



# Discovering Network Diagrams and their Configurations

Mohana Krishna Punnam

Anne-Yvonne Blin



2021 ESRI  
DEVELOPER SUMMIT

# Topics

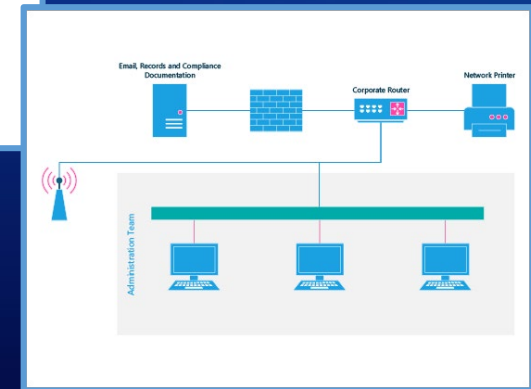
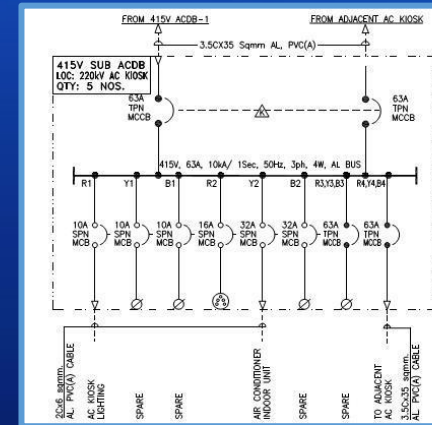
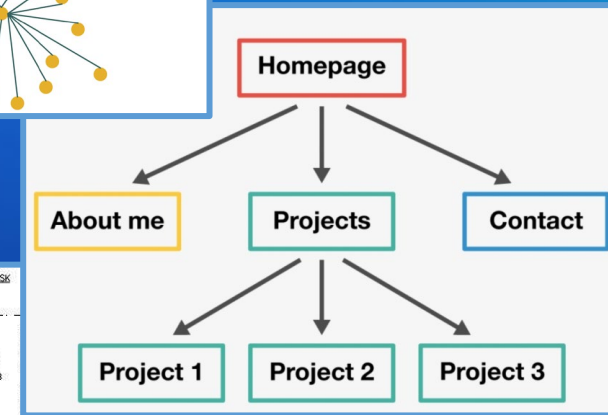
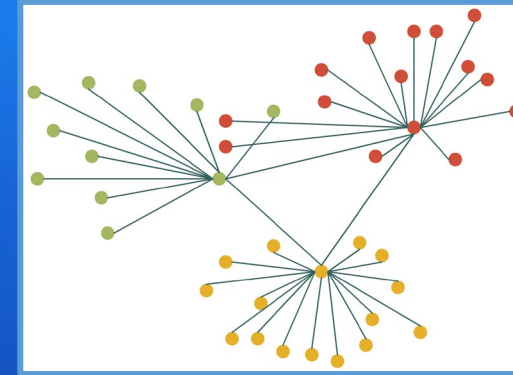
- **Introduction**
- **Diagram Features and Diagram Layer**
- **Diagram Rules, Layouts and Templates**
- **Diagram Template Configurations**
- **Network Diagrams APIs**
- **Network Diagrams Benefits**

# Introduction

The background features a vibrant, abstract composition. On the left side, there are several overlapping elements: a stylized globe, a molecular structure with red and blue spheres, and various flowing, ribbon-like shapes in shades of blue, red, and yellow. The overall aesthetic is modern and scientific, set against a gradient background that transitions from a deep blue on the left to a lighter cyan on the right.

# What is a diagram?

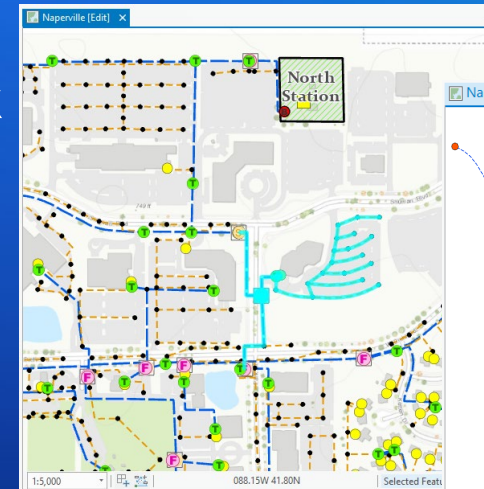
- A symbolic and simplified way to represent the components of a process, devices, or other objects.
- Representation of the data in different layouts
  - Grid Layout, Tree Form, etc.



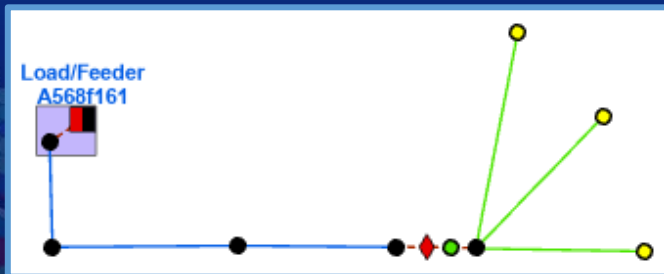
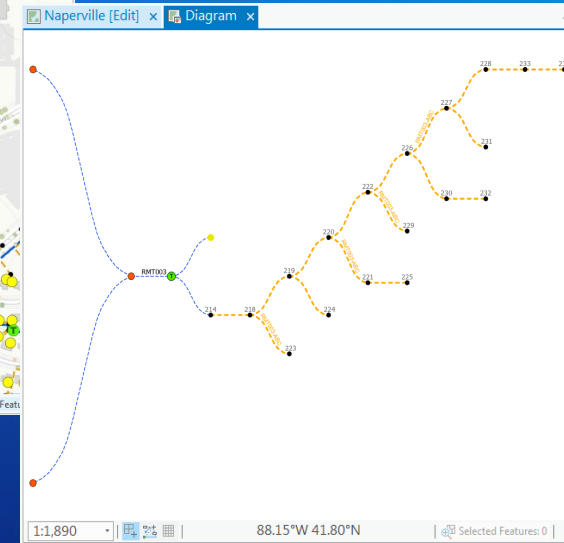
# About Network Diagrams

- Allows users to easily create simplified views of the network
- Available as part of the Utility Network and Trace Network
- Consumes the network topology

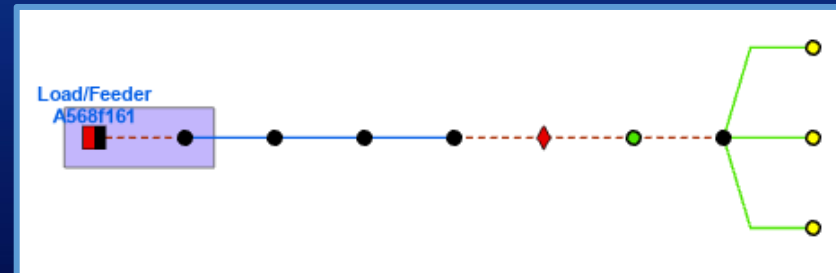
Network map



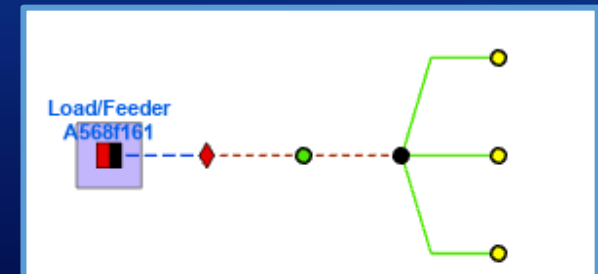
Schematic view



Geo-Schematic view



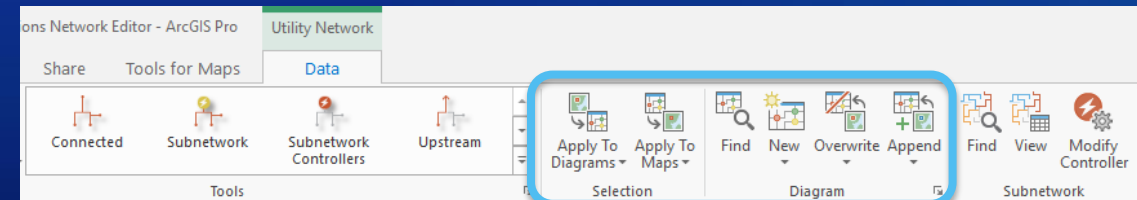
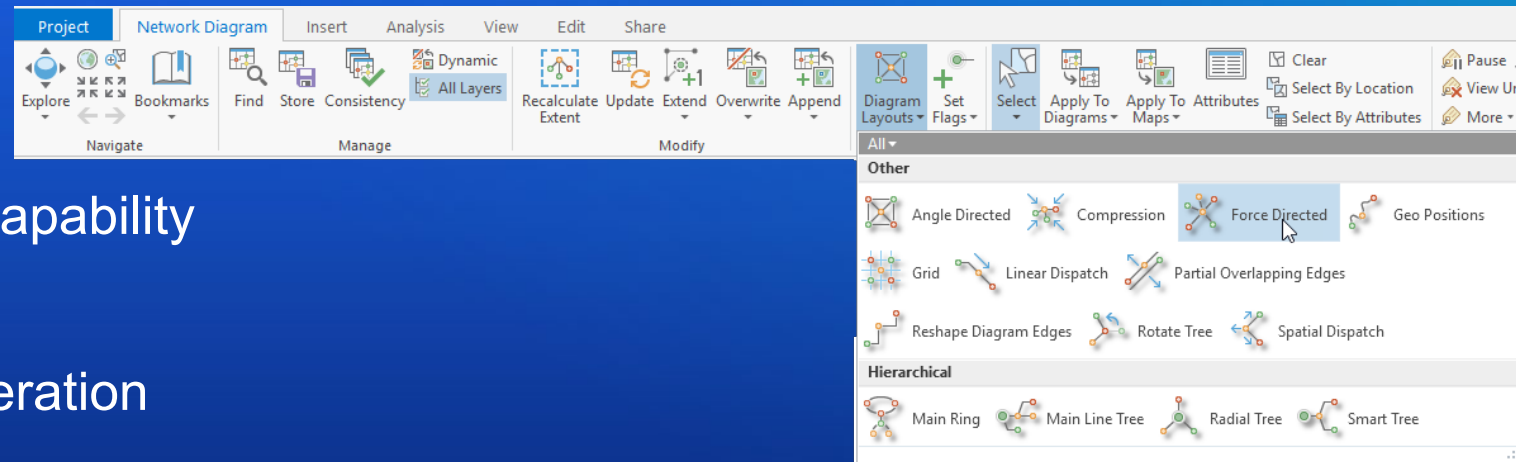
Tree view

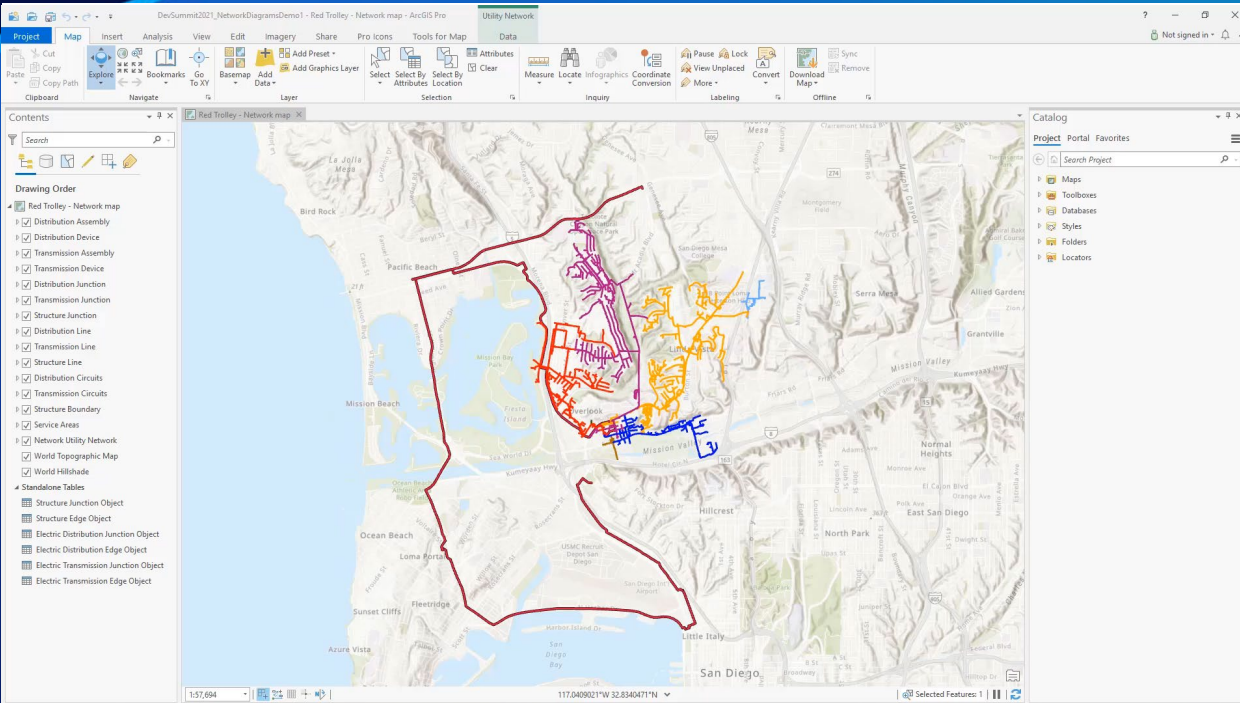


Simplified tree view

# Network Diagrams in Utility Networks and Trace Networks

- Use layout algorithms
- Allow manual editing
- Use layer appearance and labelling capability
- Stored in the database
- Created from traces executed at generation and update
- Updated based on the data changes while preserving layout





