



Network Analysis in ArcGIS Engine and ArcGIS Desktop (Deep Dive)

Michael Rice

Matt Crowder

General Information

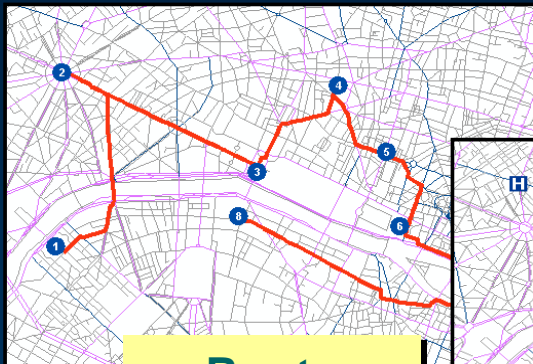
- **Prerequisites**
 - General understanding of Network Analyst
 - Knowledge of ArcObjects programming
- **Code presentation**
 - Code workflows presented in slides
 - Based on EDN samples
 - Caveats
 - No type casts
 - No fully qualified namespaces

Agenda

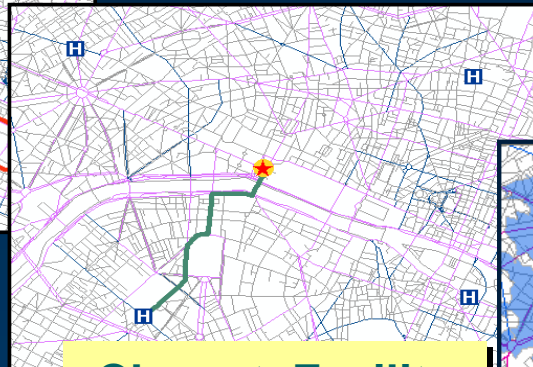
- **Introduction**
- **Engine Controls**
- **Network Analyst API**
- **Network Analyst Customization**
- **Support & Resources**
- **Questions**

What is Network Analyst?

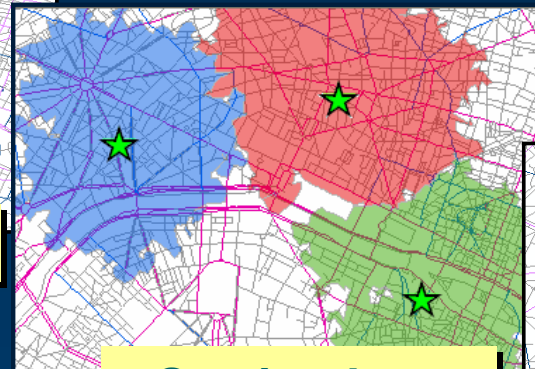
- Extension for analyzing transportation networks
 - Four network solvers



Route



Closest Facility



Service Area

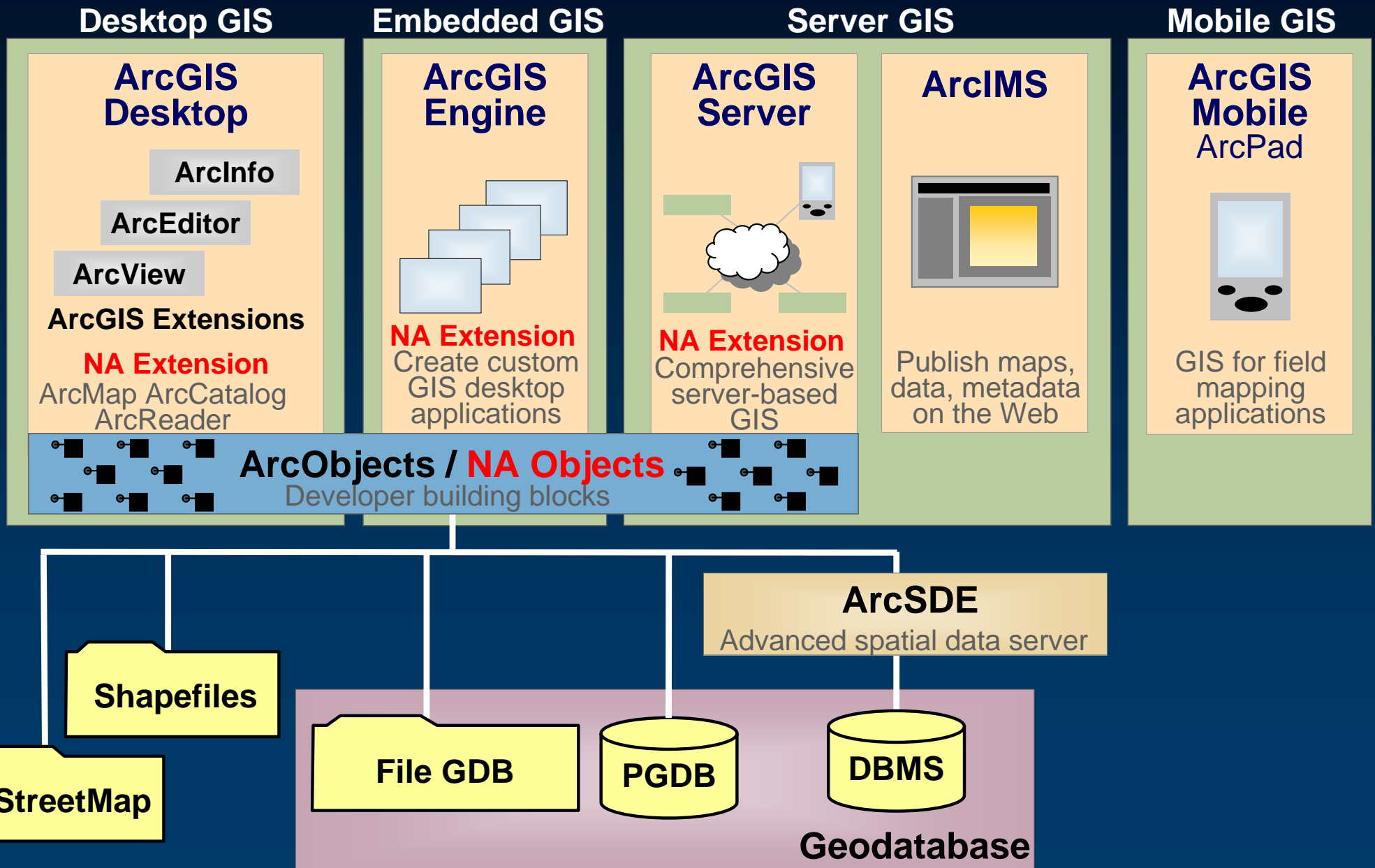


Origin-Destination (OD) Cost Matrix



- Uses Network Datasets

Overview



Agenda

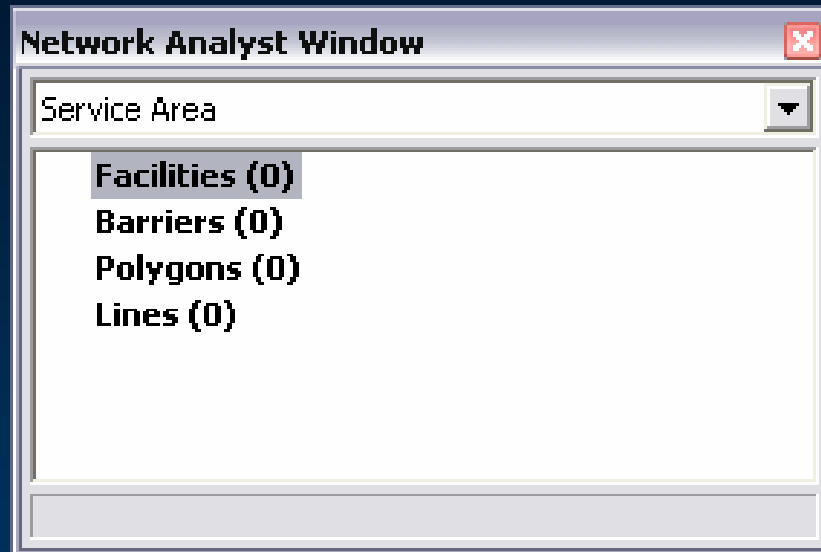
- Introduction
- **Engine Controls**
- Network Analyst API
- Network Analyst Customization
- Support & Resources
- Questions

