Network management in ArcGIS

- Many options exist today for network management in ArcGIS
  - Fiber, coax, copper, wireless, pipes, wires, circuits, devices
  - Based on ArcMap and/or geometric network
  - ArcMap will continue to be supported far into the future; no end of life published

- Currently the utility network supports spatial network management workflows
  - Network features with geometry
  - This is a great way to get started for partners or customers

- Later this summer, utility network will add additional support for nonspatial network management workflows
  - Network features without geometry

Esri confidential – contents subject to change

Nonspatial Object Support
in the Utility Network
Presentation

Nonspatial Design

ArcGIS Utility Network

April 24, 2020

Topics

- Information model
- Dirty area manager
- Associations and rules
- Error features
- Build / network index
- Trace framework
- Subnetwork management
- Upgrade
- Pro SDK
- REST API

Esri confidential – contents subject to change

Information model

Information model

summary of changes

- Telco/fiber modeling will be enhanced with new abstractions called junction objects and edge objects
  - These are nonspatial objects which may be contained inside spatial objects or other nonspatial objects (e.g., modeling fiber strands inside of fiber cables)
  - Edge objects correspond to edge elements in the network index
  - Junction objects correspond to junction elements in the network index
- Naming convention: DomainNetworkName + JunctionObject/EdgeObject
  - E.g., in the Fiber domain network, FiberJunctionObject, and FiberEdgeObject
- Structure junction and edge object tables are created
- No schema changes are required for Device, Junction, Assembly, and Line classes

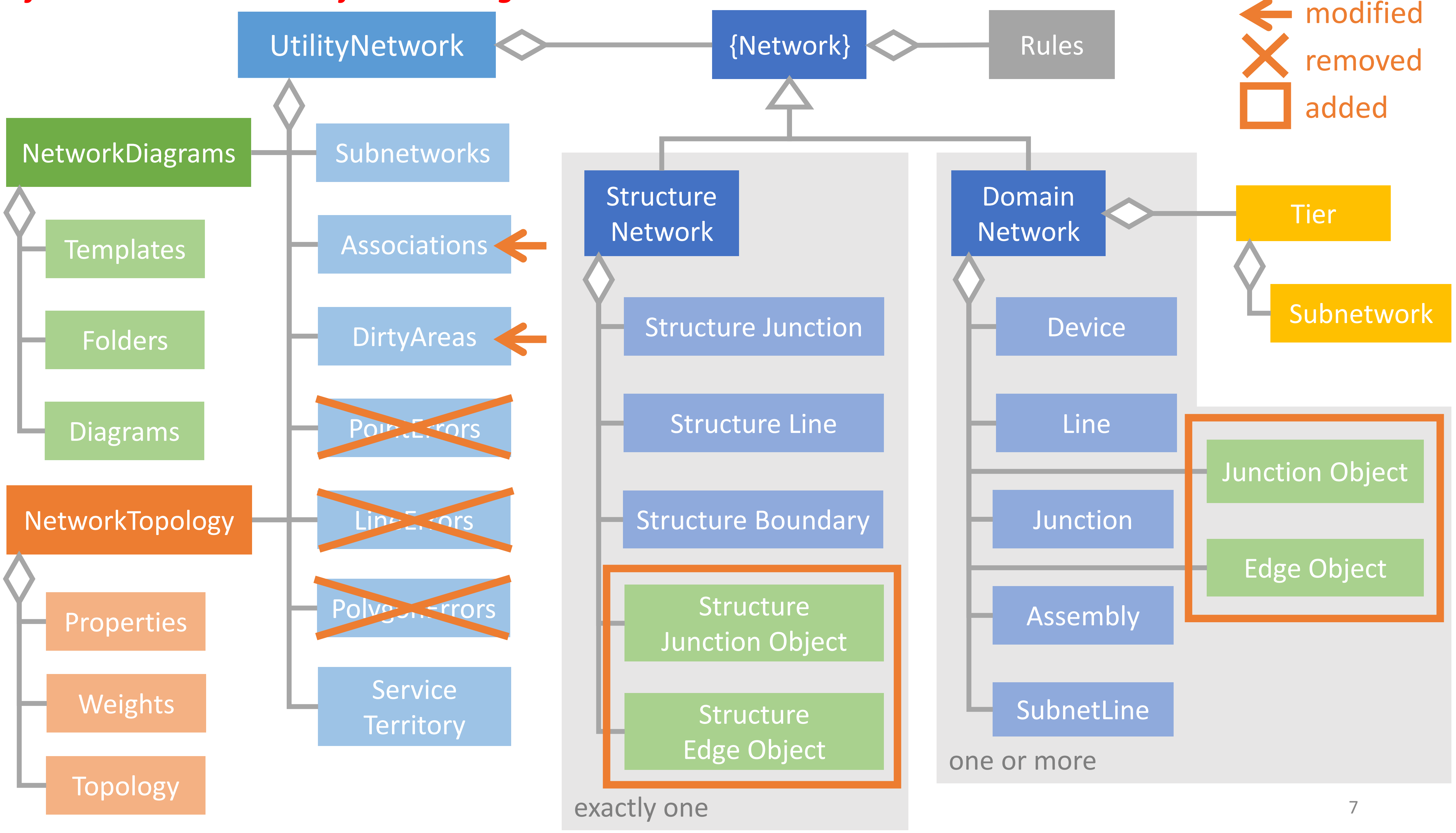
Information model

summary of changes

- The error tables that are available up to utility network with schema generation 3 (current utility network at 2.5/10.8) will be removed with schema generation 4