# Demo#1

# Network Diagrams

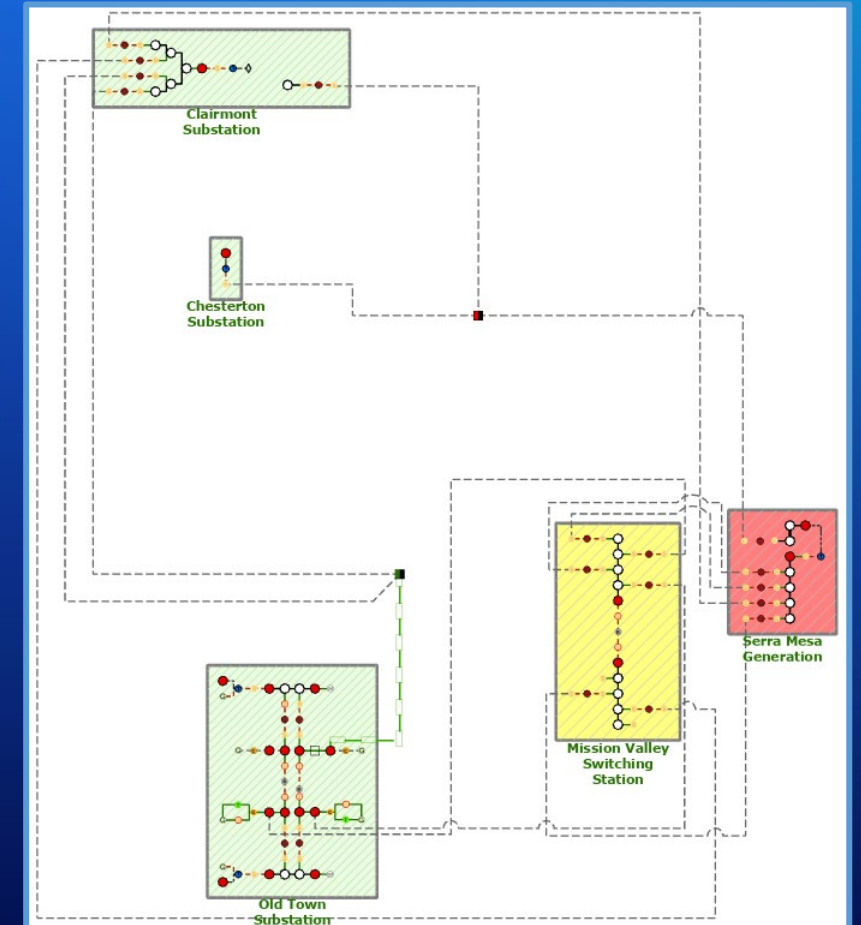
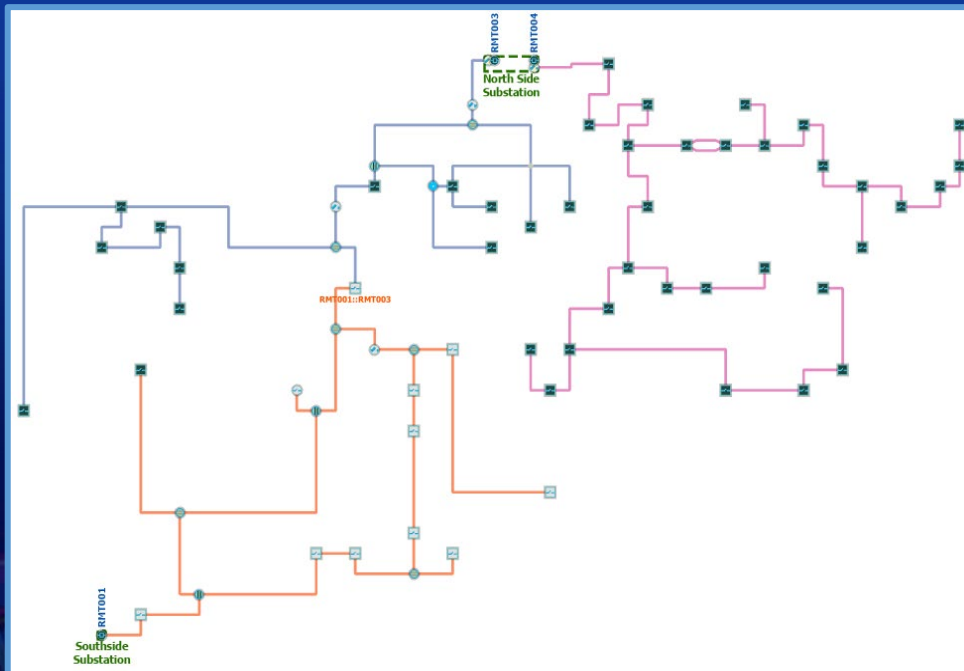
# basis

# Diagram Features and Diagram Layer

The background features a vibrant, abstract composition. On the left, there's a stylized globe with blue and red segments. Below it, a network diagram with blue nodes and red connecting lines is visible. The bottom half of the image is dominated by thick, flowing, overlapping bands of color in shades of blue, red, and yellow, creating a sense of dynamic movement and depth. The overall aesthetic is modern and technical.

# Diagram Features

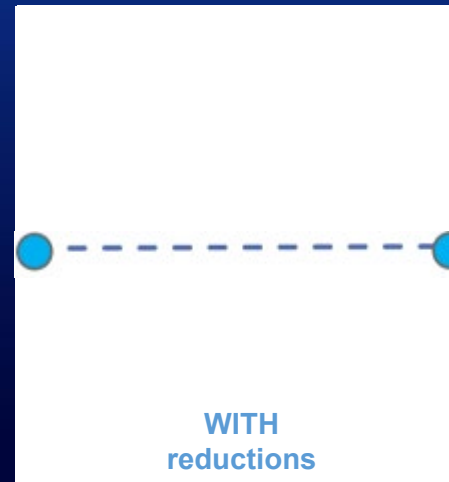
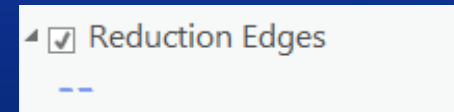
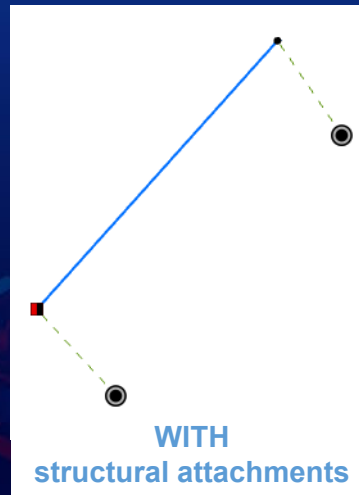
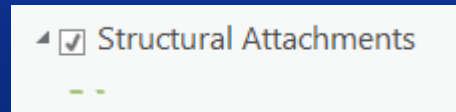
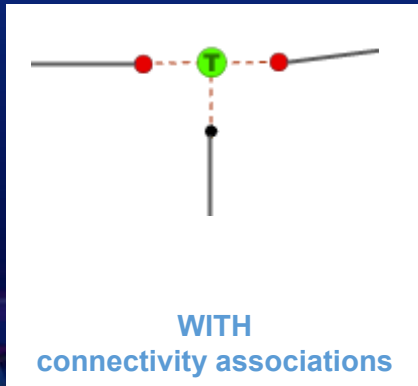
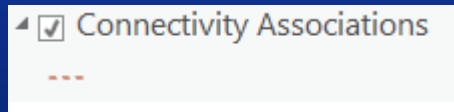
- Diagram junction depicted as symbol point geometry
- Diagram edge depicted as line geometry
- Diagram container depicted as polygon geometry





# Diagram Edges

- Network line features
- Edge objects
- Junction-junction connectivity and structural attachment associations
- Reduction edges that aggregate several network features/objects



# Diagram Containers

- Structure junctions, assemblies, devices
- Structure junction objects
- Polygon structure boundaries



Diagram Container Polygon  
TransformerBank  
Assembly

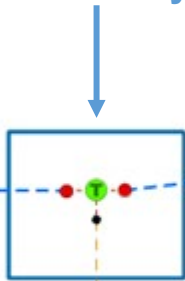
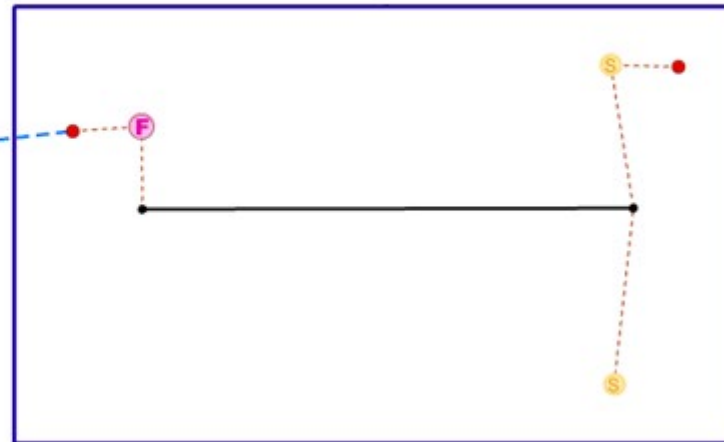



