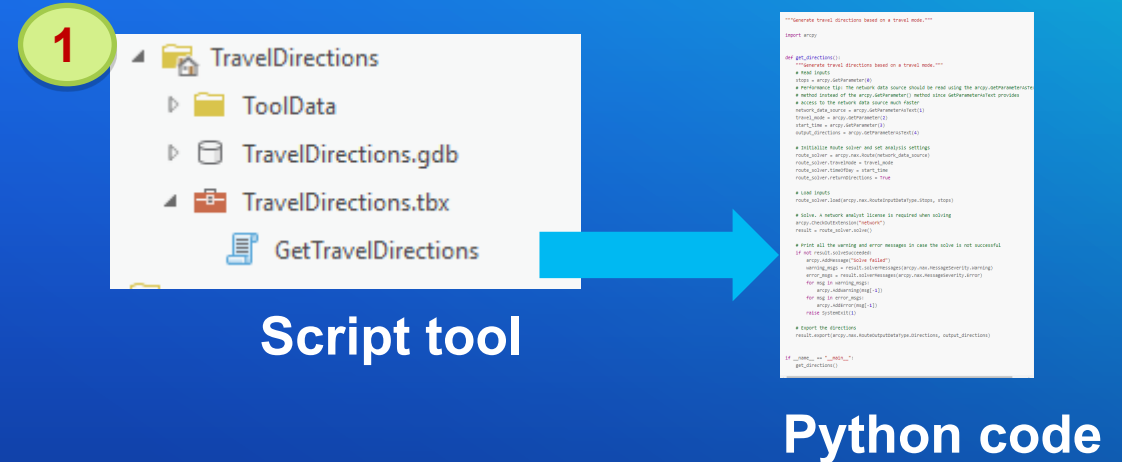
**GPService**

Map Service with  
Network Analysis  
Capability

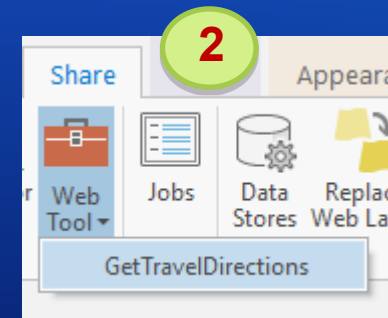
**NAServer**

# Geoprocessing service

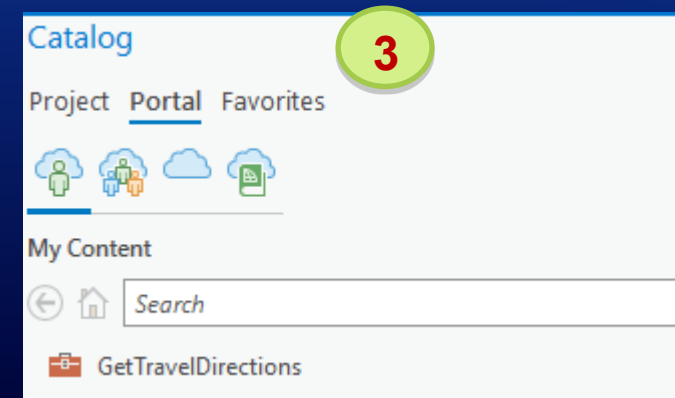
1 Author using geoprocessing script tool based on arcpy and arcpy.nax Python modules



2 Run the tool and publish the result as a web tool



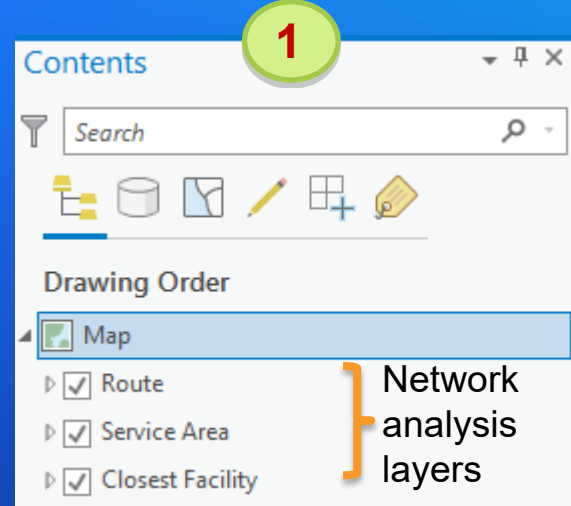
3 Use from Map Viewer, ArcGIS Pro, ArcMap, or programmatically using ArcGIS APIs or SDKs