- With schema generation 4, the error information is persisted:
  - Feature errors – with an error code and error message in the DirtyAreas table
  - Object errors – with an error code and error message in the Associations table

Esri confidential – contents subject to change



Information model

network classes

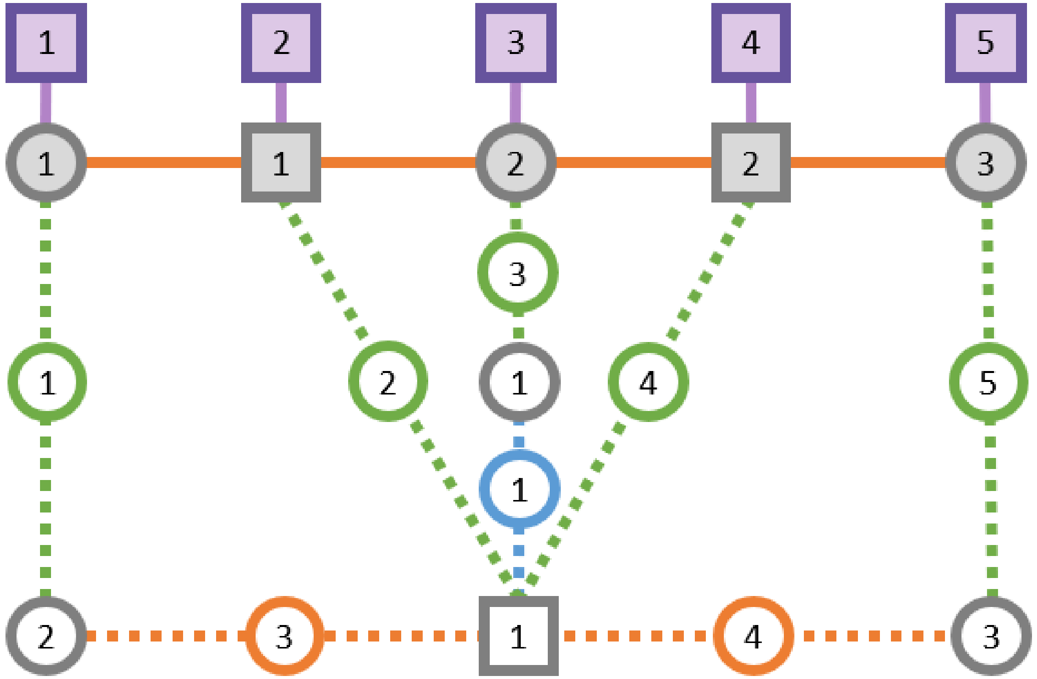
- Junction and edge objects are locatable
  - Junction objects may be contained within a spatial feature found in the containment/attachment hierarchy
  - Edge objects may be contained within a spatial feature in the containment/attachment hierarchy or locatable at their endpoints
  - Junction and edge objects may not be locatable
    - Load nonspatial objects into tables and enable network topology
- Similar to other feature tables, users can add fields to the object classes
- Edge objects are connected to junction objects via junction-edge connectivity association
- Junction objects can be controllers and can have terminal configurations

Information model

midspan junction objects

- Junction objects can occur at midspan on edge objects, but not on other junction objects or lines; they require an additional connectivity association type (in blue)
- Edits with <=0% or >=100% along the line are rejected; the midspan junction is parent to the edge object (similar to the endpoint junction objects)

- # Dirty area
- # Line
- # Point
- # Edge object
- # Junction object
- ▨ Dirty feature/object



- Dirty area relationship
- Geometric connectivity
- # Connectivity association
- # Edge-junction connectivity
- # Containment/attachment
- # Junction midspan connectivity

Junction objects

schemas

System maintained attributes
User maintained attributes

DomainJunctionObject

ObjectID
GlobalID
AssetGroup
AssetType
TerminalConfiguration
AssociationStatus

Subnetwork management:

- IsSubnetworkController
- SubnetworkControllerName
- TierName
- TierRank
- IsConnected
- SubnetworkName
- SupportedSubnetworkName

StructureJunctionObject

ObjectID
GlobalID
AssetGroup
AssetType
AssociationStatus
SubnetworkName

Edge objects

schemas

System maintained attributes
User maintained attributes

DomainEdgeObject

ObjectID
GlobalID
AssetGroup
AssetType
AssociationStatus

Subnetwork management:

- IsConnected
- SubnetworkName
- SupportedSubnetworkName

StructureEdgeObject

ObjectID
GlobalID
AssetGroup
AssetType
AssociationStatus
SubnetworkName

Enumerations

new or updated

New
Existing

```
esriDirtyAreaStatus                                  // for Status attribute of DirtyAreas table, bit value
    esriDASDisabled                          = 0
    esriDASInsertedUpdatedFeature            = 1
    esriDASDeletedFeature                    = 2
    esriDASModifiedObjects                   = 4
    esriDASFeatureError                      = 8
    esriDASObjectError                       = 16
    esriDASSubnetworkError                   = 32

esriUtilityNetworkAssociationType
    esriUNATJunctionJunctionConnectivity     = 1
    esriUNATContainment                      = 2
    esriUNATAttachment                       = 3
    esriUNATJunctionEdgeFromConnectivity     = 4   // Junction is connected to edge at the ‘from’ side of edge
    esriUNATJunctionEdgeMidspanConnectivity  = 5   // Junction is connected to the midspan of the edge
    esriUNATJunctionEdgeToConnectivity       = 6   // Junction is connected to edge at the ‘to’ side of edge

esriUtilityNetworkAssociationTableStatus             // for Status attribute of Associations table, bit value
    esriUNATSNone                            = 0    //     Nothing is deleted or dirty
    esriUNATSAssociationDeleted              = 1    //     Association itself is deleted
    esriUNATSFromDeleted                     = 2    //     From side is deleted
    esriUNATSToDeleted                       = 4    //     To side is deleted
    esriUNATSAssociationDirty                = 8    //     Association itself is dirty
    esriUNATSFromDirty                       = 16   //     From side is dirty
    esriUNATSToDirty                         = 32   //     To side is dirty
```

Esri confidential – contents subject to change

Enumerations

new or updated

New
Existing

UtilityNetworkFeatureClassUsageType

esriUNFCUTDevice
esriUNFCUTJunction
esriUNFCUTLine
esriUNFCUTAssembly
esriUNFCUTSubnetLine
esriUNFCUTStructureJunction
esriUNFCUTStructureLine
esriUNFCUTStructureBoundary
esriUNFCUTJunctionObject
esriUNFCUTEdgeObject
esriUNFCUTStructureJunctionObject
esriUNFCUTStructureEdgeObject

esriAssociationStatus

esriASNone	= 0
esriASContainer	= 1
esriASStructure	= 2
esriASContent	= 4
esriASAttachment	= 8
esriASVisibleContent	= 16
esriASConnectivity	= 32

// More combinations are added due to the built-in
// rule restriction changes

Esri confidential – contents subject to change

Associations

schema

System maintained attributes
System maintained new attributes with 2.6

ObjectID
GlobalID
From/ToNetworkSourceID
From/ToGlobalID
From/ToTerminalID
IsContentVisible
AssociationType

PercentAlong

For midspan connectivity, Nullable, Double
Accept value greater than 0.0 and less than 1.0

The valid value is checked via API
Value is set with esriUNATJunctionMidspanConnectivity

Status

Not nullable, Integer
Default: esriUNATSNone (0)

ErrorCode

Bit encoded value, Double

ErrorMessage

Nullable, String, Length 512

Esri confidential – contents subject to change

DirtyAreas

schema

System maintained attributes
System maintained new attributes with 2.6

ObjectID
GlobalID
IsRetired
SourceID
Guid
DirtyArea (geometry)

Status

Not nullable, Integer
Default value: esriDAETInsertedUpdatedFeature (1)
Value: esriDirtyAreaStatus (bit values)

Domain is not required for this field

ErrorCode

Bit encoded value, Double

ErrorMessage

Nullable, String, Length: 512

Information model

network definition

- Data element is augmented to support the two additional object classes
  - Use UsesGeometry (false) and ShapeType (null) properties of network source to indicate nonspatial object tables, no new network source is necessary

- The tier definition requires updates:
  - If necessary, the UpdateSubnetworkTraceConfiguration is enhanced (paralleling any enhancements needed by the more general trace configuration
  - ValidEdgeObjects, ValidJunctionObjects, and ValidJunctions are added (paralleling ValidLines, and ValidDevices)