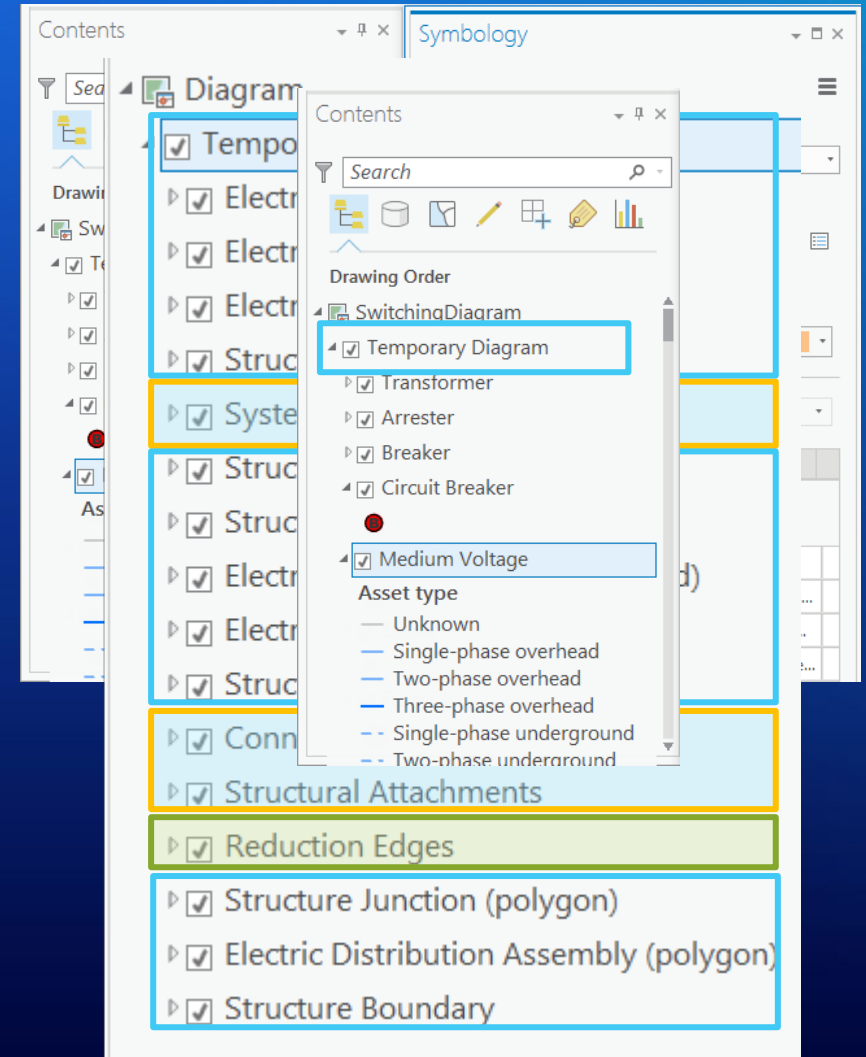


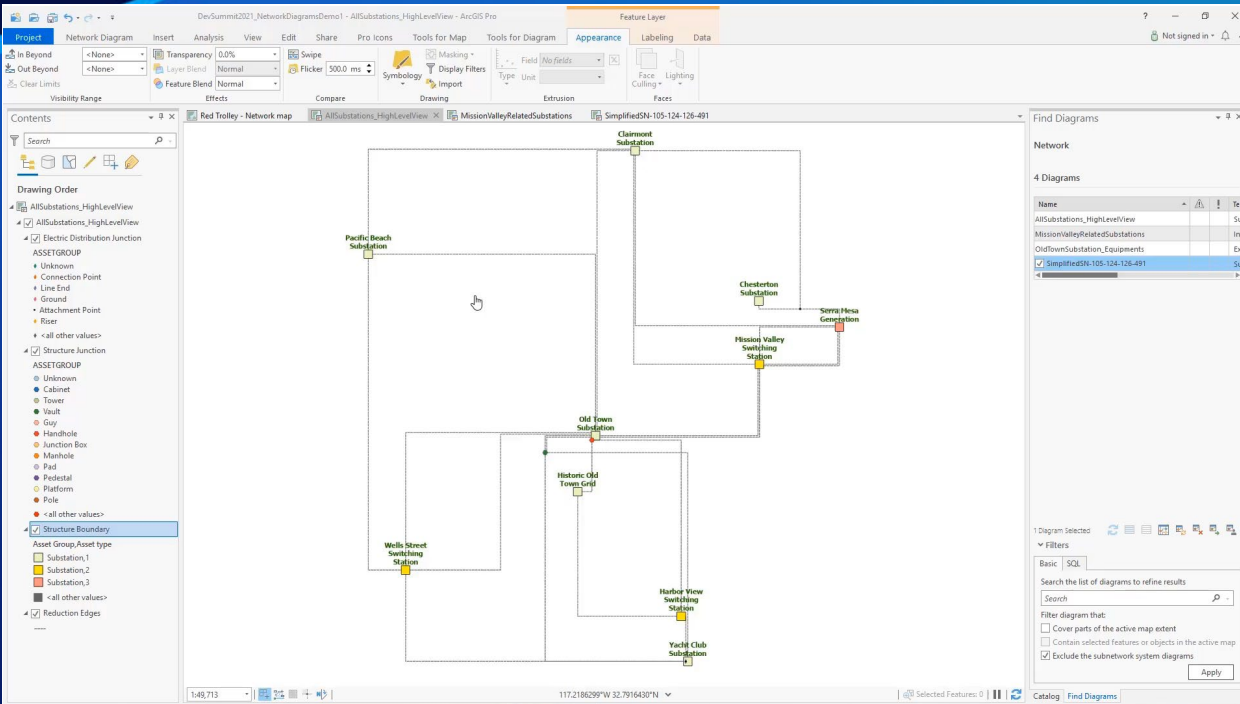
Diagram Container Polygon  
Switchgear Assembly



# Diagram Layer

- Composite layer whose sublayers are feature layers
- Each sublayer get benefits from core layer properties and labeling capability
- A diagram sublayer may reference:
  -  Diagram features that represent network features or network objects
  -  Diagram features that represent internal network elements. For example, connectivity associations, structural attachments, system junctions
  -  Pure diagram features (reduction edges)





# Demo#2

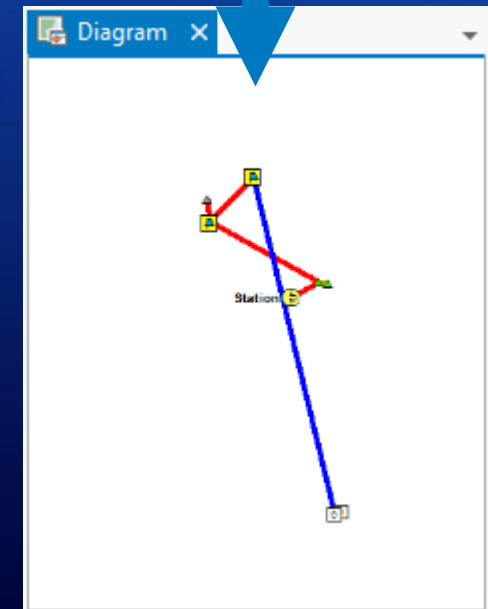
## Manual editing of diagram features

# Diagram Rules, Layouts and Templates

The background features a complex, abstract graphic design. It includes a globe in the lower right, a network diagram with nodes and lines in the lower center, and various flowing, ribbon-like shapes in shades of blue, red, and yellow. The overall aesthetic is modern and technical, suggesting a focus on data, communication, or design.

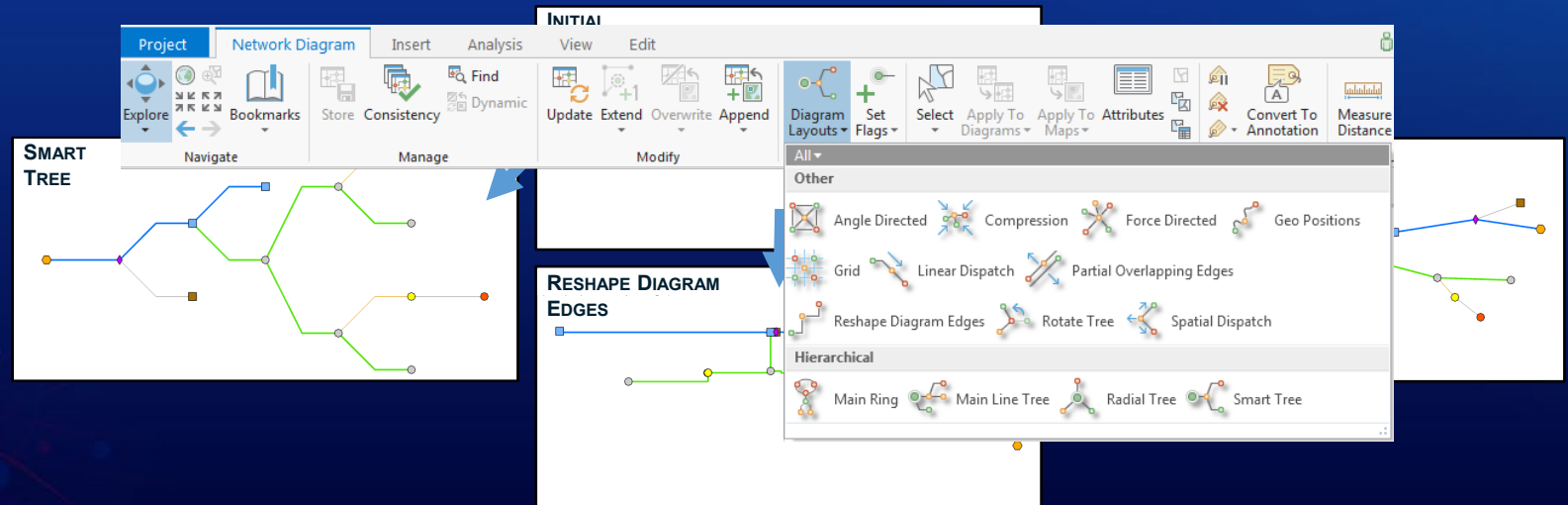
## Diagram Rules

- Executed during network diagram generation
- Start from input network feature/object sets
- Used to progressively build the content of network diagrams
- Configured to modify the diagram contents by
  - Discarding certain network features or objects,
  - Adding extra network features or objects,
  - Simplifying diagram content by aggregating network features or objects
- Re executed at each diagram update



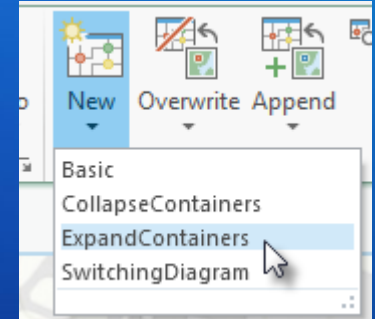
# Diagram Layouts

- Used to clarify and normalize spacing between diagram features
- Run on demand on all or parts of any open network diagram
- Configured to automatically execute at diagram generation and update
  
- Three categories of layouts
  - Tree layouts
  - Refinement layouts
  - Schematic layouts



# Diagram Templates

- **Any network diagram is based on a diagram template**
  - **Select diagram template to create a new diagram**
- **A diagram template contains**
  - **Configuration properties defining the network diagram contents**  
=> diagram rule and layout definitions
  - **Layer properties defining the network diagram presentation**  
=> diagram layer definition



# Diagram Generation Process

## 1. Elementary Build phase

- A diagram feature for each input network feature or object
- Any diagram edge with its extremity junctions
- Any content with its container

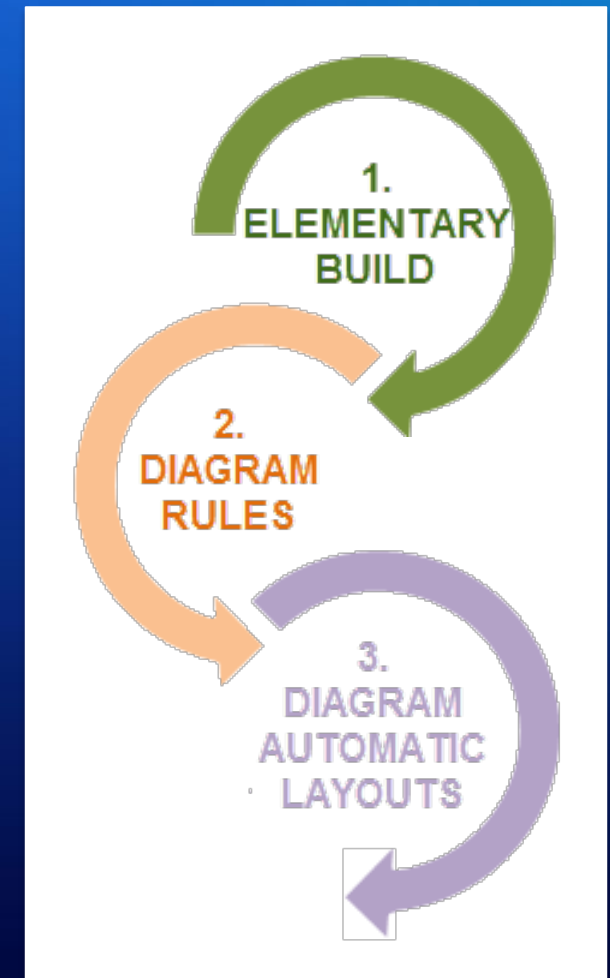
## 2. Diagram Rules phase [OPTIONAL]