# Map Service with network analysis capability

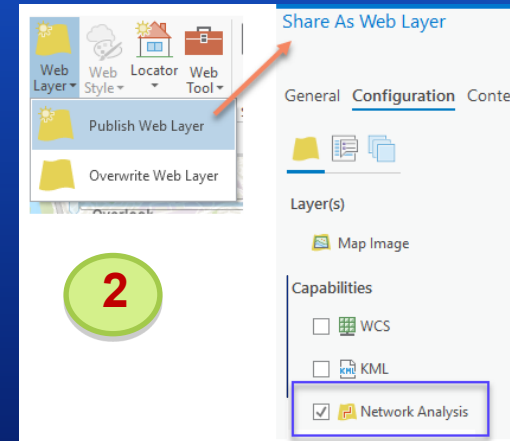
1

Author using an ArcGIS Pro map with one or more network analysis layers



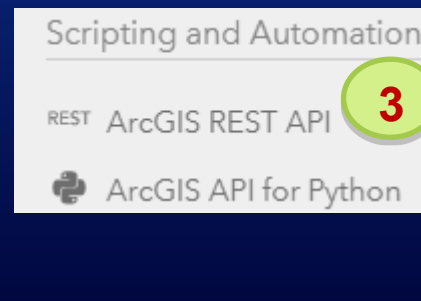
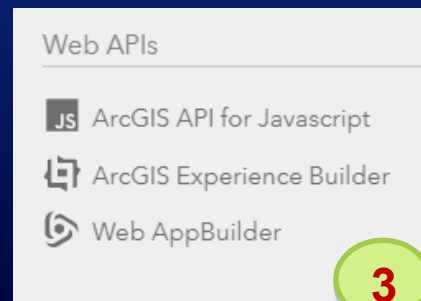
2

Publish the map as a web layer and enable the network analysis capability



3

Use programmatically with ArcGIS APIs and SDKs.





# Publishing routing services should be easy

I have a network dataset and want to publish all the routing services. Is there a single tool that can perform the authoring and publishing of these services?



YES

```
Select Command Prompt

C:\>"C:\Program Files\ArcGIS\Server\tools\PublishRoutingServices\publish routingservices.bat"
-s gisserver.domain.com -P gisportal.domain.com -u admin -p site.admin -o D:\RoutingServices\
ServiceDefinitions -n D:\data\Streets.gdb\Routing\Routing_ND
```