Information model

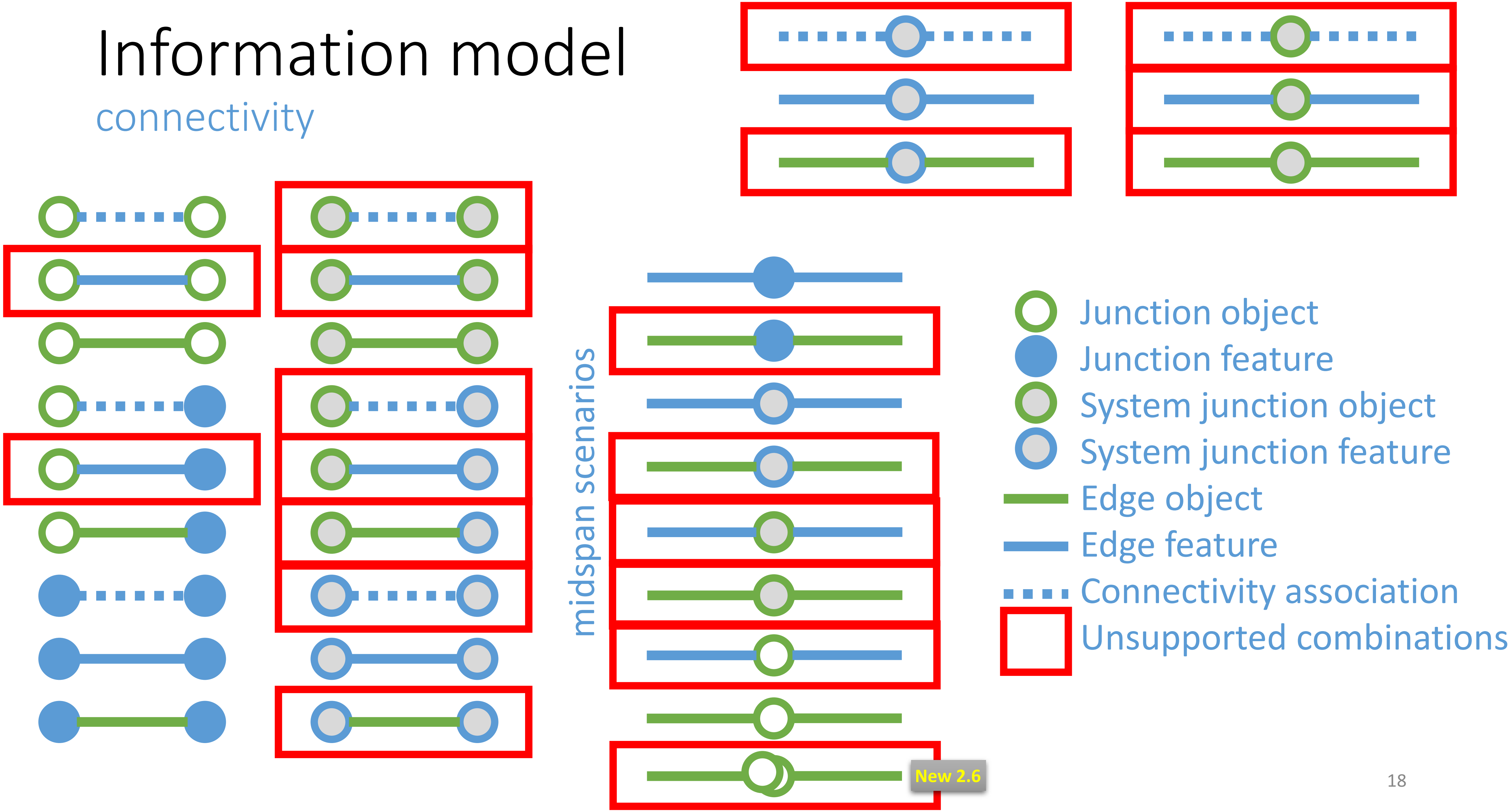
connectivity associations – additional permissible combinations

- Edge objects and point features
- Edge objects and junction objects
- Junction objects and point and line features
- Junction objects and junction objects
- Junction objects to midspan of edge objects
- Containers and content
  - Junctions and edge/junction objects
  - Boundary structures and edge/junction objects
  - Lines and edge objects

Esri confidential – contents subject to change

Information model

connectivity



midspan scenarios

New 2.6

- ○ Junction object
- ● Junction feature
- ◎ System junction object
- ◉ System junction feature
- ▬ Edge object
- ▬ Edge feature
- ▪▪▪ Connectivity association
- □ Unsupported combinations

Information model

connectivity associations – association status values

- Association status currently assumes that all features involved with associations have geometry
- A container may have both features and objects
- Association status values are not constrained when edge / junction objects are involved
  - E.g., a container's association status may be esriASVisibleContentAndContainer even if none of the content has geometry
  - Synthesize geometry of the objects will be a required capability

Information model

containment associations – additional permissible configurations

- Edge objects can contain line and edge objects
- Junction objects can contain edge and junction objects
- Lines can contain edge objects
- Devices and junctions can contain junction objects
- Structure features (assemblies, structure junctions, structure lines, and structure boundaries) can contain edge and junction objects

Esri confidential – contents subject to change

Associations

permissible configurations

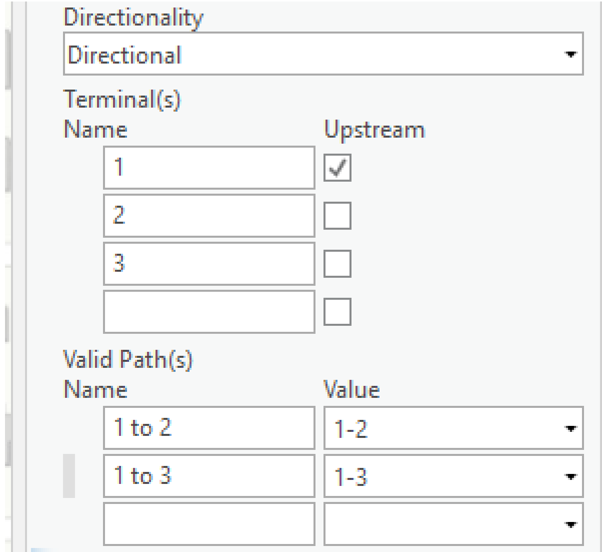
- 🟩 Connectivity association (junction-junction)
- 🟢 Connectivity – (junction-edge)
- 🟦 Connectivity - geometric coincidence (system junction)
- 🔵 Connectivity - geometric coincidence
- 🟧 Containment association
- 🟨 Structural attachment association