executed each-in-turn to build and refine the diagram graph

## 3. Diagram Automatic Layouts phase [OPTIONAL]

chained each-in-turn to lay out the diagram content

**NOTE:** This process is also quite the same at diagram update



# Diagram Generation Process

- Diagrams generation process sample

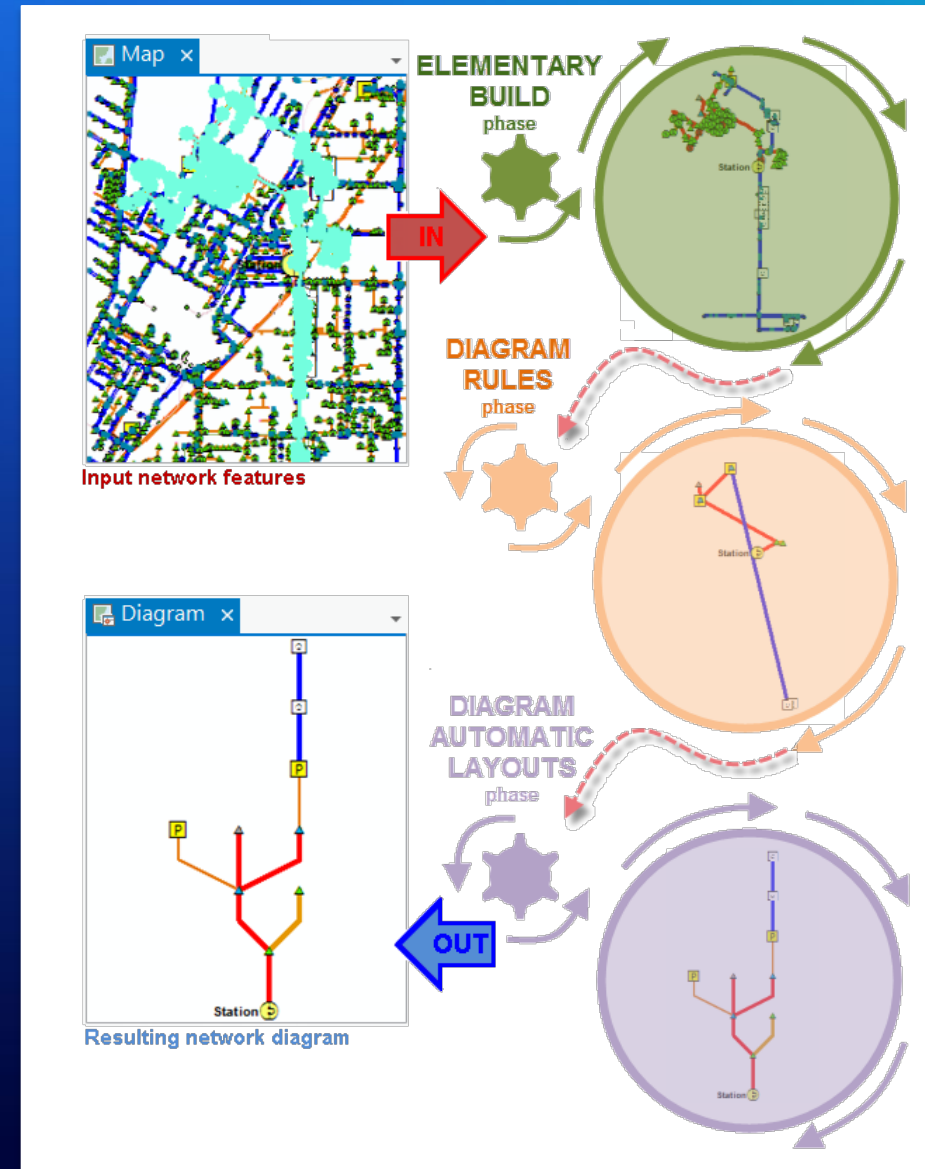
1. Elementary Build phase

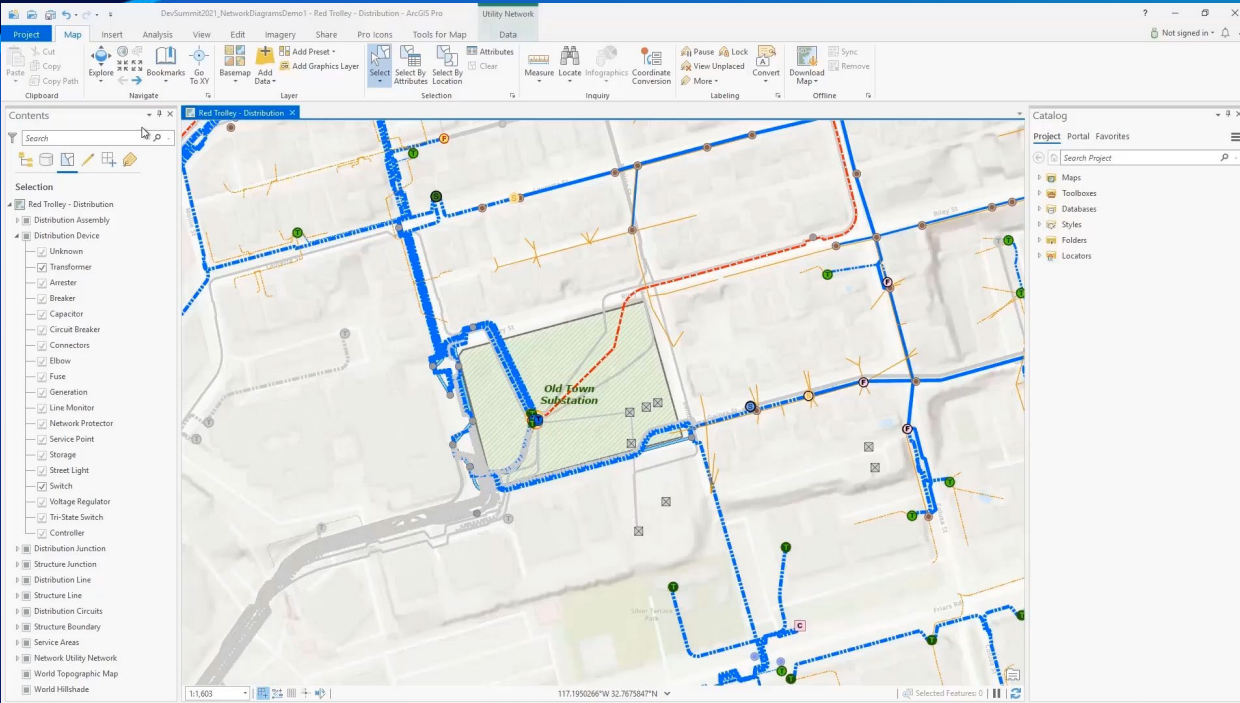
2. Diagram Rules phase

To reduce most of the non-critical intervening elements

3. Diagram Automatic Layouts phase

To execute a Smart Tree layout





# Demo#3

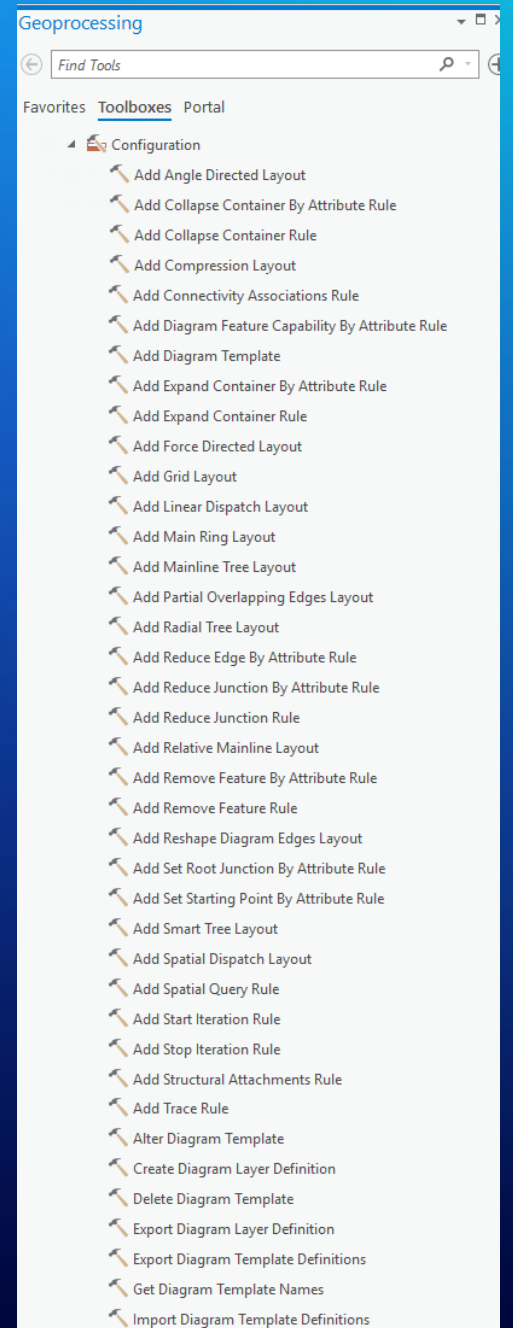
## Network diagram sample generations

# Diagram Template Configurations

The background features a vibrant, abstract composition. On the left side, there are several overlapping elements: a stylized globe with a grid pattern, a network diagram with nodes and connecting lines, and various flowing, ribbon-like shapes in shades of blue, red, and yellow. The overall aesthetic is modern and technical, suggesting themes of data, technology, or global connectivity. The text is positioned in the upper left quadrant, set against a solid blue background that transitions into the abstract graphics.

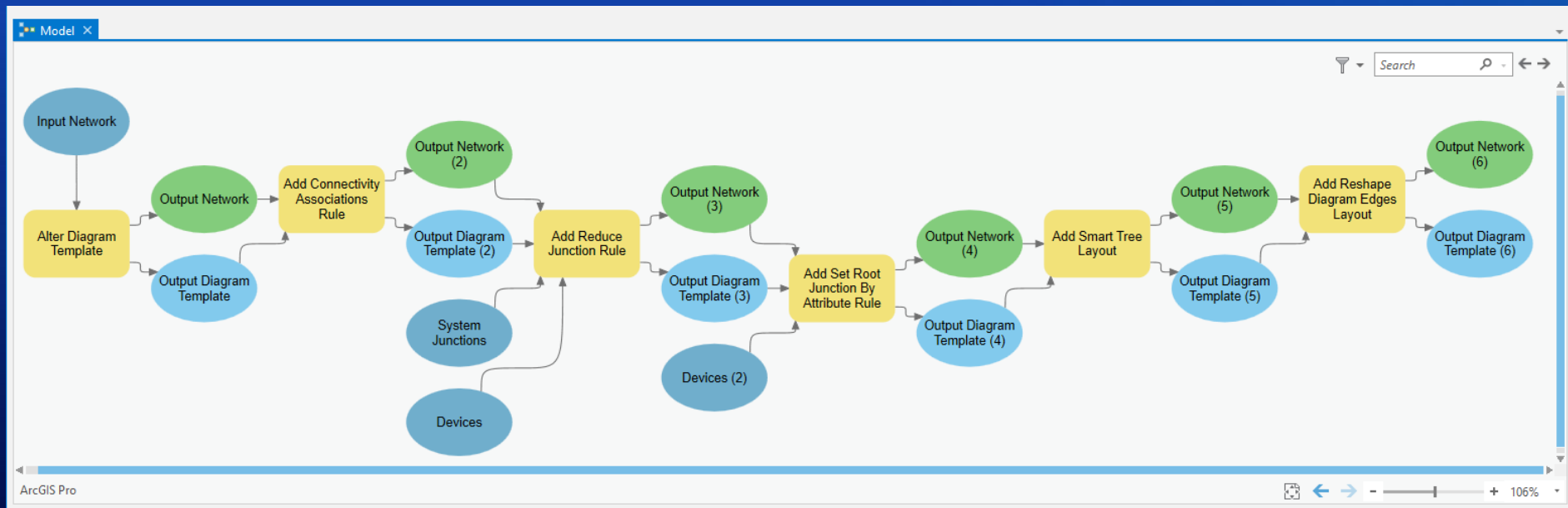
# Diagram Template Configuration Tools

- Exposed as a collection of geoprocessing tools
- Administrative tools
- Used to configure how the content of network diagrams will be
  - Built => diagram rule and layout definitions
  - Displayed => diagram layer definition
- Executed from either
  - A network in a file geodatabase
  - A network in an enterprise geodatabase