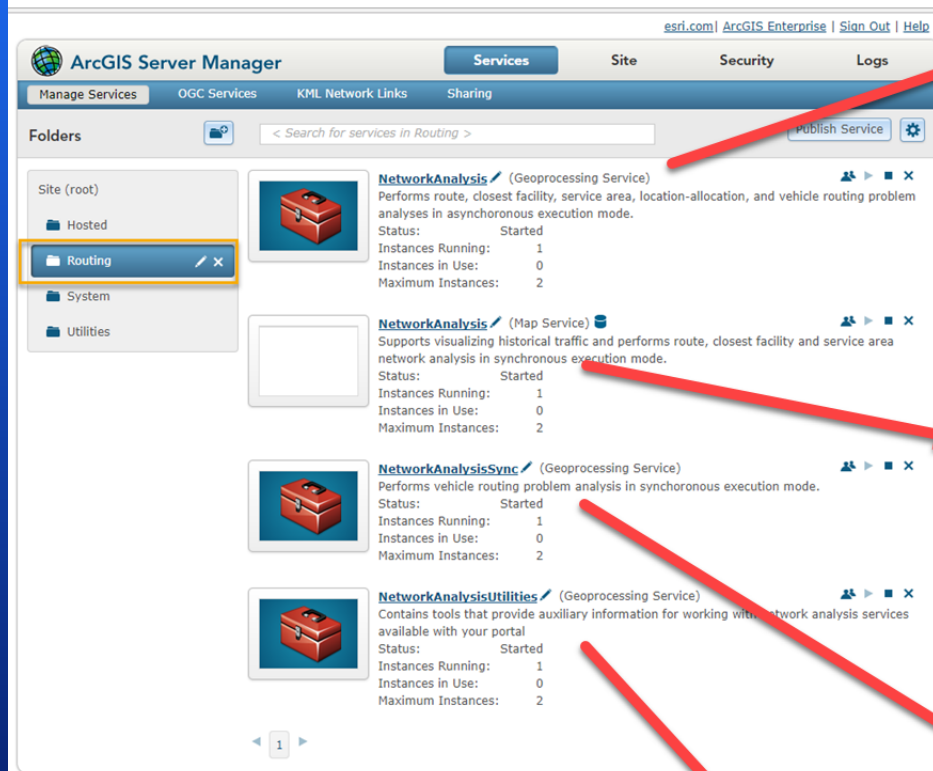
Publish Routing Services utility

# Publish Routing Services utility

- ArcGIS Server command line utility included with ArcGIS Enterprise 10.6 or later
- If publishing to a federated GIS Server, the tool registers the routing services as portal utility services
- Requires Network Analyst extension for ArcGIS Server



# Services created by Publish Routing Services tool



ArcGIS REST Services Directory  
[Home](#) > [services](#) > [Routing](#) > [NetworkAnalysis \(GPService\)](#)  
[JSON](#) | [SOAP](#)

### Routing/NetworkAnalysis (GPService)

**Service Description:** Performs route, closest facility, service area, location-allocation,

**Tasks:**

- [FindRoutes](#)
- [FindClosestFacilities](#)
- [GenerateServiceAreas](#)
- [SolveLocationAllocation](#)
- [SolveVehicleRoutingProblem](#)
- [GenerateOriginDestinationCostMatrix](#)

**Execution Type:** `esriExecutionTypeAsynchronous`

**Result Map Server Name:**

**MaximumRecords:** 1000

**Child Resources:** [Info](#)

ArcGIS REST Services Directory  
[Home](#) > [services](#) > [Routing](#) > [NetworkAnalysis \(NAServer\)](#)  
[JSON](#) | [SOAP](#)

### Routing/NetworkAnalysis (NAServer)

**Service Description :**

**Route Layers:**

- [Route](#)

**Closest Facility Layers:**

- [ClosestFacility](#)

**Service Area Layers:**

- [ServiceArea](#)

**Child Resources:** [Info](#)

ArcGIS REST Services Directory  
[Home](#) > [services](#) > [Routing](#) > [NetworkAnalysisSync \(GPService\)](#)  
[JSON](#) | [SOAP](#)

### Routing/NetworkAnalysisSync (GPService)

**Service Description:** Performs vehicle routing problem analysis in synchronous

**Tasks:**

- [SolveVehicleRoutingProblem](#)

**Execution Type:** `esriExecutionTypeSynchronous`

**Result Map Server Name:**

**MaximumRecords:** 1000

**Child Resources:** [Info](#)

ArcGIS REST Services Directory  
[Home](#) > [services](#) > [Routing](#) > [NetworkAnalysisUtilities \(GPService\)](#)  
[JSON](#) | [SOAP](#)

### Routing/NetworkAnalysisUtilities (GPService)

**Service Description:** Contains tools that provide auxiliary information for

**Tasks:**

- [GetToolInfo](#)
- [GetTravelModes](#)

**Execution Type:** `esriExecutionTypeSynchronous`

**Result Map Server Name:**

**MaximumRecords:** 1000

**Child Resources:** [Info](#)

Asynchronous  
geoprocessing  
service

Map service with  
network analysis  
capability

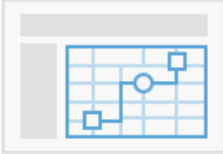
Synchronous  
geoprocessing  
service

Service to get  
travel modes

# Registering services with your portal

- Configure as Directions and Routing utility services in your portal
- Allows all apps to discover the network analysis services
- Required by many apps in order to use the services
- Configured by Publish Routing Services tool if publishing services to a federated GIS Server