Using Network Analyst in ArcGIS Engine

- **Uses same underlying Network Analyst objects as ArcMap**
- **Requires NetworkEngine license**
- **Out-of-the-box network UI controls at 9.2**

Network Analyst Controls in ArcGIS Engine 9.2

- **NAWindow**



- **Network Analyst Toolbar**



Demo

Agenda

- Introduction
- Engine Controls
- **Network Analyst API**
- Network Analyst Customization
- Support & Resources
- Questions

Network Dataset

- **Data model for representing transportation networks**
- **References source feature classes (e.g., streets)**
- **Stores connectivity**
- **Stores attribute information**
- **Found in ESRI.ArcGIS.Geodatabase library**

Opening Network Datasets

- **Open using dataset extensibility framework**
- **Pattern depends on network dataset workspace type**
 - **Shapefile and StreetMap**
 - **Network dataset managed by a workspace extension**
 - **Use IWorkspaceExtensionManager and IDatasetContainer2**
 - **Geodatabase**
 - **Network dataset managed by a feature dataset extension**
 - **Use IFeatureDatasetExtensionContainer and IDatasetContainer2**

Opening shapefile/StreetMap network datasets (C#)

// Get Workspace Extension Manager

```
IWorkspaceExtensionManager wsExtManager = networkWorkspace;
```

// Get Workspace Extension

```
UID uid = new UIDClass();
```

```
uid.Value = "esriGeoDatabase.NetworkDatasetWorkspaceExtension";
```

```
IWorkspaceExtension wsExtension = wsExtManager.FindExtension(uid);
```

// Get NetworkDataset

```
IDatasetContainer2 dsContainer = wsExtension;
```

```
INetworkDataset networkDataset = dsContainer.get_DatasetByName(  
    esriDatasetType.esriDTNetworkDataset, "Streets_ND");
```

Opening Network Datasets

- **Open using dataset extensibility framework**
- **Pattern depends on network dataset workspace type**
 - **Shapefile and StreetMap**
 - **Network dataset managed by a workspace extension**
 - **Use IWorkspaceExtensionManager and IDatasetContainer2**
 - **Geodatabase**
 - **Network dataset managed by a feature dataset extension**
 - **Use IFeatureDatasetExtensionContainer and IDatasetContainer2**

Opening geodatabase network datasets (C#)

// Get Feature Workspace

```
IFeatureWorkspace featureWorkspace = networkWorkspace;
```

// Get Feature Dataset Extension Container

```
IFeatureDatasetExtensionContainer fdsExtContainer =  
featureWorkspace.OpenFeatureDataset(featureDatasetName);
```

// Get Feature Dataset Extension

```
IFeatureDatasetExtension fdsExtension =  
fdsExtContainer.FindExtension(esriDatasetType.esriDTNetworkDataset);
```

// Get NetworkDataset

```
IDatasetContainer2 dsContainer = fdsExtension;  
INetworkDataset networkDataset = dsContainer.get_DatasetByName(  
esriDatasetType.esriDTNetworkDataset, "Streets_ND");
```

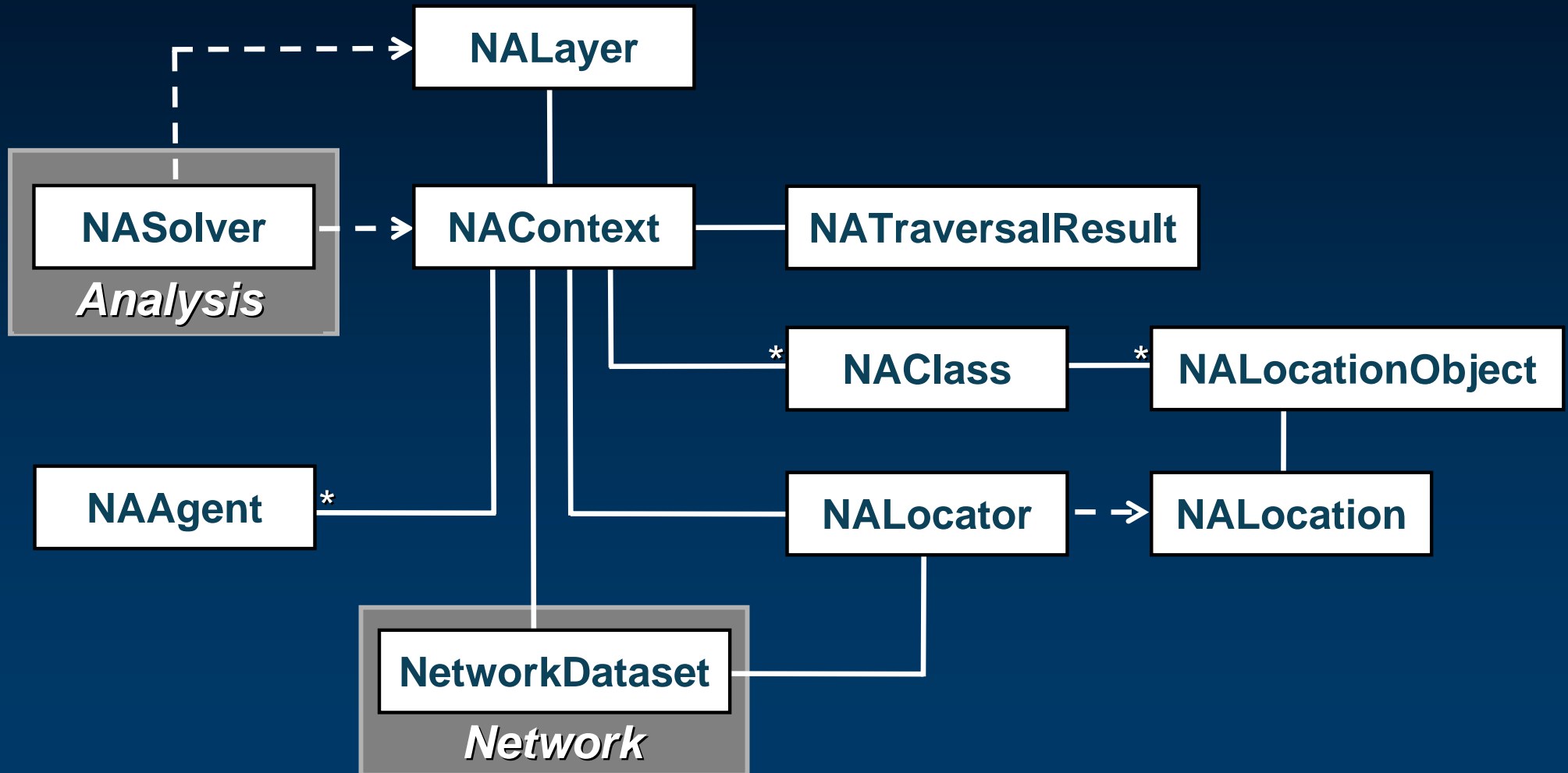

Accessing DENetworkDataset

- **What is a DENetworkDataset?**
 - Lightweight representation of the network dataset
- **Why do you need it?**
 - Used for accessing descriptive information about the network dataset

```
// Get IDatasetComponent from network dataset  
IDatasetComponent dsComponent = networkDataset;
```