# Diagram Rule and Layout Definition

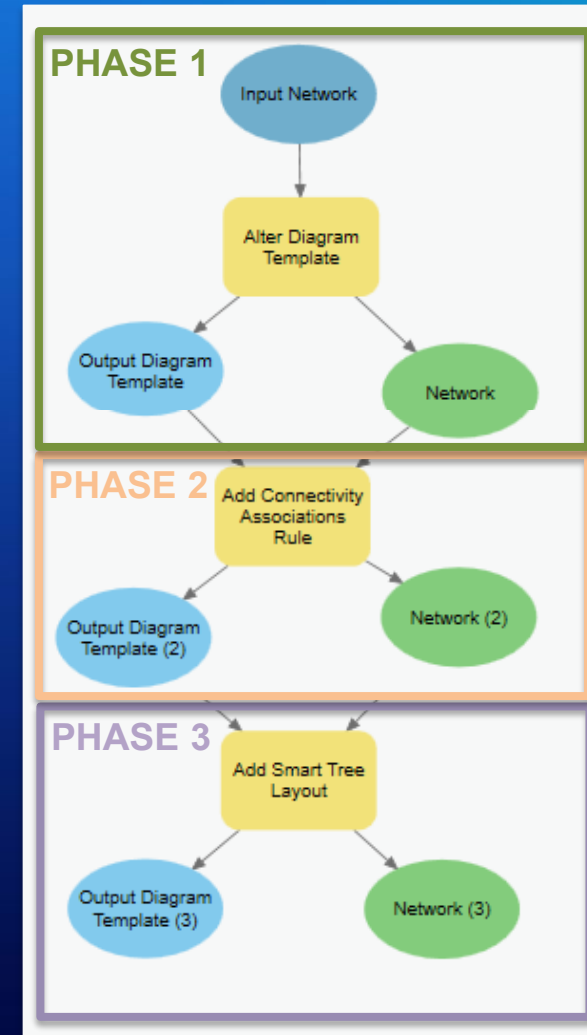
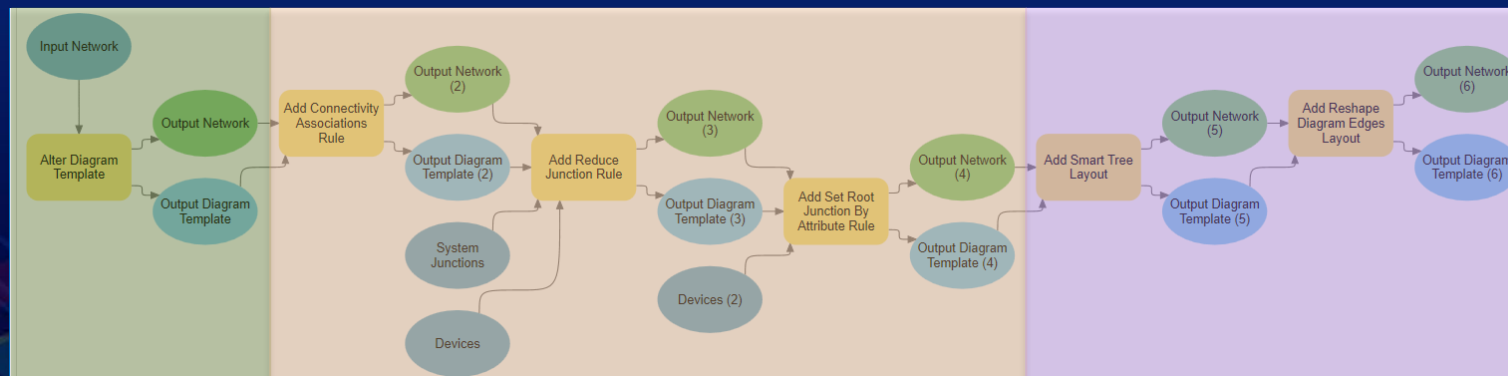
- Diagram rule and layout configuration tools easy to chain using
  - Python scripts
  - Geoprocessing models



- Model Builder or Python scripting recommended vs running tools one-by-one in the Geoprocessing pane

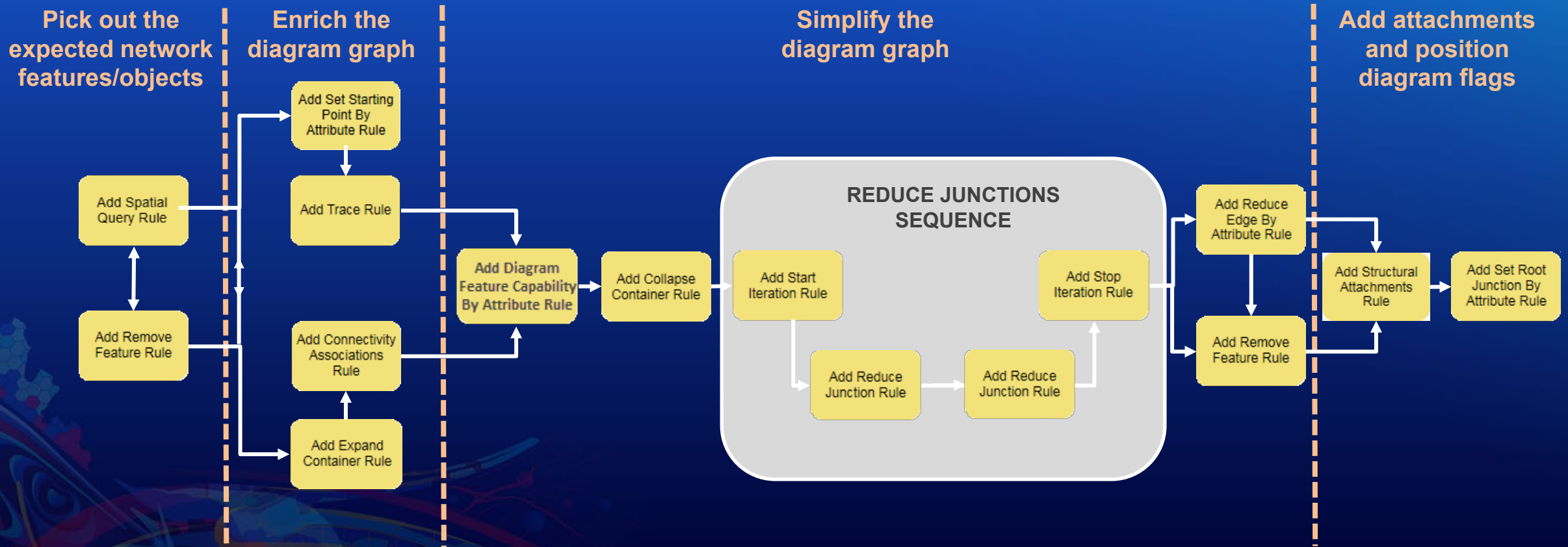
# Diagram Rule and Layout Definition

- Advantages of using geoprocessing models
  - Chaining of geoprocessing tools organized in three main phases like during network diagram generation process
    - **PHASE 1 – ELEMENTARY BUILD** (using the Alter Diagram Template tool)
    - **PHASE 2 – RULE SEQUENCE** (using the Add <xxx> Rule tools)
    - **PHASE 3 – LAYOUT SEQUENCE** (using the Add <xxx> Layout tools)
  - Help to ensure the position of each diagram rule and layout



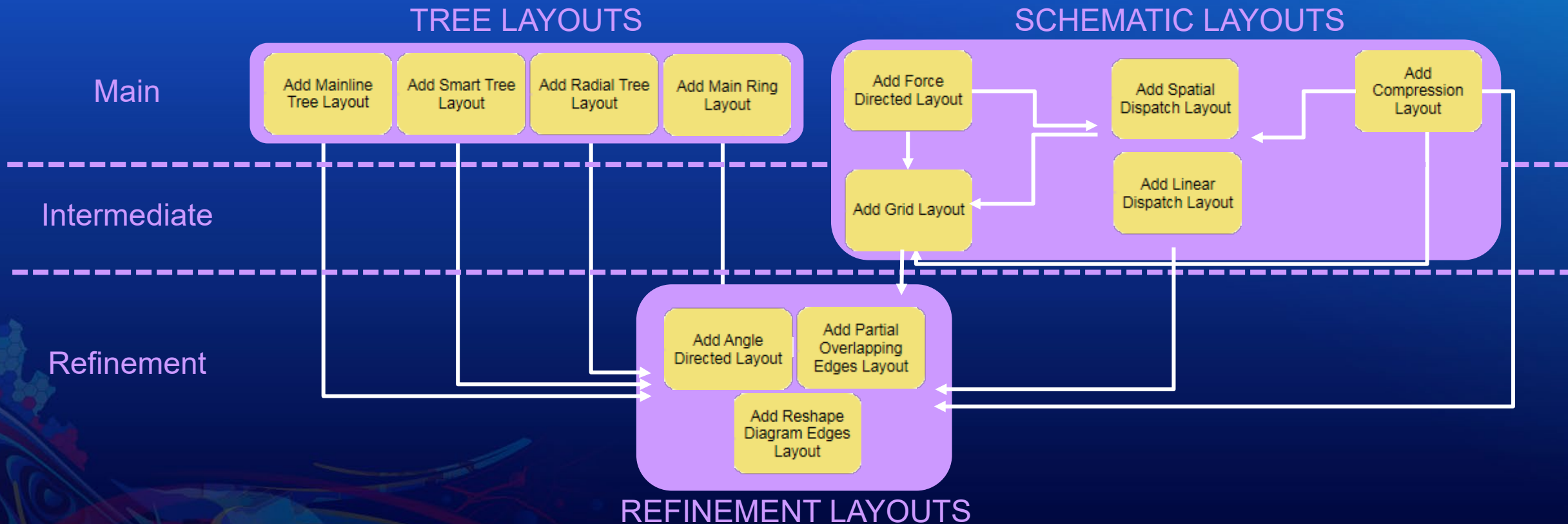
# Phase 2 - Diagram Rule Sequence Definition

- Diagram rule position in the rule sequence depends on the type of rule



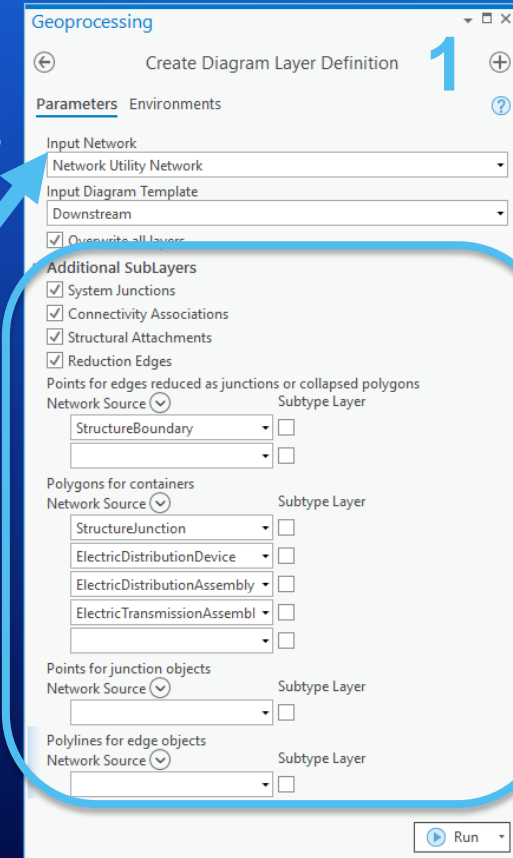
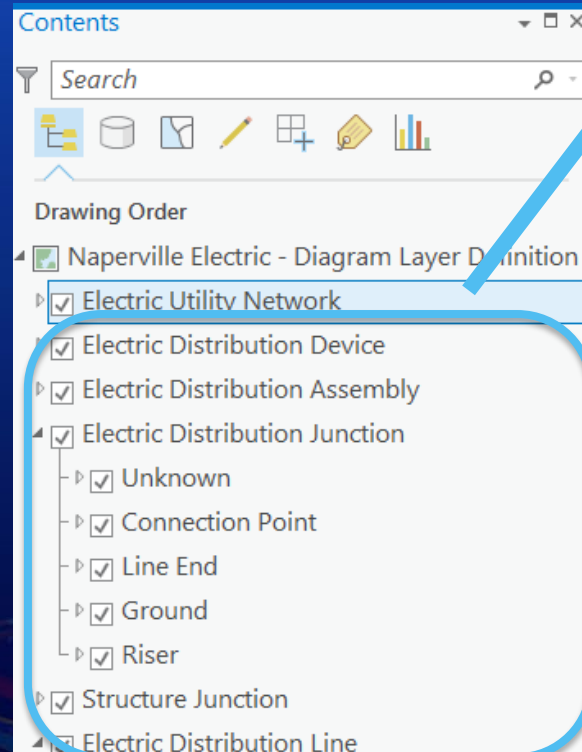
# Phase 3 - Diagram Layout Sequence Definition

- Diagram layout position in the layout sequence depends on the type of layout



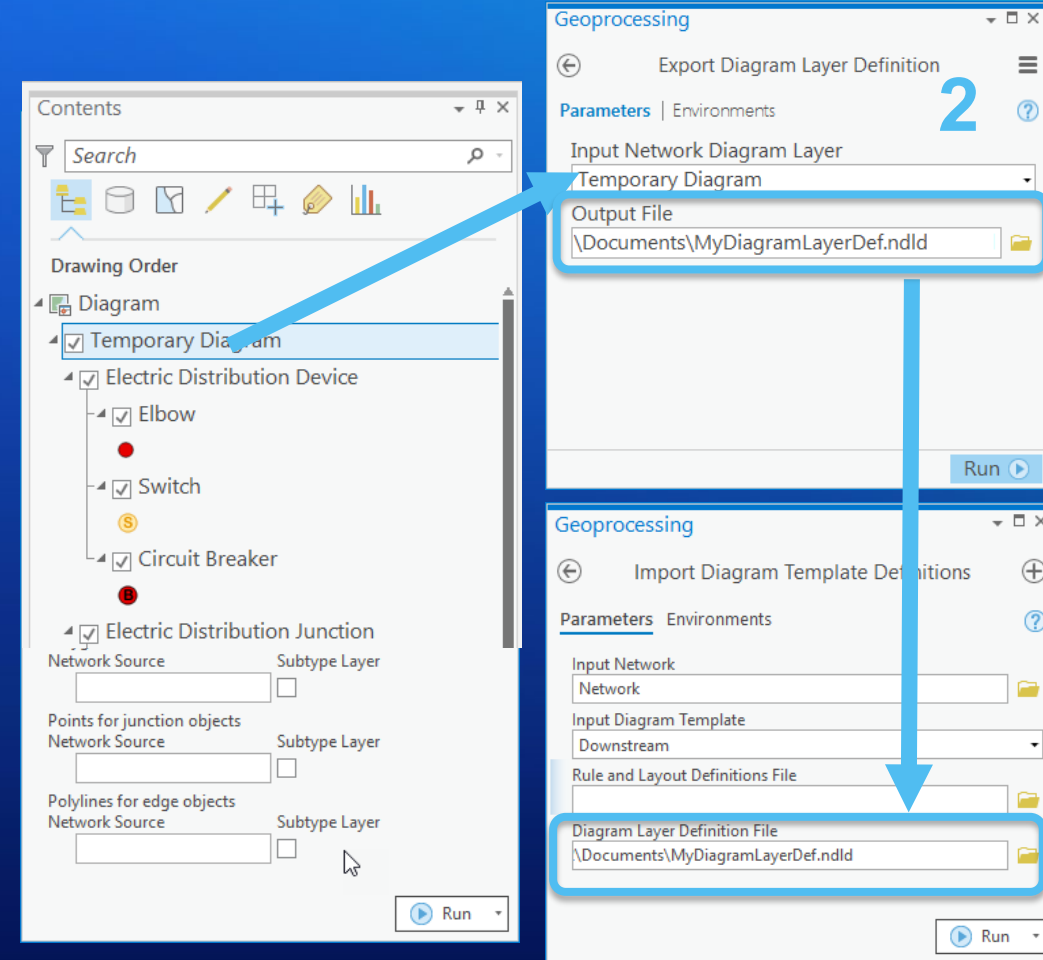
# Diagram Layer Management Tools

- Different tools to use for different purposes
  1. Initialize a diagram layer definition based on layers in a network map