		From: Container / Structure										
		Device	Junction	Line	Edge Object	Junction Object	Assembly	Structure Junction	Structure Line	Structure Junction Object	Structure Edge Object	Structure Boundary
To: Content / Attachment	Device	🟩🟧	🟩🟧	🔵🟧	🟢🟧	🟩🟧	🟧	🟨🟧	🟧	🟨🟧	🟧	🟧
	Junction	🟩🟧	🟩🟧	🔵🟧	🟢🟧	🟩🟧	🟧	🟨🟧	🟧	🟨🟧	🟧	🟧
	Line	🔵🟧	🔵🟧	🟦🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
	Edge Object	🟢🟧	🟢🟧	🟧	🟧	🟢🟧	🟧	🟧	🟧	🟧	🟧	🟧
	Junction Object	🟩🟧	🟩🟧	🟧	🟢🟧	🟩🟧	🟧	🟨🟧	🟧	🟨🟧	🟧	🟧
	Assembly						🟧	🟨🟧		🟨🟧		🟧
	Structure Junction							🟨🟩🟧	🔵	🟨🟩🟧	🟢	🟧
	Structure Line							🔵🟧	🟦🟧	🟧	🟧	🟧
	Struct Junc Object							🟨🟩🟧		🟨🟩🟧	🟢	🟧
	Struct Edge Object							🟢🟧	🟧	🟢🟧	🟧	🟧
	Structure Boundary							🟧		🟧		🟧

Information model

Terminals

- Junction objects can serve as network controllers; as a consequence, they also need to support the terminal capability
  - Needed for more general modeling capabilities; e.g., transceivers as three terminal junction objects
- Directional terminal paths are necessary to support certain types of telco equipment – e.g., transceivers



Directionality

Directional

Terminal(s)

Name	Upstream
1	☑
2	□
3	□
	□

Valid Path(s)

Name	Value
1 to 2	1-2
1 to 3	1-3

Behavior changes

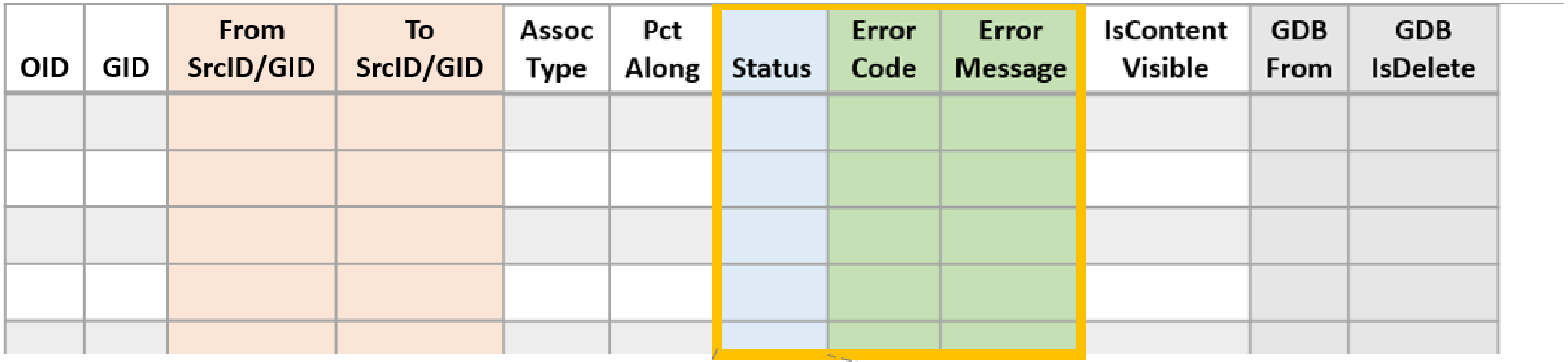
- When creating a connectivity association, do not flip the order (i.e., resorted by sourceID/GlobalID; the lowest sourceID goes first)
  - Required to preserve the digitized direction
- Allow the deletion of an association that will lead to unlocatable objects
- Do not allow multiple parents on a point feature or a junction object
  - If this is detected during validate, an error will be created
- Allow multiple parents on a line feature or an edge object

Dirty area management

Association table

schema

Associations



OID	GID	From SrcID/GID	To SrcID/GID	Assoc Type	Pct Along	Status	Error Code	Error Message	IsContent Visible	GDB From	GDB IsDelete

- New Fields
  - Status
  - Error code
  - Error message
  - Percent along

Status bits:
1 – Deleted association
2 – Deleted FROM feature/object
3 – Deleted TO feature/object
4 – Modified association
5 – Modified FROM feature/object
6 – Modified TO feature/object

Dirty area table

schema

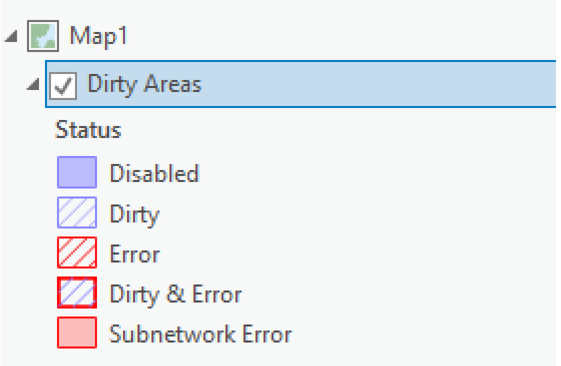
OID	GID	SrcID/GID	Status	Error Code	Error Message	GDB From	GDB IsDelete
10	D10	L1	**FFFTTF**	000…0010	<feature>	T17	T
11	D11	P2	**FFFFTF**	100…0000	<object>	T17	T
12	D12	L1	**FFTFFF**	<null>	<null>	T18	
13	D13	L1	**FFFTFF**	000…1010	<multiple>	T25	

- New Fields
  - Status
  - Error code
  - Error message
- Removed Fields
  - Edit type

Status bits:
1 – Inserted/updated feature
2 – Deleted feature
3 – Modified object(s)
4 – Feature error
5 – Object error
6 – Subnetwork error

How do I view my errors?