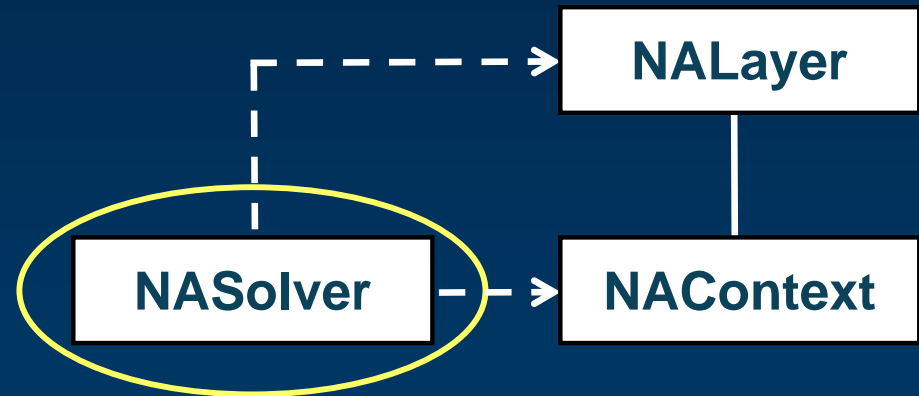
```
// Get data element as IDENetworkDataset  
IDENetworkDataset deNetworkDataset =  
    dsComponent.DataElement;
```

Overview of Network Analyst Objects



NASolver

- The NASolver is responsible for...
 - Managing the NAContext and NALayer
 - Performing the actual network analysis
- Used to set analysis properties
 - Impedance attribute
 - Restrictions
 - Hierarchy
 - Solver-specific properties



Create solver and modify properties (C#)

// Create new route solver

```
INARouteSolver naRouteSolver = new NARouteSolverClass();
```

// Change route-solver-specific settings using INARouteSolver

```
naRouteSolver.FindBestSequence = true;
```

```
naRouteSolver.UseTimeWindows = true;
```

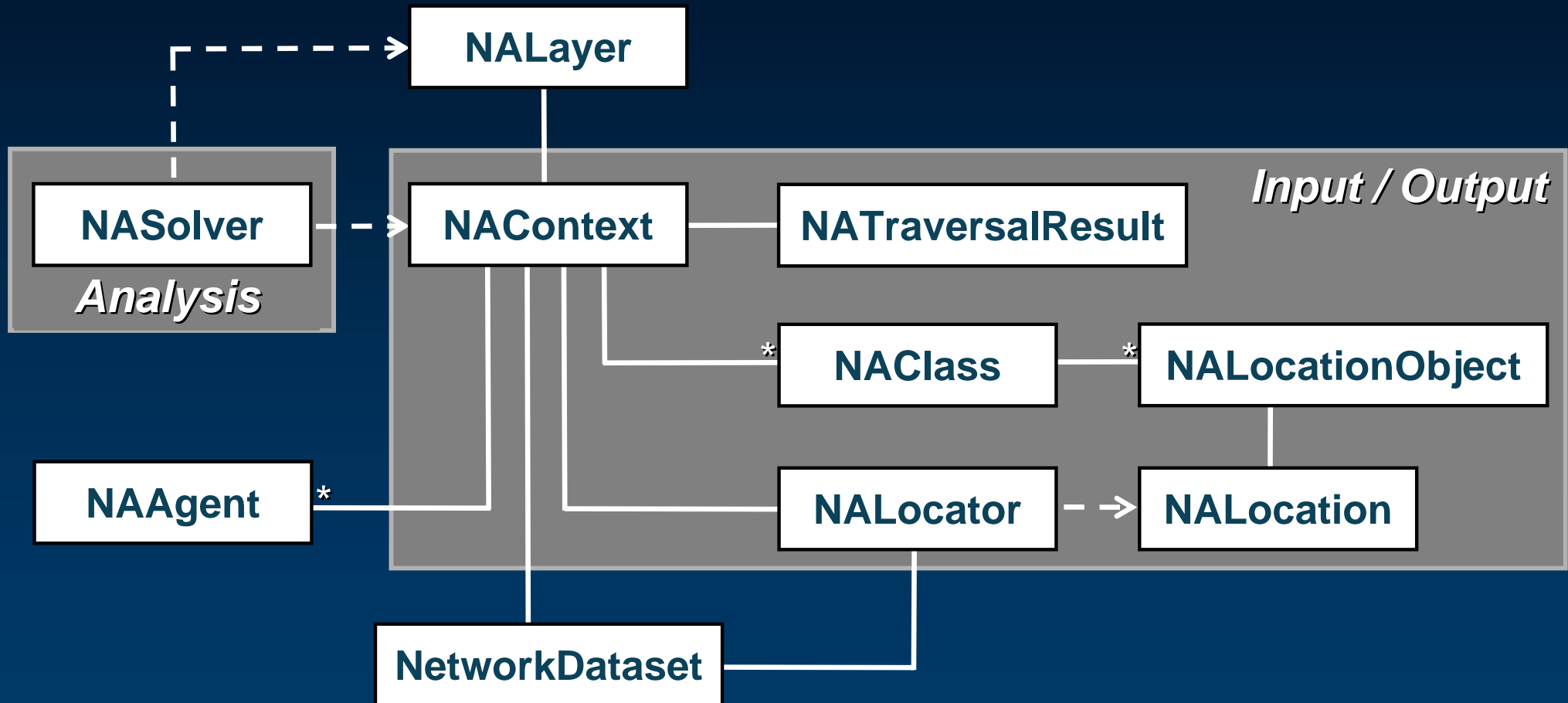
// Change general solver settings using INASolverSettings

```
INASolverSettings solverSettings = naRouteSolver;
```

```
solverSettings.ImpedanceAttributeName = "DriveTime";
```