Directions and Routing



Configure your routing services. Choose the routing service type

Route
Route
Route (Asynchronous)
Closest Facility
Closest Facility (Asynchronous)
Service Area
Service Area (Asynchronous)
Vehicle Routing Problem
Vehicle Routing Problem (Asynchronous)
Location Allocation (Asynchronous)
Routing Utilities
Traffic
Origin Destination Cost Matrix (Asynchronous)

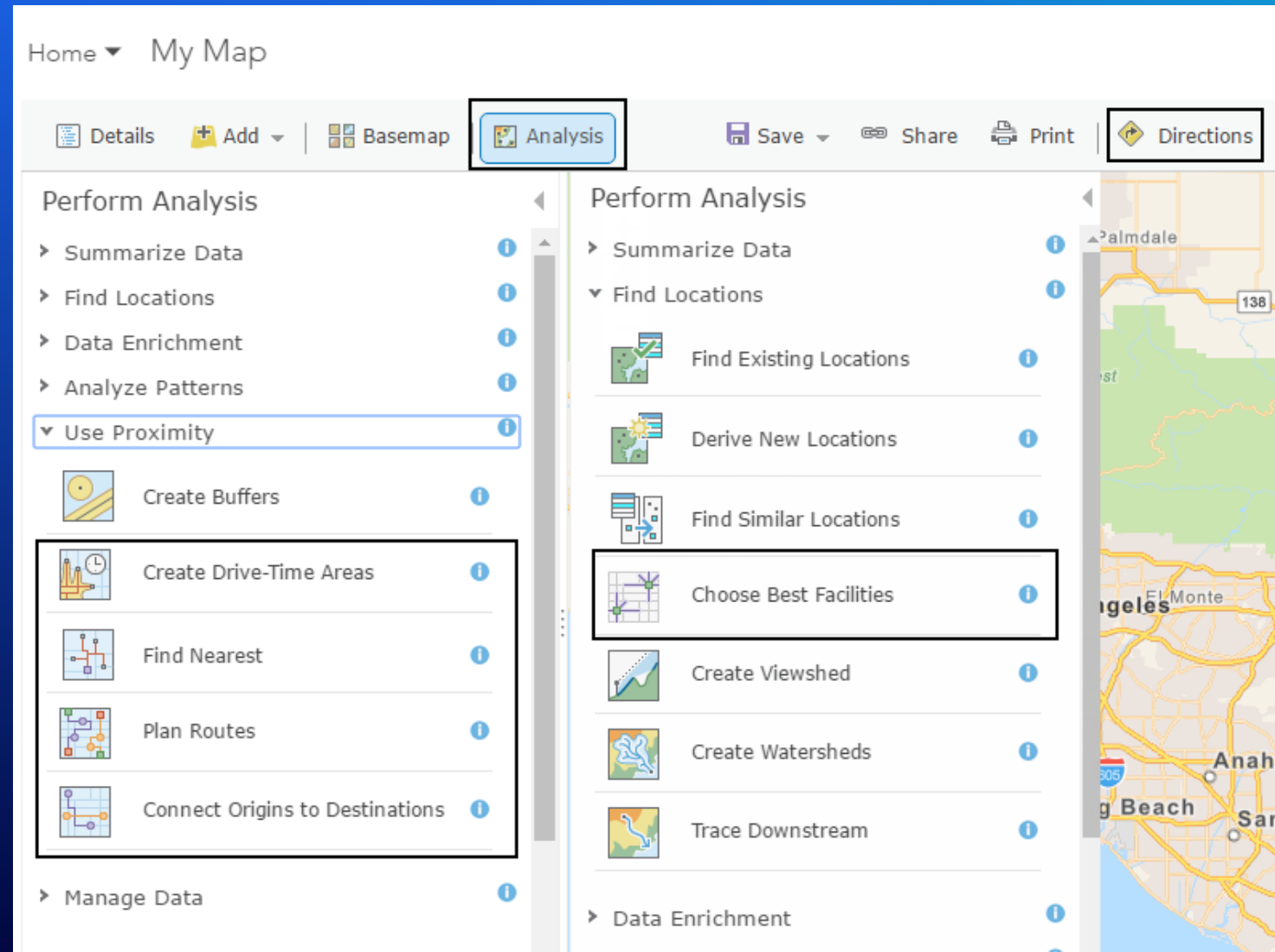