- Unique Value Renderer
- When the utility network layer is added to a map in Pro, Dirty Areas will automatically be symbolized based on the error type



Map1

☑ Dirty Areas

Status

🟦 Disabled

▨ Dirty

▨ Error

▨ Dirty & Error

🟥 Subnetwork Error

Rules and associations

Rules

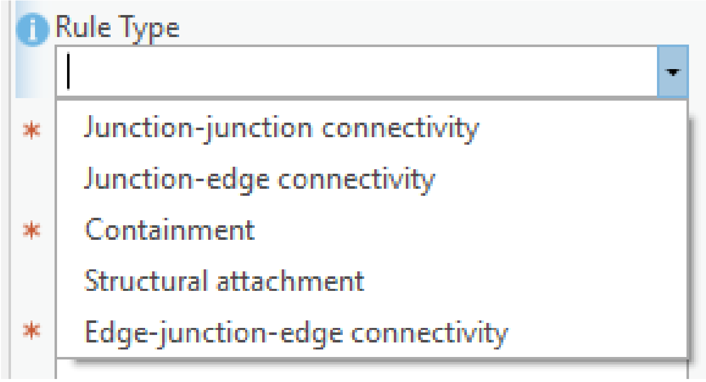
Summary of changes

- Updated GP tools:
  - Add Rule
  - Import Rules
  - Export Rules

Rules

Types

- Connectivity
    - Junction – junction
    - Junction – edge
    - Edge – junction – edge
- Containment
- Structural Attachment



Rule Type

* Junction-junction connectivity
Junction-edge connectivity
* Containment
Structural attachment
* Edge-junction-edge connectivity

Esri confidential – contents subject to change

Rules

- 🟩 Junction - Junction
- ▧ Junction - Edge (foreign key)
- ■ System junction
- 🟧 Containment
- 🟨 Structural attachment
- 🟦 Junction - Edge
- ▩ Edge – Junction - Edge
- **N** New

From To	Device	Junction	Junction Object	Line	Edge Object	Assembly	Structure Junction	Structure Junc Obj	Structure Line	Structure Edge Obj	Structure Boundary
Device	🟩 🟧N	🟩 🟧N	🟩 🟧N	🟦 🟧N	▧ 🟧N	🟧	🟨 🟧	🟨N 🟧N	🟧	🟧N	🟧
Junction	🟩 🟧N	🟩 🟧	🟩 🟧N	🟦 🟧N	▧ 🟧N	🟧	🟨 🟧	🟨N 🟧N	🟧	🟧N	🟧
Junction Object	🟩 🟧	🟩 🟧N	🟩 🟧	🟧N	▧ 🟧N	🟧	🟨N 🟧	🟨N 🟧N	🟧N	🟧N	🟧
Line	🟦 🟧N	🟦 🟧N	🟧N	▩ ■ 🟧	▩N 🟧	🟧	🟧	🟧N	🟧	🟧N	🟧
Edge Object	▧ 🟧	▧ 🟧N	▧ 🟧	▩N 🟧	▩N 🟧	🟧	🟧	🟧N	🟧	🟧N	🟧
Assembly						🟧	🟨 🟧	🟨N 🟧N			🟧
Structure Junction							🟩N 🟨 🟧	🟩N 🟨N 🟧N	🟦	▧N	🟧
Structure Junc Obj							🟩N 🟨N 🟧N	🟩N 🟨N 🟧N		▧N	🟧N
Structure Line							🟦 🟧	🟧N	▩ ■ 🟧	▩N 🟧N	🟧
Structure Edge Obj							▧N 🟧N	▧N 🟧N	▩N 🟧N	▩N 🟧N	🟧N
Structure Boundary							🟧	🟧N			🟧

Association roles

Used to support the rules

	Structure	Container	None
Structure Junction	X	X	X
Structure Boundary		X	X
Structure Junction Object	X	X	X
Structure Line		X	X
Structure Edge Obj		X	X

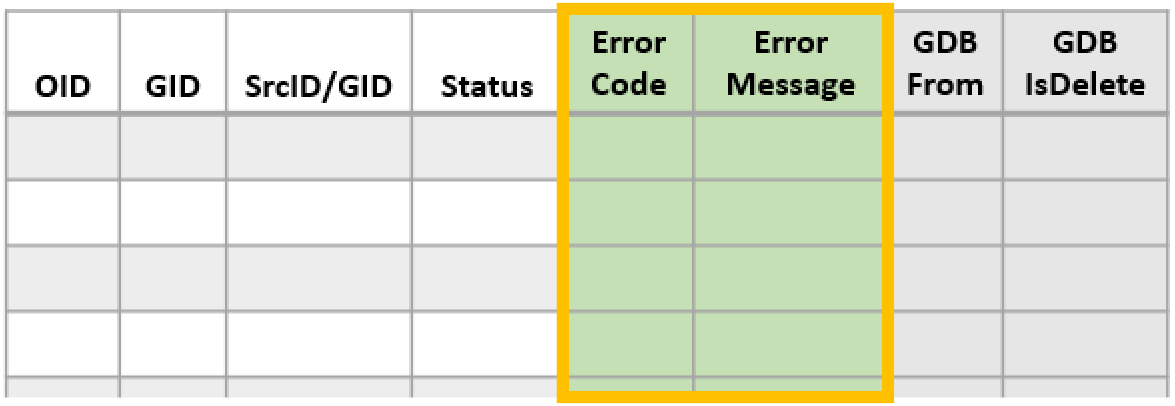
	Structure	Container	None
Device		X	X
Junction		X	X
Assembly		X	X
Junction Object		X	X
Line		X	X
Edge Object		X	X