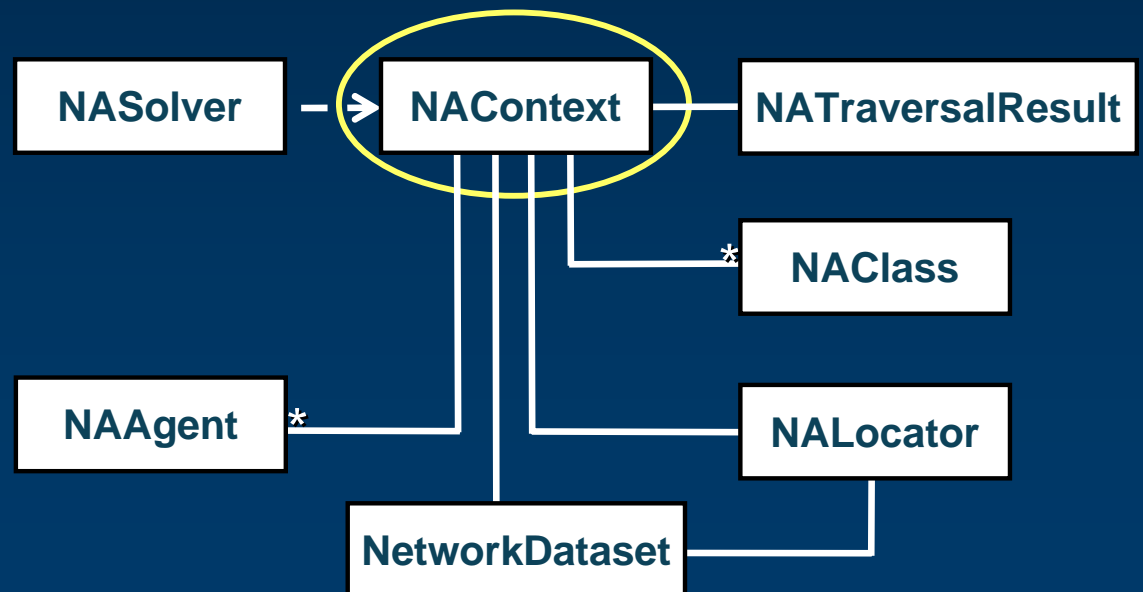
```
solverSettings.UseHierarchy = true;
```

Overview of Network Analyst Objects



NAContext

- Centrally manages the environment of a particular analysis
- The context holds references to the collection of objects associated with analysis
- This collection includes...
 - Network dataset
 - NAClasses
 - NASolver
 - NALocator
 - NATraversalResult
 - NAAgents



Creating/managing an NAContext (C#)

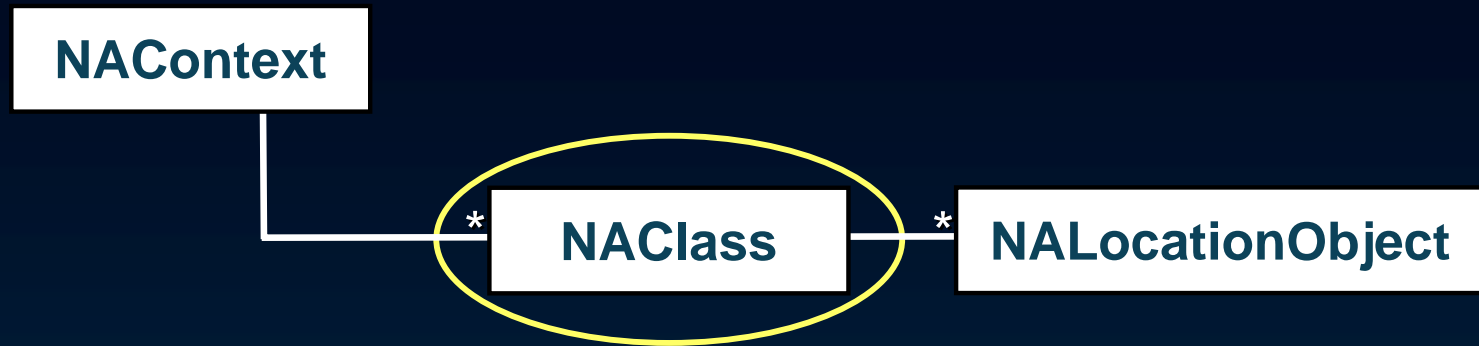
```
// Get the INASolver interface from the route solver
INASolver naSolver = naRouteSolver;

// Create and bind a context using the network dataset
INAContext naContext =
    naSolver.CreateContext(deNetworkDataset, naSolver.Name);
INAContextEdit naContextEdit = naContext;
naContextEdit.Bind(networkDataset, new GPMessagesClass());

// ...

// MUST update context if solver properties have been changed
naSolver.UpdateContext(naContext);
```

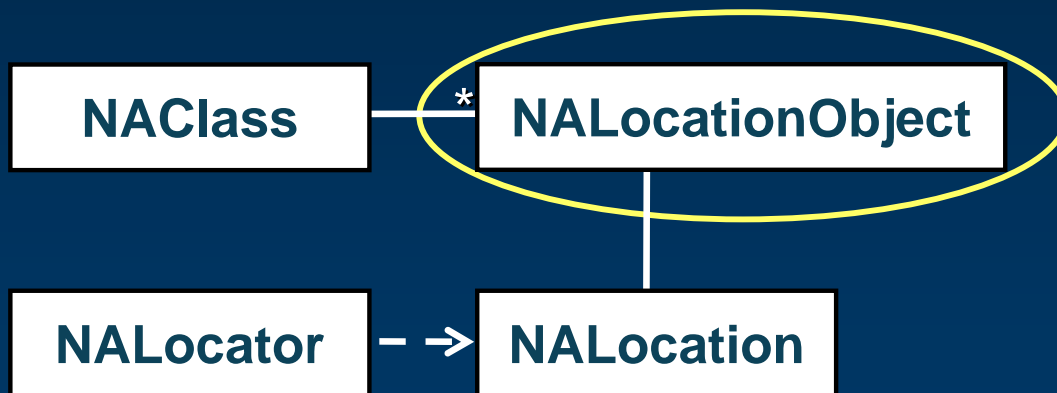
NAClass



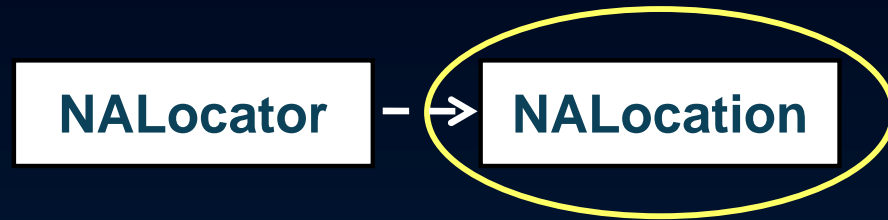
- **In-Memory Feature Class**
- **Holds input/output of a problem instance**
 - Network locations (e.g., stops)
 - Analysis results (e.g., routes)
 - Descriptive info (e.g., route name)
- **Can be searched, updated, etc.**
- **Persisted within NAContext**