## Updating services created by Publish Routing Services utility

- Manually delete existing routing services then recreate new ones using the publish routing services utility
  - Upgrading to newer versions of ArcGIS Enterprise
  - Upgrading street data (for example, using newer version of StreetMap Premium)

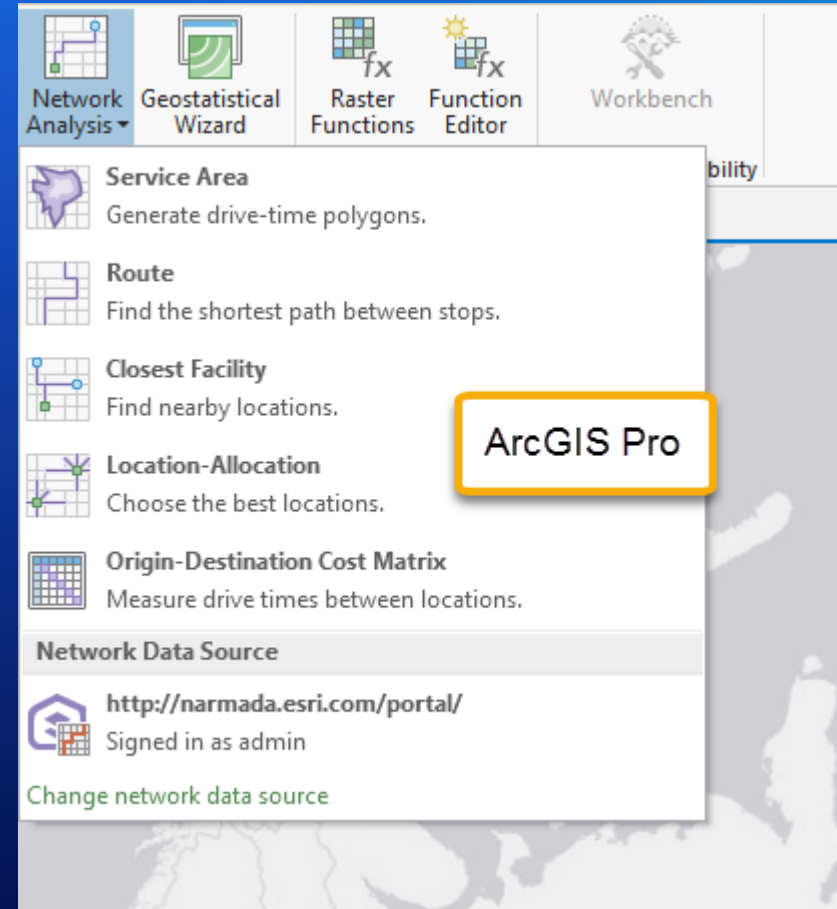
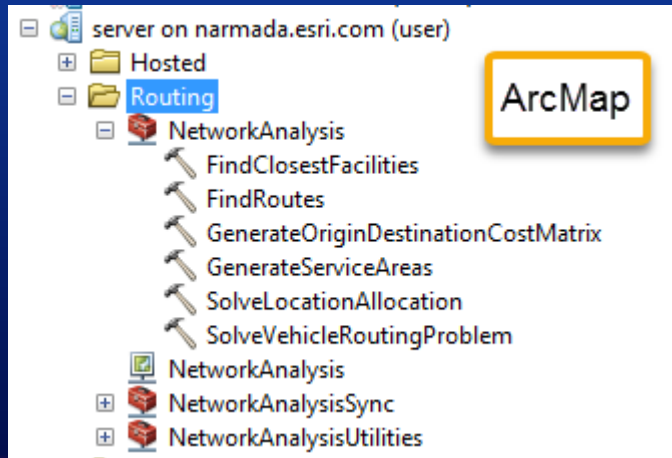
# Using services in Map Viewer