Error model

Error model

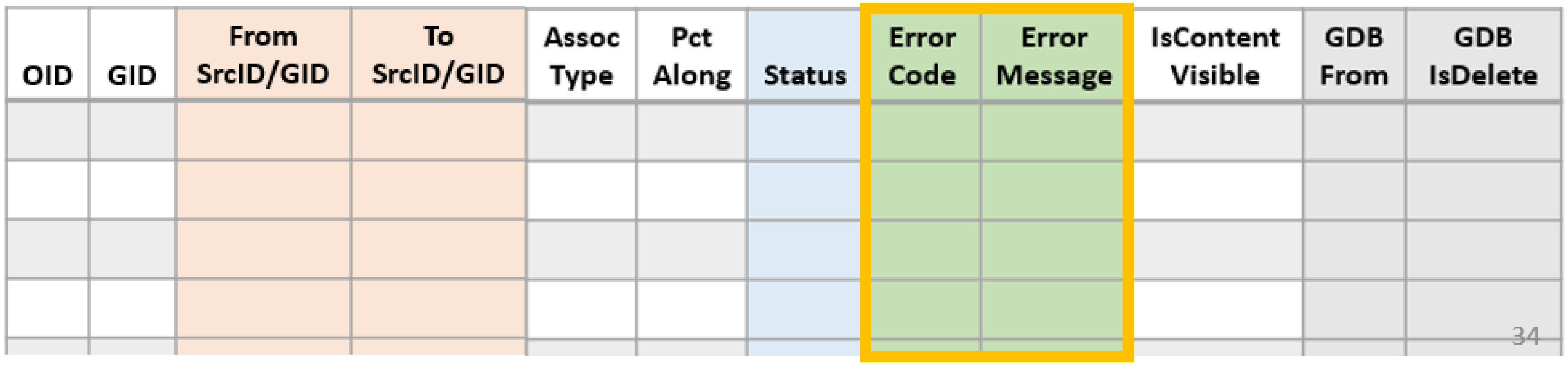
- For performance and scalability, the existing error model is being revised
  - We will remove the point, line, and polygon error tables
  - Errors will be persisted in the DirtyAreas and Associations tables using existing codes and messages
    - ErrorCode
    - ErrorMessage

DirtyAreas



OID	GID	SrcID/GID	Status	Error Code	Error Message	GDB From	GDB IsDelete

Associations



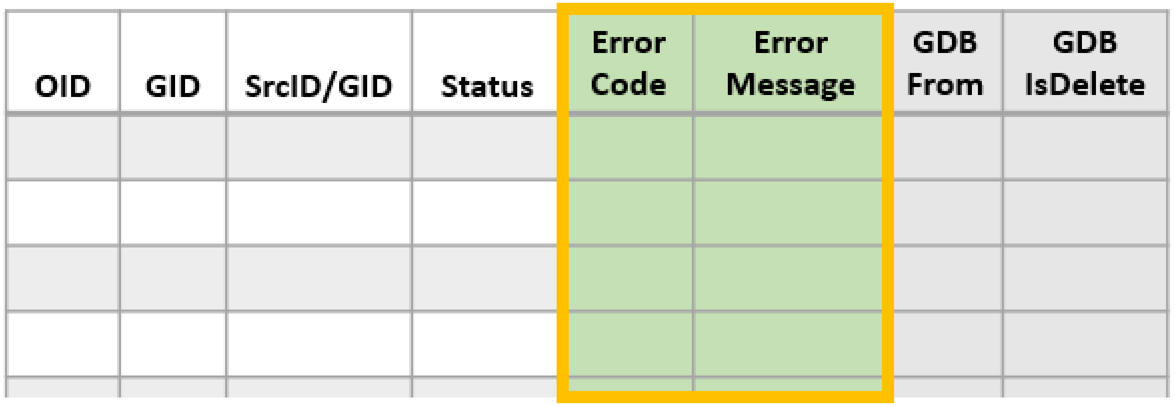
OID	GID	From SrcID/GID	To SrcID/GID	Assoc Type	Pct Along	Status	Error Code	Error Message	IsContent Visible	GDB From	GDB IsDelete