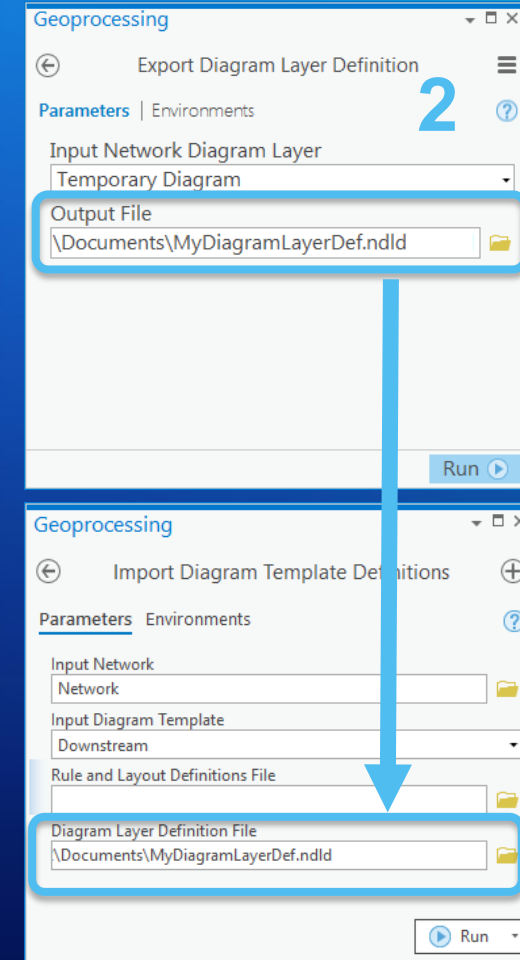
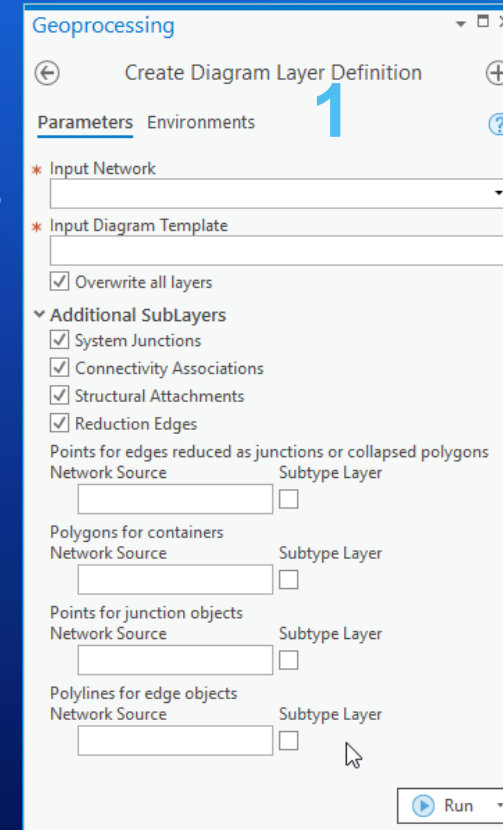
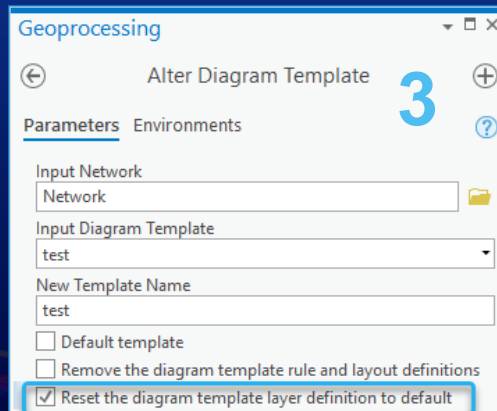
# Diagram Layer Management Tools

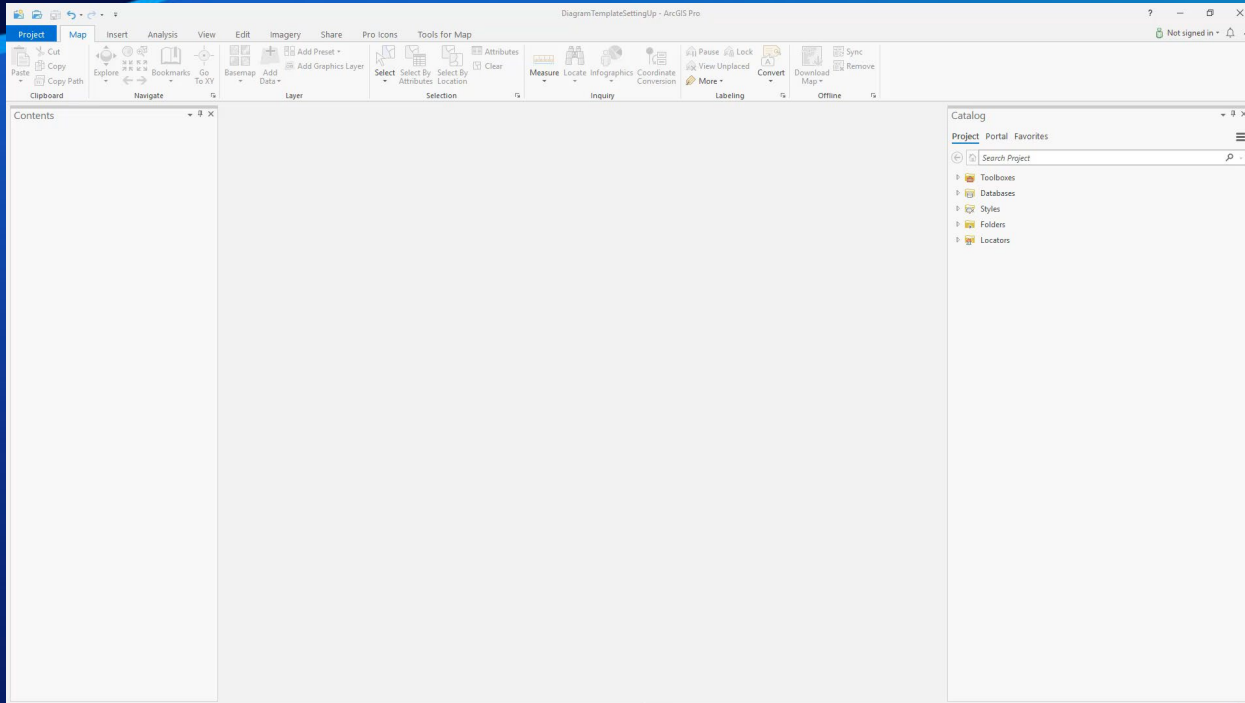
- Different tools to use for different purposes
  1. Initialize a diagram layer definition based on layers in a network map
  2. Refine an existing network diagram layer, then export the definition of the refined network layer and reimport it on the template



# Diagram Layer Management Tools

- Different tools to use for different purposes
  1. Initialize a diagram layer definition based on layers in a network map
  2. Refine an existing network diagram layer, then export the definition of the refined network layer and reimport it on the template
  3. Reset diagram layer definition on a template





# Demo#4

## Create a template to explore and control network data



# Network Diagrams APIs

# Network Diagrams APIs

- ArcGIS Pro SDK for .NET
- ArcGIS REST API
- ArcGIS Developer Enterprise SDK

# Network Diagrams ArcGIS Pro SDK for .NET

- Introduced with ArcGIS Pro 2.1
- Extend the core network diagram functionality in ArcGIS Pro
- Develop add-in commands and tools

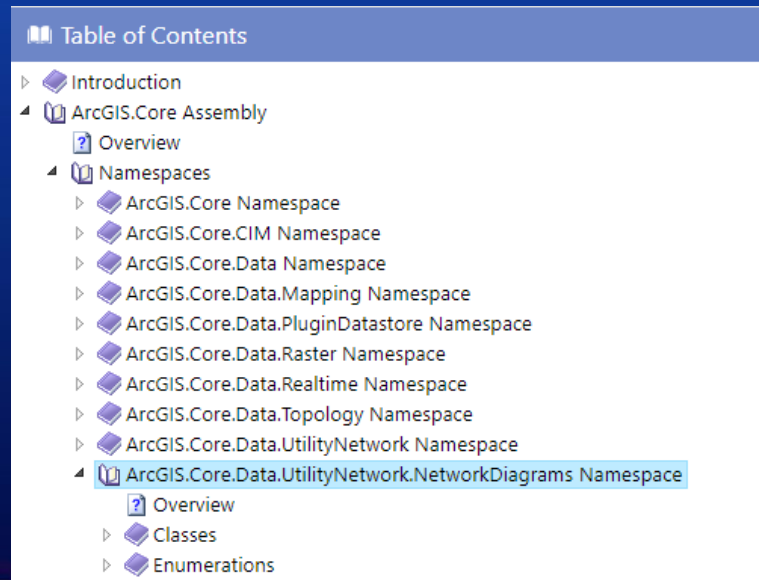
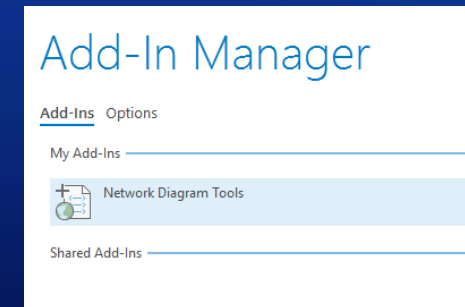


Table of Contents

- Introduction
- ArcGIS.Core Assembly
  - Overview
  - Namespaces
    - ArcGIS.Core Namespace
    - ArcGIS.Core.CIM Namespace
    - ArcGIS.Core.Data Namespace
    - ArcGIS.Core.Data.Mapping Namespace
    - ArcGIS.Core.Data.PluginDatastore Namespace
    - ArcGIS.Core.Data.Raster Namespace
    - ArcGIS.Core.Data.Realtime Namespace
    - ArcGIS.Core.Data.Topology Namespace
    - ArcGIS.Core.Data.UtilityNetwork Namespace
    - ArcGIS.Core.Data.UtilityNetwork.NetworkDiagrams Namespace
      - Overview
      - Classes
      - Enumerations



See <https://pro.arcgis.com/en/pro-app/latest/sdk/api-reference/#topic17701.html>

# Network Diagrams ArcGIS Pro SDK API - Diagram Manager

**UtilityNetwork**  
Sealed Class  
→ Dataset

Methods

- GetDiagramManager() : DiagramManager

**DiagramTemplate**  
Sealed Class  
→ CoreObjectsBase

Properties

- DiagramManager { get; } : DiagramManager
- Name { get; } : string

Methods

- GetNetworkDiagram(string name) : NetworkDiagram
- GetNetworkDiagrams() : IReadOnlyList<NetworkDiagram>