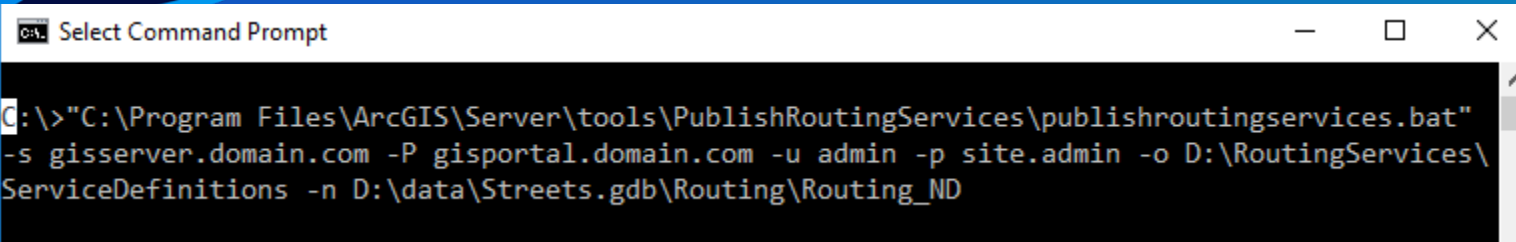
- Directions button
- Analysis tools
  - Most tools in the Use Proximity toolset
  - Choose Best Facilities tool in Find Locations toolset
  - Summarize Nearby tool in Summarize Data toolset



# Using services in ArcGIS Desktop

- Use from Network Analysis gallery in the Analysis tab within ArcGIS Pro
- Use from an ArcGIS Server connection in ArcMap





```
C:\>"C:\Program Files\ArcGIS\Server\tools\PublishRoutingServices\publish routingservices.bat"
-s gisserver.domain.com -P gisportal.domain.com -u admin -p site.admin -o D:\RoutingServices\
ServiceDefinitions -n D:\data\Streets.gdb\Routing\Routing_ND
```

# Demo: Publish routing services utility

Max Zeng



# Life beyond Publish Routing Services utility

- Publish your own geoprocessing service
  - If you need the service to perform additional analysis along with network analysis. For example, find a route and calculate an elevation profile for the route
  - Combine multiple network analyses in a single service. For example, determine accessible open houses and then find the best route to visit them
- Author a geoprocessing script tool using arcpy and arcpy.nax Python modules

# Network Analysis Workflow with arcpy.nax Python Module

1. Initialize the analysis object (based on a specific network data source)
2. Set the properties for the analysis
3. Load the inputs
4. Solve the analysis
5. Work with the results

*Common to all the network analyses*

# arcpy.nax Analysis (Solver) Classes

- Easy-to-use python objects for network analysis
- Analysis class for each solver
  - Set properties
  - Load inputs
  - Solve
- Analysis class for solve results
  - Access outputs

## OriginDestinationCostMatrix

accumulateAttributeNames  
allowSaveLayerFile  
defaultDestinationCount  
defaultImpedanceCutoff  
distanceUnits  
ignoreInvalidLocations  
lineShapeType  
networkDataSource  
overrides  
searchQuery  
searchTolerance  
searchToleranceUnits  
timeOfDay  
timeUnits  
timeZone  
travelMode

count()  
fieldMappings()  
fieldNames()  
insertCursor()  
load()  
solve()

## OriginDestinationCostMatrixResult

isPartialSolution  
solveSucceeded

count()  
export()  
fieldNames()  
saveAsLayerFile()  
searchCursor()  
solverMessages()