NALocationObject / NALocationFeature

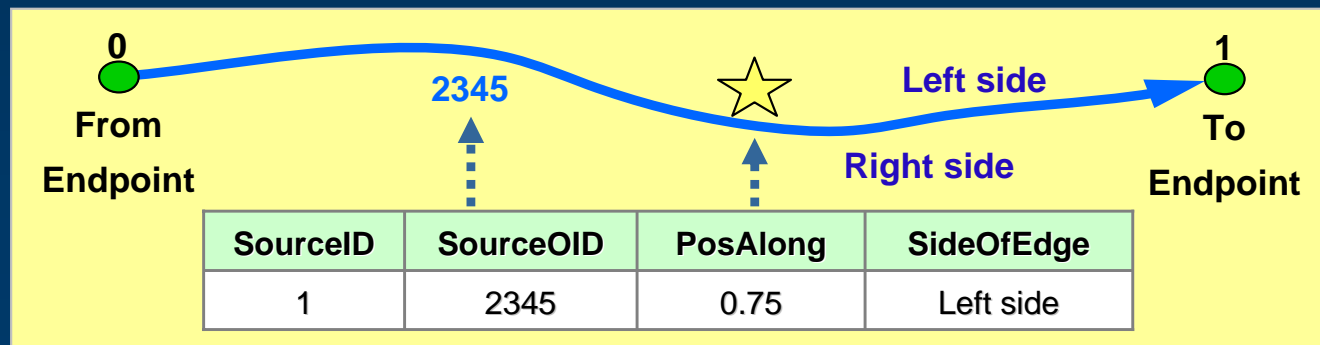
- Specialized feature to represent network location
- Each NALocationObject has an associated NALocation



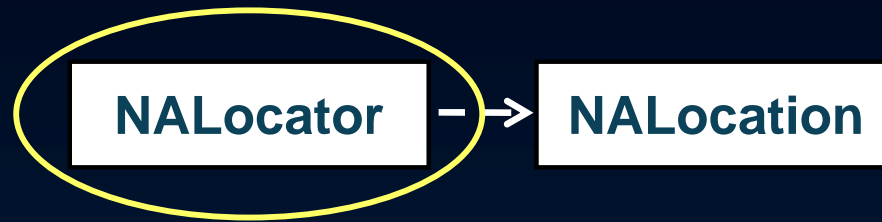
NALocation



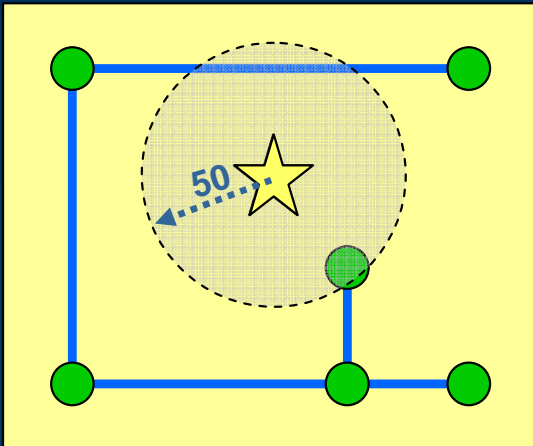
- Represents position within the network dataset
- Referenced by NALocationObject
- Four key properties
 - SourceID
 - SourceOID
 - SourcePosition (i.e., position along feature)
 - Side (i.e., topological side of feature)



NALocator



- Provides position discovery within the network dataset
 - Populates NALocation from known geometry
 - Populates geometry from known NALocation
- Manages locator agents
 - Feature locator agents – perform spatial searches
 - Field locator agents – map to existing location fields



Use Network Location Fields

Property	Field
PosAlong	EdgePosition
SideOfEdge	EdgeSide
SourceID	
SourceOID	

DelivSide
FromTime
ToTime
UnloadMinutes
BusinessType
SourceID
SourceOID
EdgePosition
EdgeSide

Find NALocation using NALocator (C#)

```
// Get the NALocator from the context
```

```
INALocator naLocator = naContext.Locator;
```

```
// Get/Create a point for loading
```

```
IPoint inputPoint = new PointClass();
```

```
inputPoint.X = -117.195533;
```

```
inputPoint.Y = 34.057058;
```

```
// Create a new NALocation to populate from the NALocator
```

```
INALocation naLocation = new NALocationClass();
```

```
IPoint outputPoint = new PointClass();
```

```
double offsetDistance = 0;
```

```
// Query the NALocator to locate the input point along the network
```

```
naLocator.QueryLocationByPoint(inputPoint, ref naLocation,  
ref outputPoint, ref offsetDistance);
```

Create new NALocationObject (C#)

```
// Get the appropriate NAClass from the context
IFeatureClass stopsFC = naContext.NAClasses.get_ItemByName("Stops");

// Create a new feature in the Stops feature class
IFeature stopFeature = stopsFC.CreateFeature();

// Initialize default feature values
IRowSubtypes rowSubtypes = stopFeature;
rowSubtypes.InitDefaultValues();

// Set the shape of the stop feature
stopFeature.Shape = inputPoint;

// Cast to INALocationObject, set its NALocation, and store feature
INALocationObject naLocationObject = stopFeature;
naLocationObject.NALocation = naLocation;
stopFeature.Store();
```

Batch locate/load features into NAClass (C#)

// Create a batch locator/loader helper object

```
INAClassLoader2 naClassLoader = new NAClassLoaderClass();
```

// Setup the loader to load into the "Stops" NAClass

```
naClassLoader.Initialize(naContext, "Stops", featureCursor);
```

// Load input features

```
int rowsIn = 0, rowsLoaded = 0;
```

```
naClassLoader.Load(featureCursor, null, ref rowsIn, ref rowsLoaded);
```

NATraversalResult

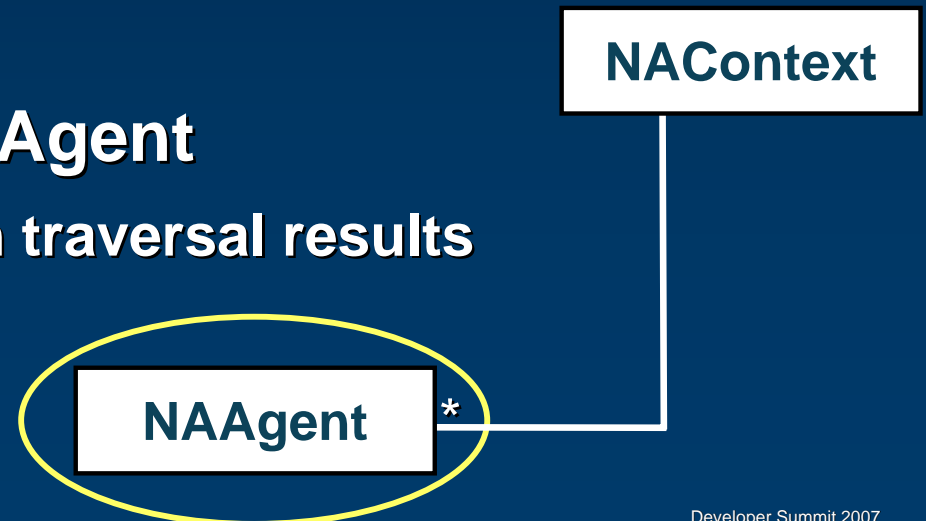
- Intermediate network traversal data generated during solve
- Contains three feature classes
 - Edges
 - Junctions
 - Turns



- Result features reference source features in NAClass feature classes and network dataset source feature classes
- Used to generate output geometries, directions, and other ancillary data
- Not persisted within map documents (.mxd) or layer files (.lyr)
- See sample: “NATraversalResult”