**NetworkDiagram**  
Sealed Class  
→ CoreObjectsBase

**DiagramManager**  
Sealed Class  
→ CoreObjectsBase

Methods

- CreateNetworkDiagram(DiagramTemplate diagramTemplate, IReadOnlyList<Guid> globalIDs) : NetworkDiagram
- GetDiagramTemplate(string name) : DiagramTemplate
- GetDiagramTemplates() : IReadOnlyList<DiagramTemplate>
- GetNetwork<T>() : T
- GetNetworkDiagram(string name) : NetworkDiagram
- GetNetworkDiagrams() : IReadOnlyList<NetworkDiagram>
- GetNetworkDiagrams(Envelope extentOfInterest) : IReadOnlyList<NetworkDiagram>
- GetNetworkDiagrams(Envelope extentOfInterest, IReadOnlyList<Guid> globalIDs) : IReadOnlyList<NetworkDiagram>
- GetNetworkDiagrams(IReadOnlyList<Guid> globalIDs) : IReadOnlyList<NetworkDiagram>

# Network Diagrams ArcGIS Pro SDK API - Network Diagram

**NetworkDiagram**  
Sealed Class  
↳ CoreObjectsBase

Properties

- DiagramManager { get; } : DiagramManager
- DiagramTemplate { get; } : DiagramTemplate
- Name { get; } : string

Methods

- AddFlag(NetworkDiagramFlagType flagType, int diagramElementID) : void
- Append(IReadOnlyList<Guid> globalIDs) : void
- ApplyLayout(DiagramLayoutParameters layoutParameters) : void
- ApplyLayout(DiagramLayoutParameters layoutParameters, DiagramElementObjectIDs subset) : void
- ApplyTemplateLayouts() : void
- Delete() : void
- Extend(NetworkDiagramExtendType extendType) : void
- Extend(NetworkDiagramExtendType extendType, IReadOnlyList<Guid> globalIDs) : void
- FindDiagramFeatures(FindDiagramFeatureQuery query) : IReadOnlyList<FindResultItem>
- FindInitialNetworkRows() : IReadOnlyList<FindResultItem>
- FindNetworkRows(FindNetworkRowQuery query) : IReadOnlyList<FindResultItem>
- GetAggregations() : IReadOnlyList<DiagramAggregation>
- GetConsistencyState() : NetworkDiagramConsistencyState
- GetContent(bool addDiagramInfo, bool addGeometries, bool addAttributes, bool addAggregations) : string
- GetDiagramInfo() : NetworkDiagramInfo
- GetFlags(NetworkDiagramFlagType flagType) : IReadOnlyList<DiagramFlag>
- Overwrite(IReadOnlyList<Guid> globalIDs) : void
- QueryDiagramElements(DiagramElementQueryByElementTypes query) : DiagramElementQueryResult
- QueryDiagramElements(DiagramElementQueryByExtent query) : DiagramElementQueryResult
- QueryDiagramElements(DiagramElementQueryByObjectIDs query) : DiagramElementQueryResult
- RemoveFlag(NetworkDiagramFlagType flagType, int diagramElementID) : void
- RemoveFlags(NetworkDiagramFlagType flagType) : void
- SaveLayout(NetworkDiagramSubset subset, bool keepVertices) : void
- Store(string name, NetworkDiagramAccessType accessType, string tag) : void
- Update() : void

**NetworkDiagramInfo**  
Sealed Class

Properties

- Access { get; } : NetworkDiagramAccessType
- CanExtend { get; } : bool
- CanStore { get; } : bool
- ContainerMargin { get; } : double
- CreationDate { get; } : DateTime
- Creator { get; } : string
- DiagramExtent { get; } : Envelope
- IsStored { get; } : bool
- IsHistorical { get; } : bool
- IsSystem { get; } : bool
- LastUpdateBy { get; } : string
- LastUpdateDate { get; } : DateTime
- NetworkExtent { get; } : Envelope
- Tag { get; } : string

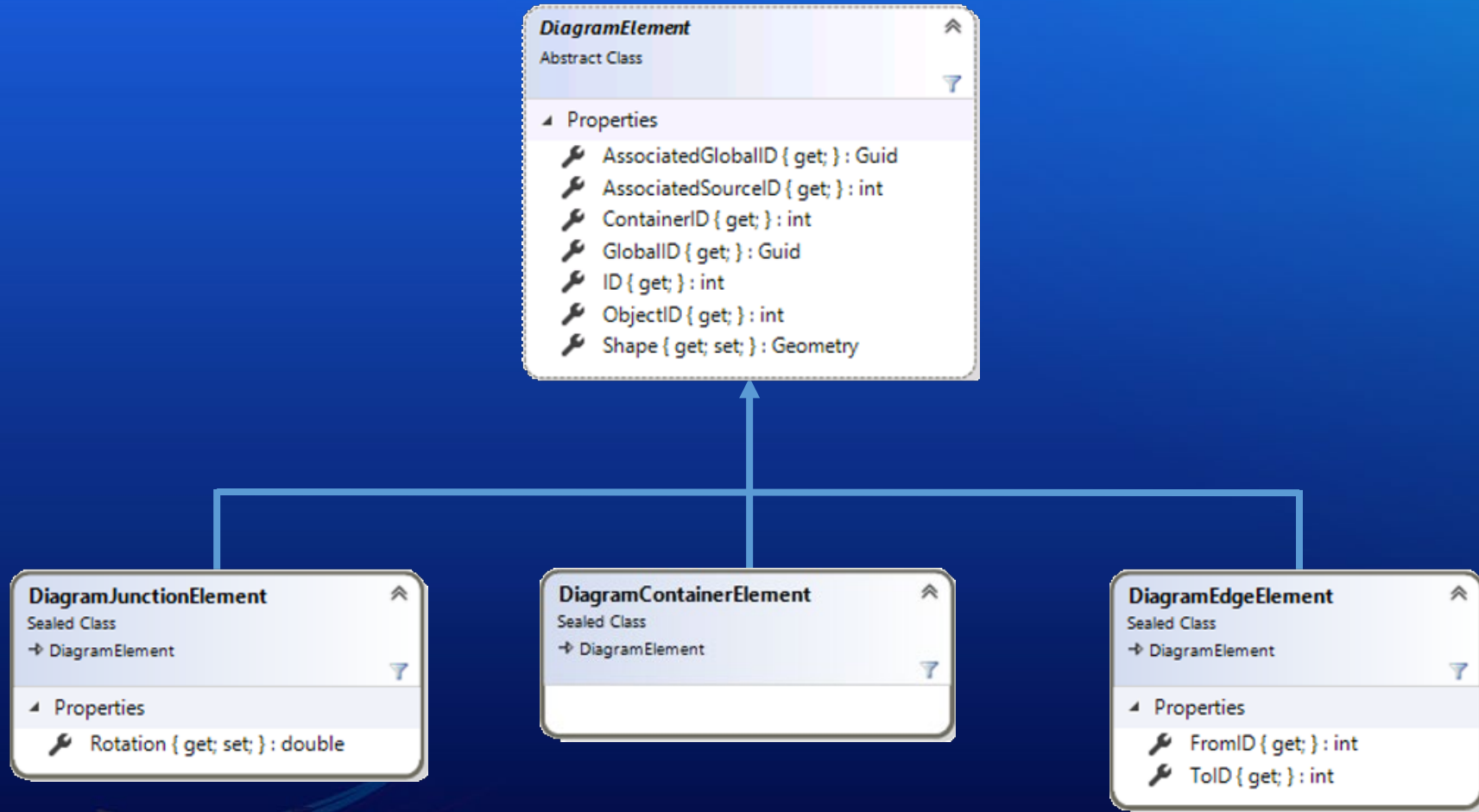
**NetworkDiagramConsistencyState**  
Enum

- esriDiagramIsConsistent
- esriDiagramNotConsistentWithTopology
- esriDiagramHasDirtyFeatures

**NetworkDiagramAccessType**  
Enum

- PublicAccess
- ProtectedAccess
- PrivateAccess

# Network Diagrams ArcGIS Pro SDK API - Diagram Element



# Network Diagrams ArcGIS Pro SDK API - Diagram Aggregations

**NetworkDiagram**  
Sealed Class  
→ CoreObjectsBase

Methods

- GetAggregations() : IReadOnlyList<DiagramAggregation>

**DiagramAggregation**  
Sealed Class

Properties

- AggregatedBy { get; } : int
- AggregationType { get; } : NetworkDiagramAggregationType
- AssociatedGlobalID { get; } : Guid
- AssociatedSourceID { get; } : int

**NetworkDiagramAggregationType**  
Enum

- NoneAggregation
- EdgeAggregation
- JunctionAggregation
- ContainerAggregation

# Network Diagrams ArcGIS Pro SDK API - Diagram Layouts



**NetworkDiagram**  
Sealed Class  
↳ CoreObjectsBase

Methods

- ApplyLayout(DiagramLayoutParameters layoutParameters) : void
- ApplyLayout(DiagramLayoutParameters layoutParameters, **DiagramElementObjectIDs subset**) : void

**DiagramElementObjectIDs**  
Sealed Class

Properties

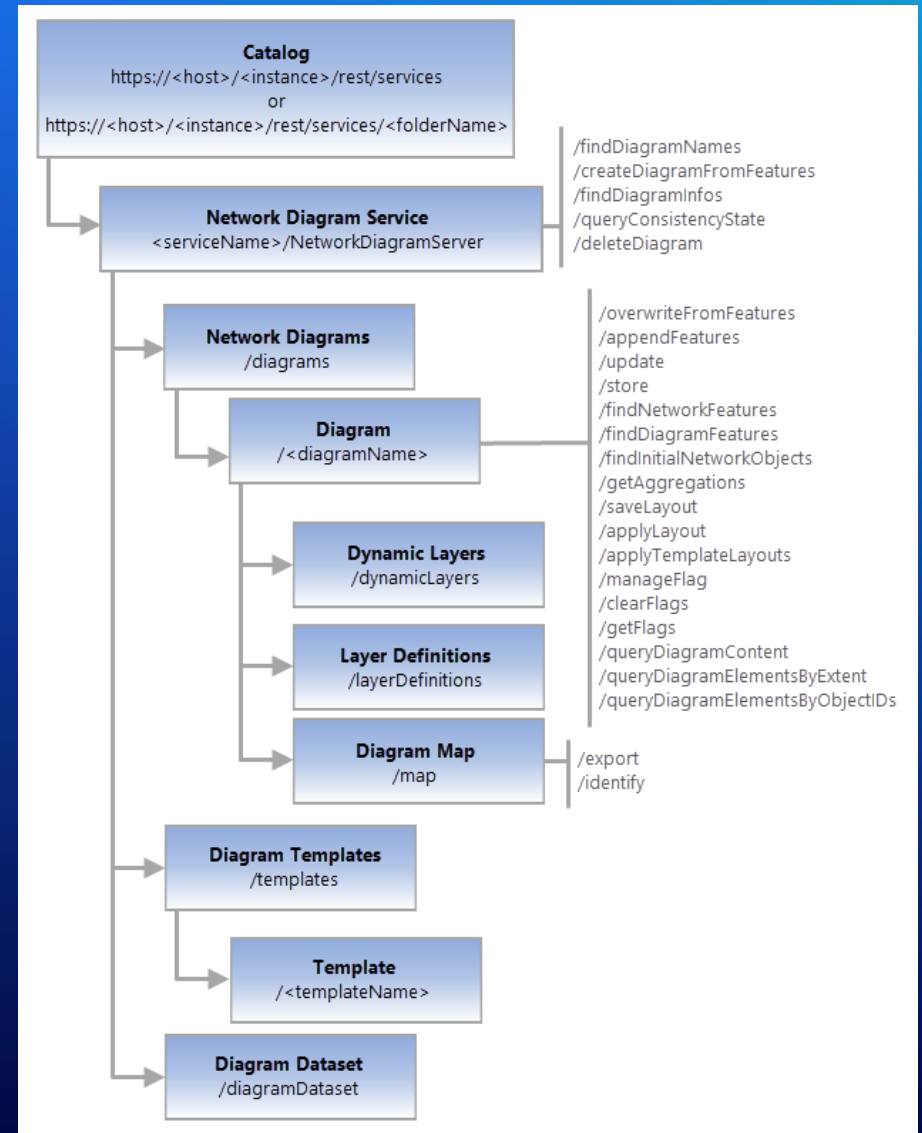
- ContainerObjectIDs { get; set; } : IReadOnlyList<long>
- EdgeObjectIDs { get; set; } : IReadOnlyList<long>
- JunctionObjectIDs { get; set; } : IReadOnlyList<long>

Methods

- DiagramElementObjectIDs()

# Network Diagrams ArcGIS REST API

- Introduced with ArcGIS Enterprise 10.6
- To develop your web applications based on network diagram services
- REST resource hierarchy related to a network diagram service
  - Information about the service itself
  - Various network diagram functions