Properties

Methods

# Example script tool for generating travel directions

```
"""Generate travel directions based on a travel mode."""

import arcpy

def get_directions():
    """Generate travel directions based on a travel mode."""
    # Read inputs
    stops = arcpy.GetParameter(0)
    # Performance tip: The network data source should be read using the arcpy.GetParameterAsText() method instead of the
    # arcpy.GetParameter() method since GetParameterAsText provides access to the network data source much faster
    network_data_source = arcpy.GetParameterAsText(1)
    travel_mode = arcpy.GetParameter(2)
    start_time = arcpy.GetParameter(3)
    output_directions = arcpy.GetParameterAsText(4)

    # Initialize Route solver and set analysis settings
    route_solver = arcpy.nax.Route(network_data_source)
    route_solver.travelMode = travel_mode
    route_solver.timeOfDay = start_time
    route_solver.returnDirections = True

    # Load inputs
    route_solver.load(arcpy.nax.RouteInputDataType.Stops, stops)

    # Solve. A network analyst license is required when solving
    arcpy.CheckOutExtension("network")
    result = route_solver.solve()

    # Print all the warning and error messages in case the solve is not successful
    if not result.solveSucceeded:
        arcpy.AddMessage("Solve failed")
        warning_msgs = result.solverMessages(arcpy.nax.MessageSeverity.Warning)
        error_msgs = result.solverMessages(arcpy.nax.MessageSeverity.Error)
        for msg in warning_msgs:
            arcpy.AddWarning(msg[-1])
        for msg in error_msgs:
            arcpy.AddError(msg[-1])
        raise SystemExit(1)

    # Export the directions
    result.export(arcpy.nax.RouteOutputDataType.Directions, output_directions)

if __name__ == "__main__":
    get_directions()
```

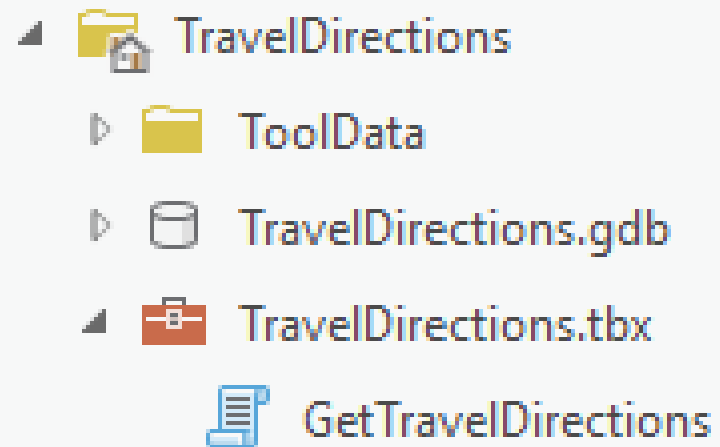
# Tips and tricks when authoring geoprocessing services

- Use a network dataset layer instead of a network dataset catalog path
  - Significant performance improvement as the connection to the network dataset is established once during the startup of the service
- Write a geoprocessing script tool using arcpy.nax Python module
  - Faster execution times as compared to services created from geoprocessing model tool
  - Use Feature Set data type for inputs
  - Write outputs to memory-based workspace instead of a file geodatabase
  - Avoid writing too many messages from your script tool



# Tips and tricks when publishing geoprocessing services

- Set Input mode for network dataset as constant
  - Hides the network dataset from service consumers
- Register the folder containing the network dataset as a data store instead of copying data
  - You control the location of the data and allows for easier data updates
- Do not store the network dataset on a file share
  - If the ArcGIS Site has multiple machines, manually copy the network dataset to a local storage on all the machines