NAAgent

- **Supplementary object attached to the context**
 - Persisted with NAContext
- **Used for various post-processing tasks on the context and/or its result**
- **Notified when the context or its result have been updated**
- **Example: NAStreetDirectionsAgent**
 - Used to generate directions from traversal results



Solve and generate directions (C#)

// Solve the route

```
IGPMessages gpMessages = new GPMessagesClass();  
bool isPartialSolution = naSolver.Solve(naContext, gpMessages, null);
```

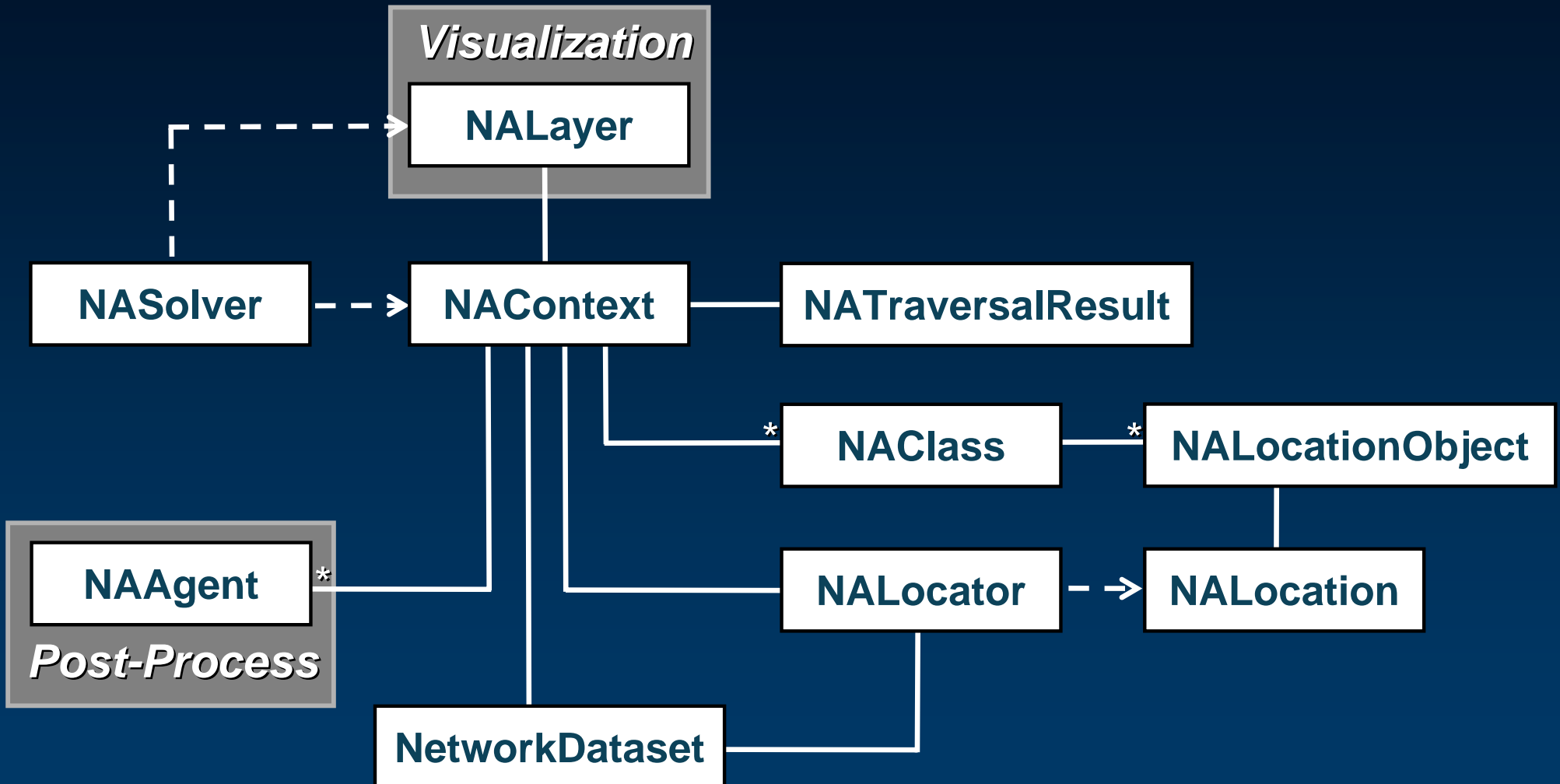
// Get the NAStreetDirectionsAgent from the context

```
INASStreetDirectionsAgent directionsAgent =  
    naContext.Agents.get_ItemByName("StreetDirectionsAgent");
```

// Generate directions and save to XML file

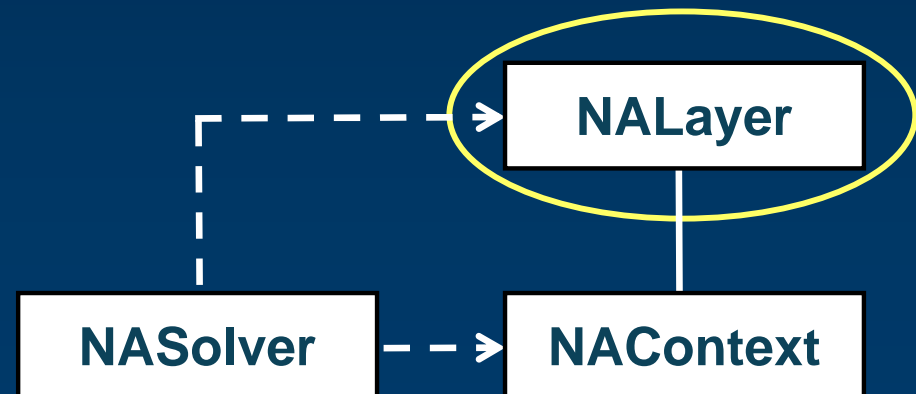
```
directionsAgent.LengthUnits = esriNetworkAttributeUnits.esriNAUMeters;  
directionsAgent.TimeAttributeName = "TravelTime";  
directionsAgent.Execute(null, null);  
directionsAgent.DirectionsContainer.SaveAsXML("C:\Temp\Directions.xml");
```

Overview of Network Analyst Objects



NALayer

- Composite map layer
- Holds a reference to the NAContext
- Exposes the NAClass feature classes within the NAContext as feature layers
- Saved in map documents (.mxd) or layer files (.lyr)
- Used by ArcMap, ArcGIS Server, and Geoprocessing
- See sample: “Route layer”



Create a network analysis layer (C#)

```
// Create a network analysis layer for the context  
INALayer naLayer = naSolver.CreateLayer(naContext);
```

```
// Add the layer to a map  
IMap map = GetMap();  
map.AddLayer(naLayer);
```

```
// ...
```

Access a network analysis layer (C#)

- ArcMap

```
// In ArcMap, access the Network Analyst extension
```

```
INetworkAnalystExtension naExtension =  
    application.FindExtensionByName("Network Analyst");
```

```
// Access the current network analysis layer from the NAWindow
```

```
INALayer naLayer = naExtension.NAWindow.ActiveAnalysis;
```