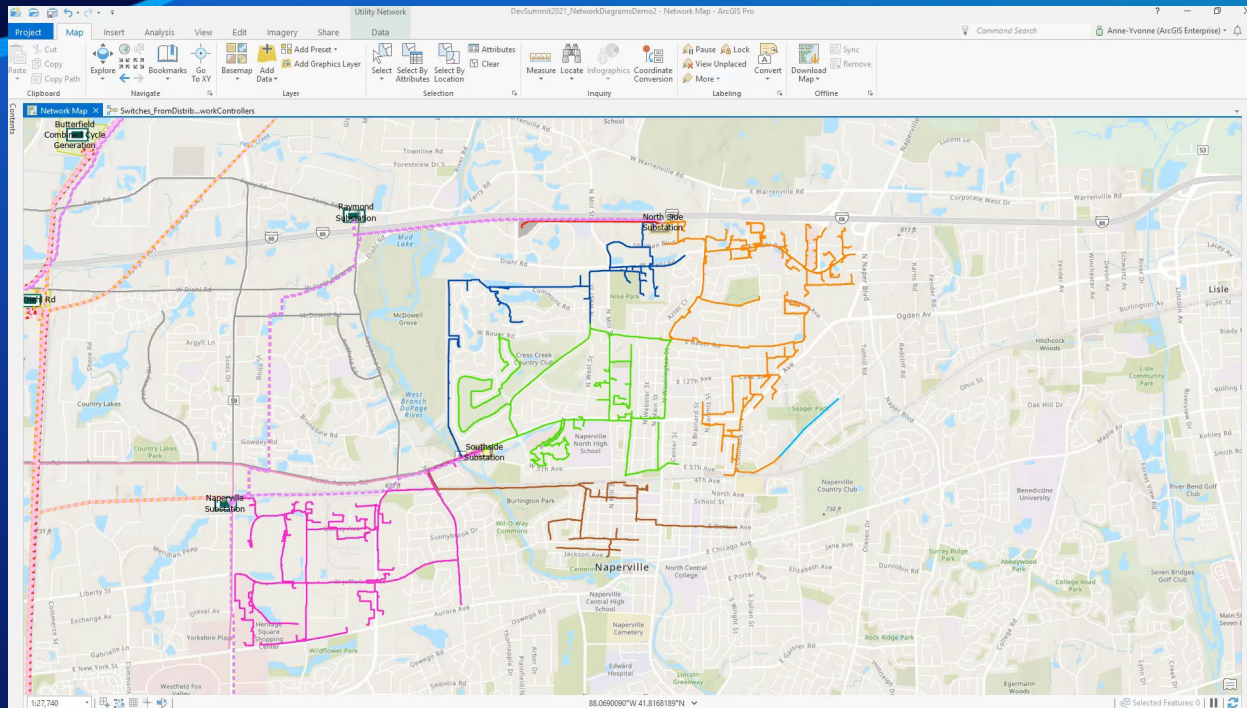
# Network Diagrams ArcGIS Developer Enterprise SDK

- New in ArcGIS Enterprise 10.9
- Part of the Geodatabase library
- To create SOEs that target network diagrams
- OMD close to the Pro SDK OMD

```
public IDiagramDataset GetDiagramDataset()  
{  
    IBaseNetwork unBaseNetwork = GetUtilityNetwork();  
  
    if (unBaseNetwork != null)  
        return unBaseNetwork.DiagramDataset as IDiagramDataset;  
  
    return null;  
}
```

```
public IEnumDiagramTemplate GetDiagramTemplates(IDiagramDataset DiagramDataset)  
{  
    if (DiagramDataset != null)  
        return DiagramDataset.DiagramTemplates;  
  
    return null;  
}
```

```
public IEnumNetworkDiagram GetDiagrams(IDiagramDataset DiagramDataset)  
{  
    if (DiagramDataset != null)  
        return DiagramDataset.NetworkDiagrams;  
  
    return null;  
}
```



# Demo#5

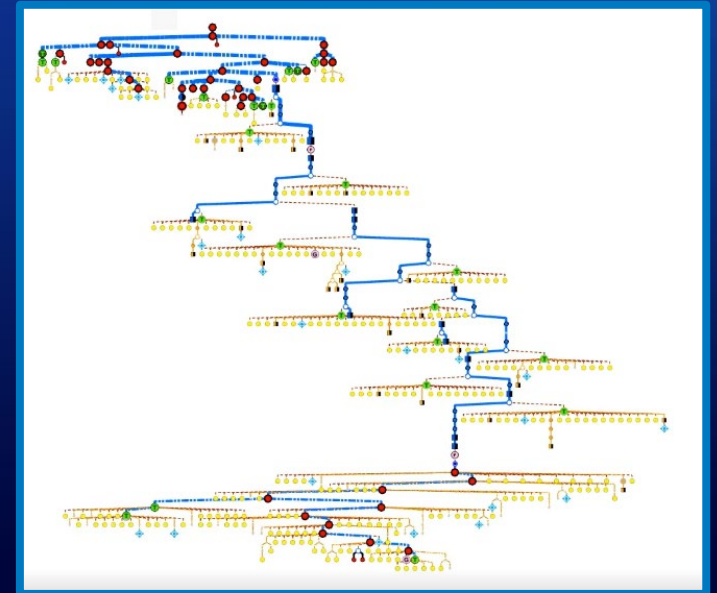
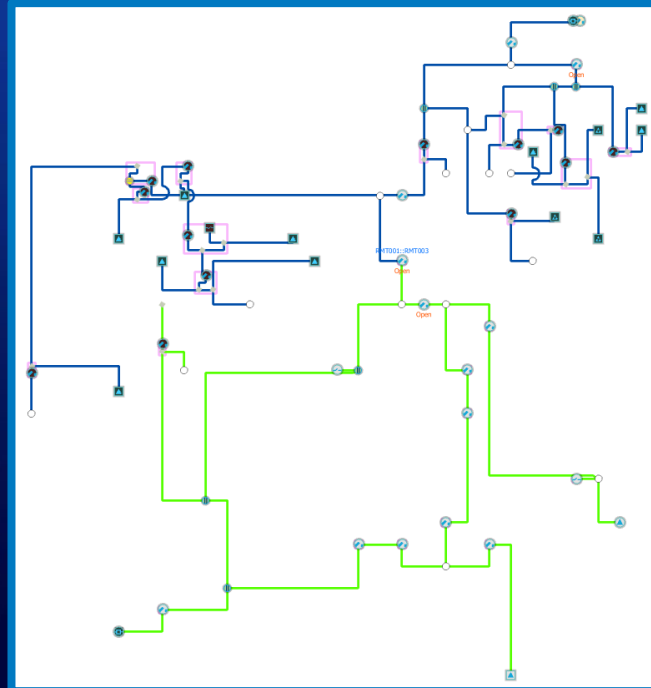
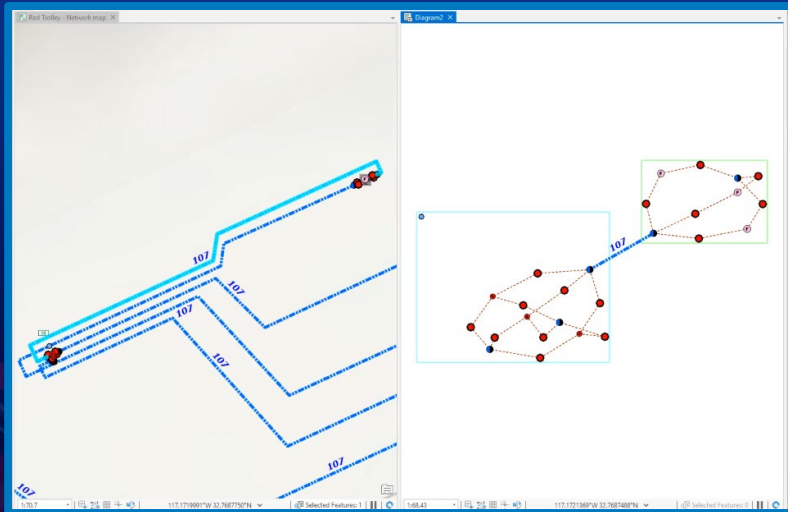
## Switching diagrams, sample add-ins and web app



# Network Diagrams Benefits

# Network Diagrams Benefits

- Explore networks and control data quality
- Provide clear and synthetic views of your network areas
- Built from preset network traces that reflect the current network status at update



# To Learn More

- **ArcGIS blogs**

- <https://www.esri.com/arcgis-blog/products/utility-network/data-management/best-practices-to-configure-great-diagram-templates/>
- <https://www.esri.com/arcgis-blog/products/utility-network/data-management/optimizing-layer-definitions-on-your-diagram-templates/>

- **Tutorial to set up templates to create diagrams showing interconnected substations**

- <https://www.youtube.com/watch?v=h58hZNFvSXI&feature=youtu.be>

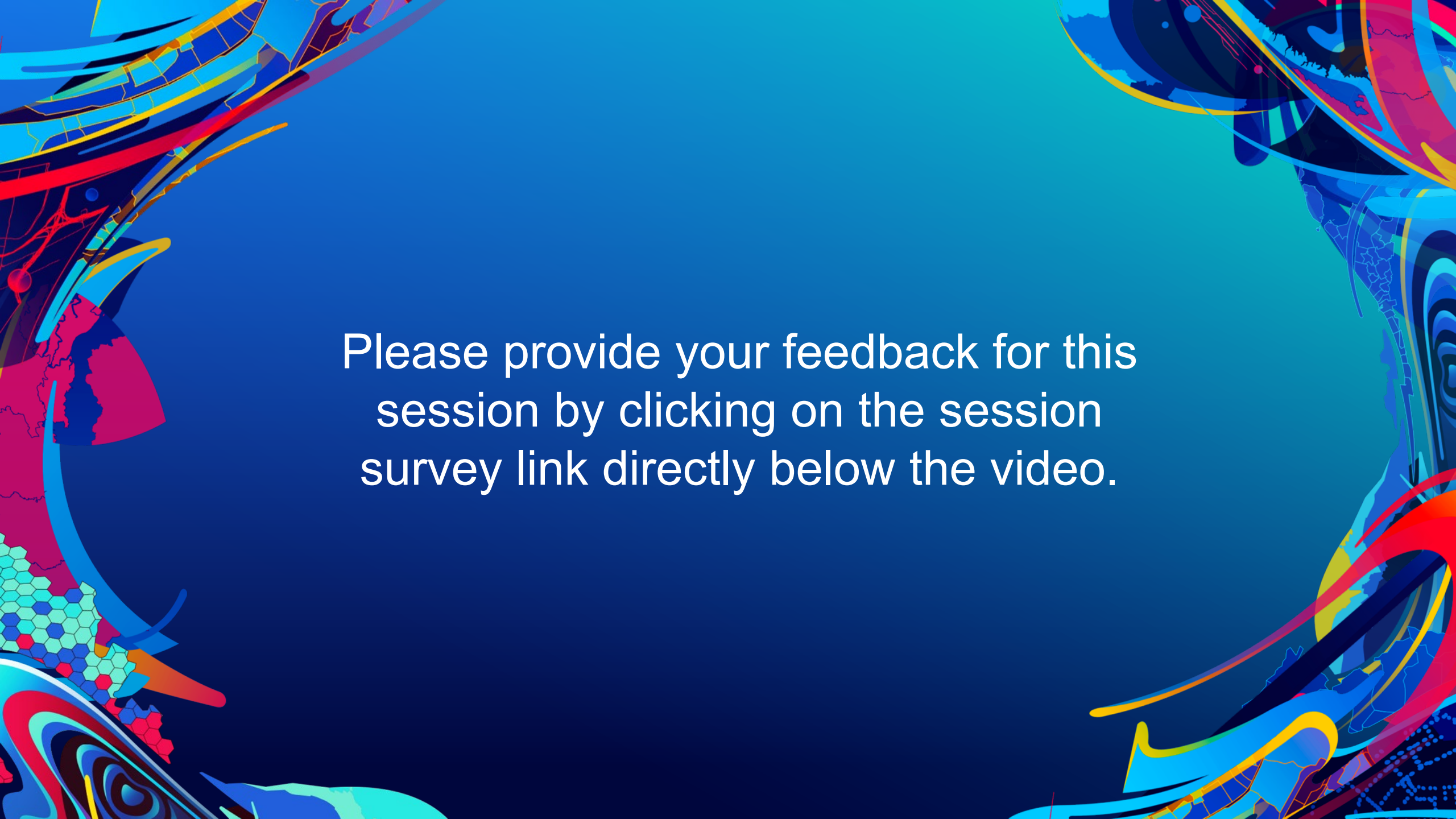
- **Help Systems**

- <https://pro.arcgis.com/en/pro-app/latest/help/data/network-diagrams/about-network-diagrams.htm>
- <https://pro.arcgis.com/en/pro-app/latest/tool-reference/network-diagram/an-overview-of-the-network-diagram-toolbox.htm>



esri®

THE  
SCIENCE  
OF  
WHERE®



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