# Demo: Create a geoprocessing service

Max Zeng

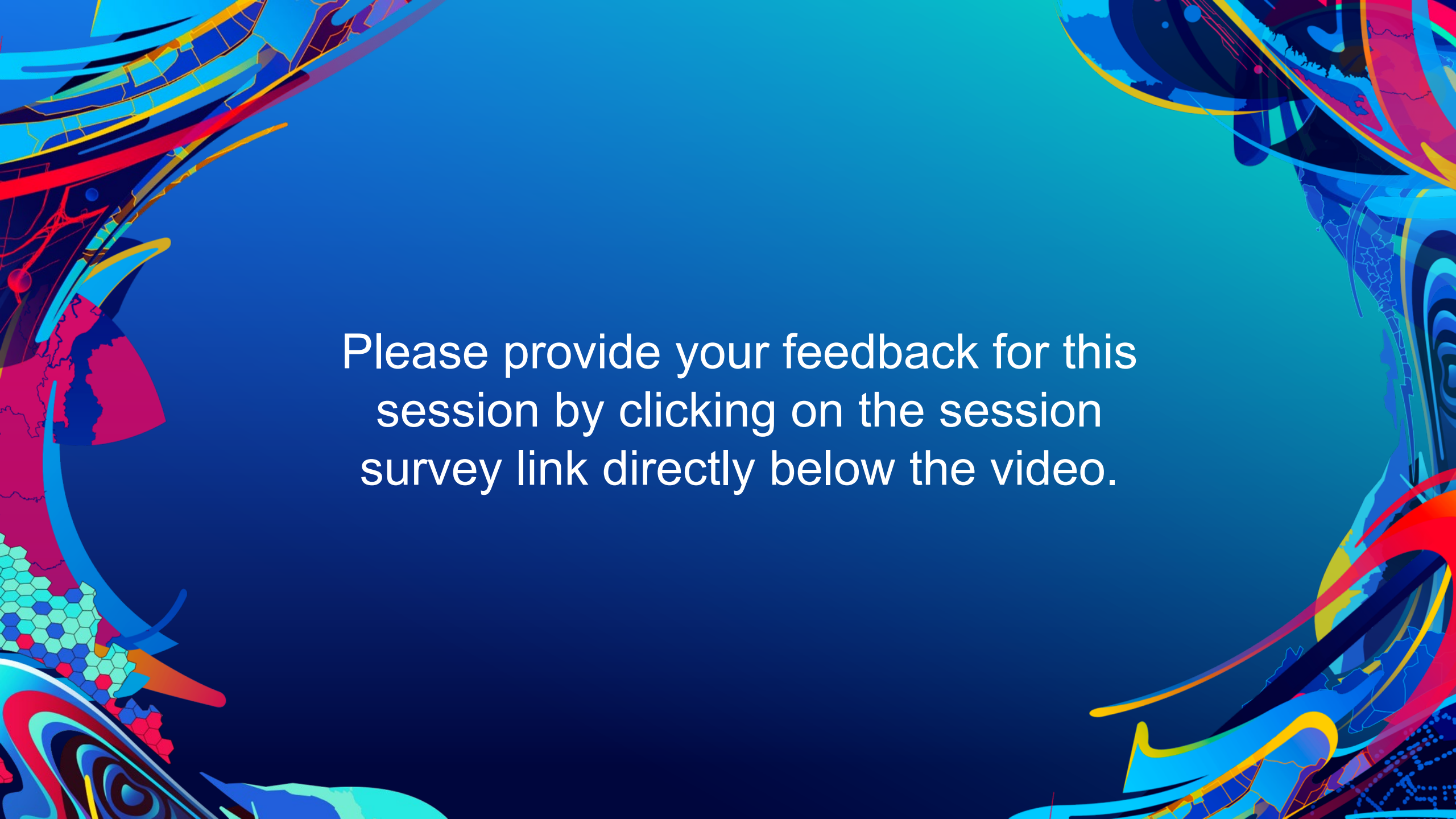
# Help and Resources

- [Network Analysis Concepts](#)
- [Network Analysis using web services](#)
- [Publishing routing services](#)
- [REST API doc](#)

Slides and code samples for this workshop on Publishing Your Own Routing Services

<http://esriurl.com/ds21rs>



The background is a vibrant, abstract composition. It features a central area of solid blue and teal. The left and right sides are framed by complex, colorful patterns. On the left, there are swirling shapes in red, yellow, and blue, along with a section of a hexagonal grid in shades of green and blue. On the right, there are more swirling patterns in blue, red, and yellow, with some areas resembling a stylized face or mask. The overall effect is dynamic and modern.

Please provide your feedback for this session by clicking on the session survey link directly below the video.