- Engine

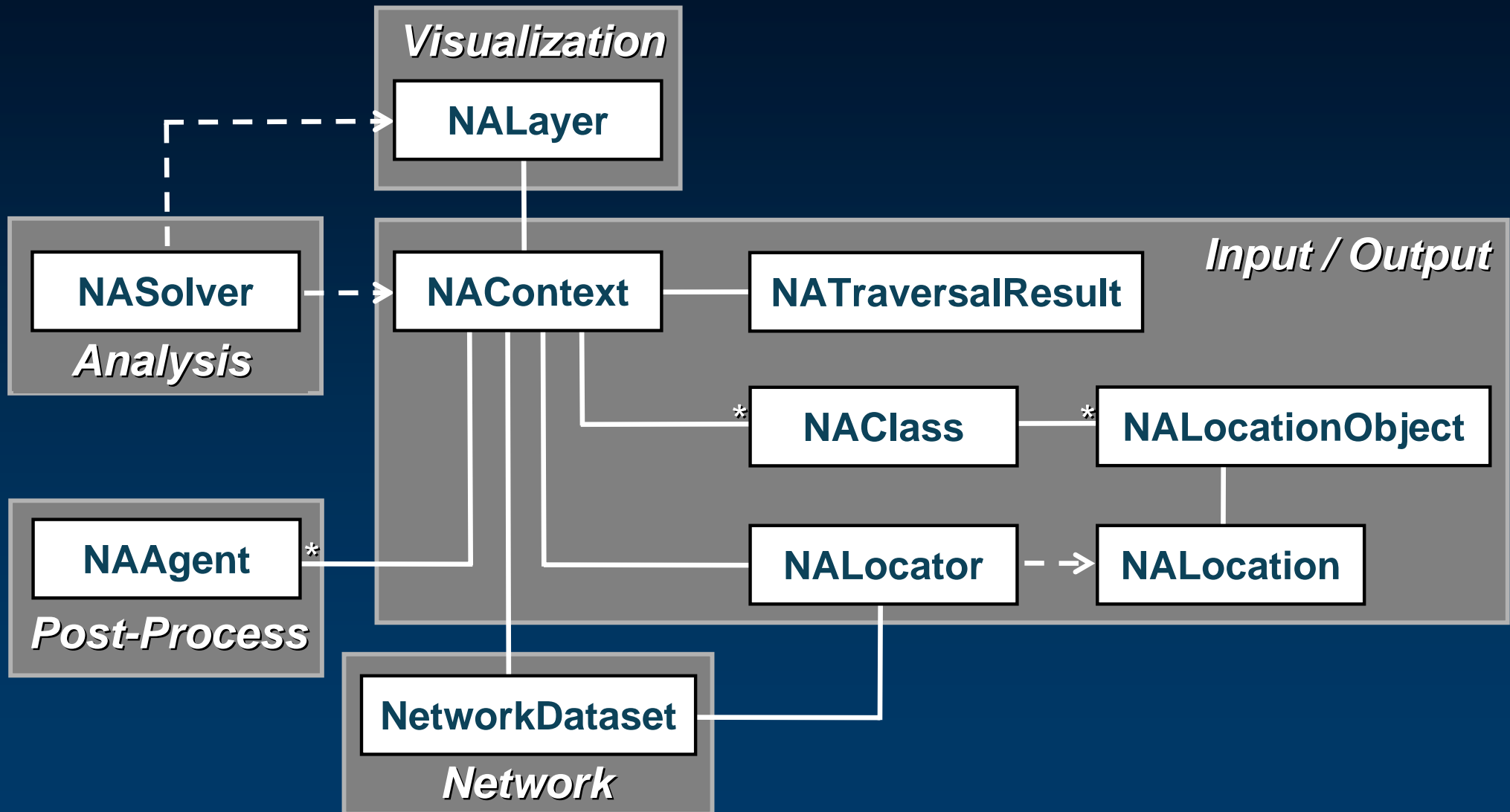
```
// In Engine, create the Network Analyst environment
```

```
IEngineNetworkAnalystEnvironment naEnvironment = new  
    EngineNetworkAnalystEnvironmentClass();
```

```
// Access the current network analysis layer from the NAWindow
```

```
INALayer naLayer = naEnvironment.NAWindow.ActiveAnalysis;
```

Overview of Network Analyst Objects



Agenda

- Introduction
- Engine Controls
- Network Analyst API
- **Network Analyst Customization**
- Support & Resources
- Questions

Demo

Relationship between Solvers and Evaluators

NASolver



`INetworkElement.get_AttributeValue(...)`

NetworkDataset

Edge element table

Feature class	Feature ID	Element ID	From-To Direction			To-From Direction		
			Minutes	Miles	DynMin	Minutes	Miles	DynMin
Streets	4485	1	1.0	1.7	?	1.2	1.4	?
Streets	4519	2	0.8	2.3	?	0.7	2.2	?
Streets	4520	3	2.0	1.6	?	2.0	1.4	?
Streets	5321	4	1.2	4.0	?	1.8	4.0	?
Streets	5356	5	2.2	1.8	?	2.6	2.8	?
Streets	6889	6	6.0	3.7	?	7.0	3.4	?
Streets	6890	7	10.0	1.0	?	9.6	1.3	?
Streets	7277	8	1.5	7.2	?	1.5	5.2	?
Streets	7296	9	8.2	4.0	?	8.6	4.3	?
Streets	10456	10	1.7	3.2	?	1.5	3.5	?



`Evaluator.QueryValue()`

Creating a Custom Evaluator – Why?

- **For network attribute values that are not stored/cached**
 - Real-Time Traffic speeds
 - Selection Restrictions
 - Slowdown Polygons
- **Network attribute values based on custom logic**
 - Height/Weight/Length restrictions by vehicle type
 - Global Turns
- **Order of magnitude faster than the VBScript evaluator**

Creating a Custom Evaluator – How?

- **Create two COM objects**
 - One evaluator object that implements **INetworkEvaluator2** and **INetworkEvaluatorSetup**
 - Used to evaluate and return the attribute value for network elements
 - Another editor object that implements **IEvaluatorEditor**
 - Used to setup the appropriate evaluator for an attribute in **ArcCatalog**
- **For recognition in ArcCatalog**
 - Register the editor object in “**ESRI Network Evaluator Editors**”
- **See samples: “Selection Restriction Evaluator” and “Global Turn Evaluator”**

Custom Evaluator – Key Concepts

- **Cached vs. Dynamic Attribute Values**
 - Built into network or evaluated at solve time
 - Specified by return value of `INetworkEvaluator.CacheAttribute`
- **Initialize vs. Refresh**
 - Place one-time-only initialization code in the `INetworkEvaluatorSetup.Initialize` method
 - Place per-solve internal state updates in the `INetworkEvaluator2.Refresh` method
- **QueryValue**
 - Place logic to assess the attribute value for a particular network element in `INetworkEvaluatorSetup.QueryValue`

Custom Evaluator Considerations

- To open a network dataset that uses a custom evaluator, a client machine must have the custom evaluator DLL installed
- **INetworkEvaluatorSetup.QueryValue()** needs to be fast and efficient!
 - Called for **every** element searched in the network
 - Affects the speed for solving an analysis
 - Move setup and update procedures to the **INetworkEvaluatorSetup.Initialize** and **INetworkEvaluator2.Refresh** methods, respectively

Elements Queried During Analysis



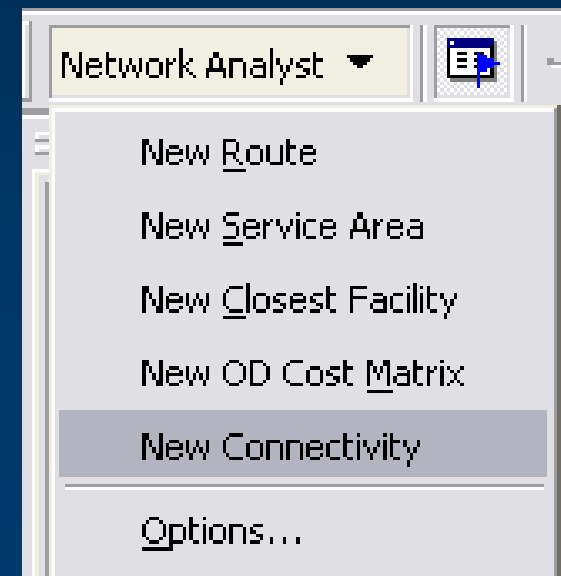
Demo

Creating a Custom Solver – Why?

- **Need custom logic that differs slightly from existing solvers**
 - **A* Route Solver**
- **Have analysis needs that aren't met by existing solvers**
 - **Connectivity Solver**
- **Have pre-existing solutions to embed into the ArcGIS Network Analyst framework**

Creating a Custom Solver – How?

- **Create a COM object for the solver**
 - Must implement **INASolver**, **INASolverSettings**, and **IPersistStream**
 - Used to perform the actual analysis, manage context, etc.
- **For recognition in ArcMap or Engine Controls**
 - Register the solver object in “**ESRI Network Analyst Solvers**” component category
- **See sample: “Custom Solver”**



Custom Solver – Key Concepts

- **NAContext creation and management**
- **NALayer creation and management**
- **Solving the analysis**
- **Writing the output**

NAContext Management

- **Allows solver to specify the input and output NAclass feature classes and fields required**
- **Implement INASolver.CreateContext**
 - Create NAclassDefinition for each NAclass
 - Pass NAclassDefinition to INAContextEdit.CreateAnalysisClass
- **Implement INASolver.UpdateContext**
 - Add and/or remove dynamic fields for input/output NAclasses

NALayer Management

- **Allows solver to specify the layers and symbology required**
- **Implement INASolver.CreateLayer**
 - Create NALayer composite layer
 - Create individual feature layers for each NAClass
 - Add feature layers to NALayer composite layer
- **Implement INASolver.UpdateLayer**
 - Update symbology based on solver results

Solving

- **Perform the network analysis using the NAClass feature classes and the network dataset**
- **Implement INASolver.Solve**
 - Consume the NetworkForwardStar
 - Implement your analysis algorithm
- **Using NetworkForwardStar**
 - Setup traversal properties
 - Add restrictions
 - Load barriers
 - Use QueryAdjacencies

Write Output

- **Write the results to a `NATraversalResult` object and/or output `NAClass` feature classes to be viewed/queried**
- **Tail-end of `INASolver.Solve` implementation**
- **Use insert cursors for fast feature creation**

Custom Solver Considerations

- **Framework implementation can be quite involved**
 - **Must write code to create NAContext, NAClassDefinitions, NAClasses, NALayer, and symbology**
 - **Must write logic to manage low-level network traversal/search**
- **No ArcMap UI support for custom solvers at 9.1**
- **Full ArcMap / Engine UI support for custom solvers at 9.2**
- **C++ code typically executes 2+ times faster than .NET code**

Additional Customization Possibilities

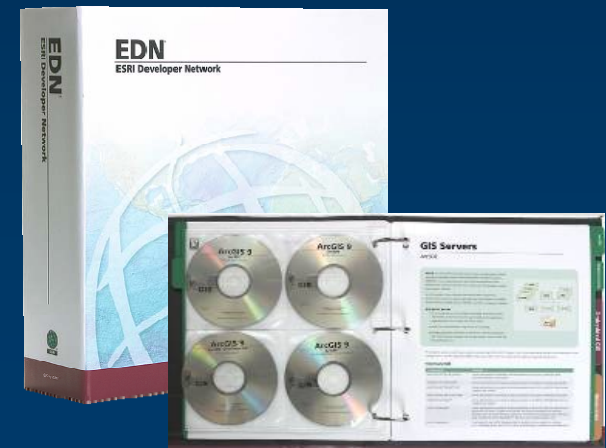
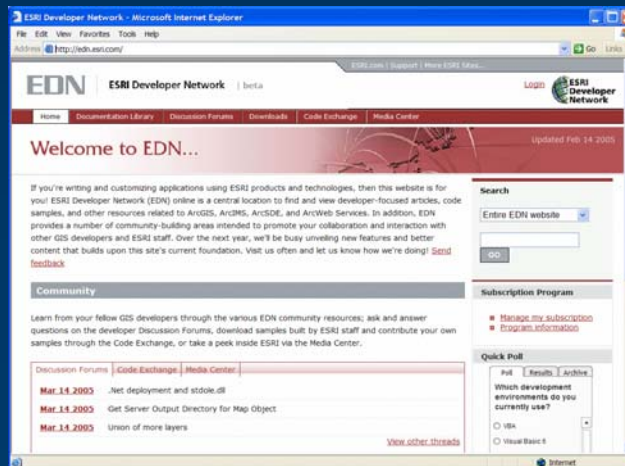
- **Custom NAAgent**
 - Used for custom post-processing on analysis results
- **Custom NALocatorAgent**
 - Used for custom NALocation population
- **Custom NASymbolizer**
 - Used for providing custom NALayer symbolization

Agenda

- Introduction
- Engine Controls
- Network Analyst API
- Network Analyst Customization
- **Support & Resources**
- Questions

ESRI Developer Network (EDN)

- ArcGIS Engine
 - Network Extension fully supported for developers
- ArcGIS Server
 - Network Extension fully supported for developers
- Developer Samples
- Technical Document
 - Programming ArcGIS Server Network Analyst Applications



ESRI Support Center

- Online portal to technical information
 - Knowledge Base
 - Technical articles
 - White papers
 - System requirements
 - Downloads
 - Patches and service packs
 - Data models
 - ArcScripts and samples
 - User forums
 - Discussion groups
 - E-mail lists



<http://support.esri.com>

For more information

- **Network Analyst Product Page**
 - <http://www.esri.com/software/arcgis/extensions/networkanalyst/index.html>
- **Free webcast - Introduction to Network Analyst**
 - http://campus.esri.com/acb2000/showdetl.cfm?DID=6&Product_ID=837
- **Free Podcasts**
 - http://www.esri.com/news/podcasts/instructional_series.html
- **Training - Working with ArcGIS Network Analyst**
 - <http://training.esri.com/ilt/schedule/index.cfm?fa=courseLink&courseID=D+50099071>
- **.NET Samples on EDN**
 - <http://edn.esri.com/index.cfm?fa=codeExch.querySamples92&activeTab=samples&passedLanguage=All&passedCategory=Networks&products=ArcView&languages=All&categories=Network+extension>

Tech Talk

- Informal Q&A
- In Tech Talk Area 4

Session Evaluations Reminder

Questions ?

... Thank you