Error model

spatial errors

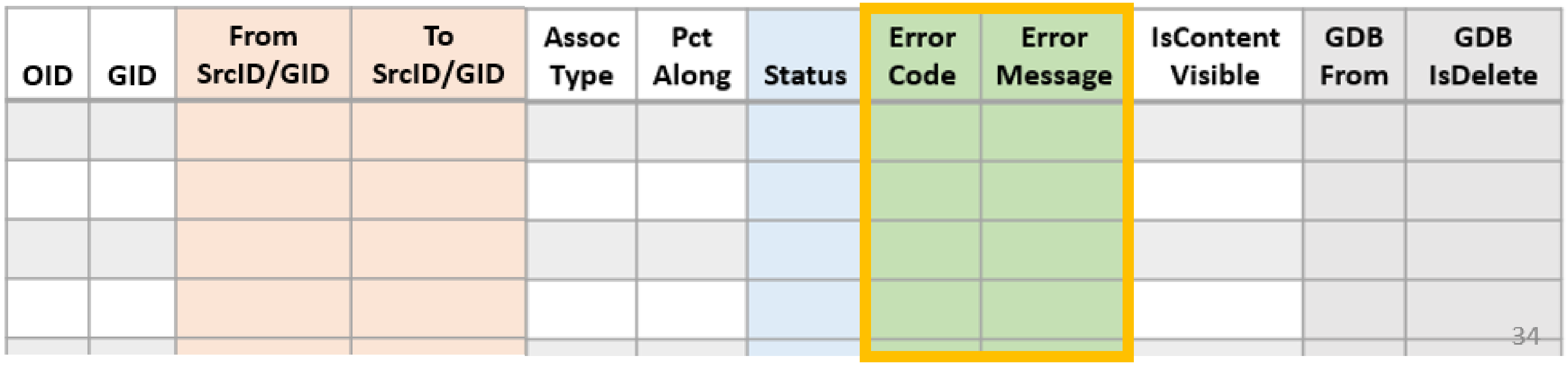
- Written to the dirty area table
- Bit encoded value

DirtyAreas



OID	GID	SrcID/GID	Status	Error Code	Error Message	GDB From	GDB IsDelete

Associations



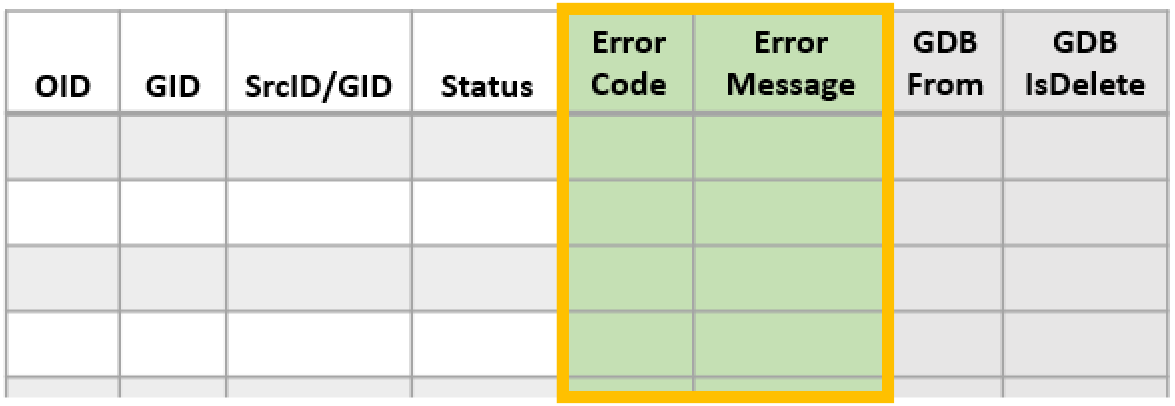
OID	GID	From SrcID/GID	To SrcID/GID	Assoc Type	Pct Along	Status	Error Code	Error Message	IsContent Visible	GDB From	GDB IsDelete

Error model

nonspatial errors

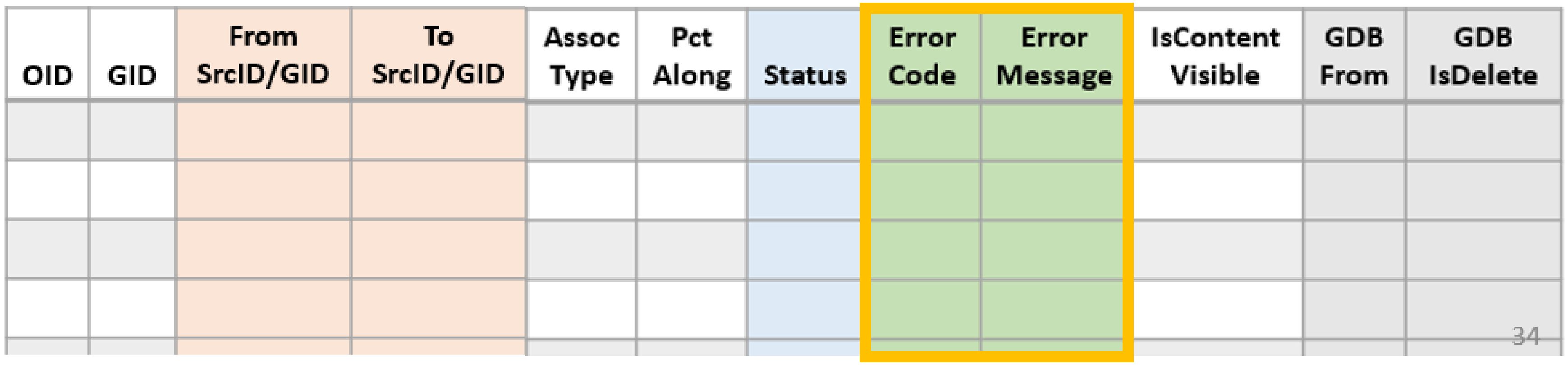
- Written to association table; bit encoded value
- When object is in error, write to all associations in which that object is a child
- When association itself is in error write to only the errored association

DirtyAreas



OID	GID	SrcID/GID	Status	Error Code	Error Message	GDB From	GDB IsDelete

Associations



OID	GID	From SrcID/GID	To SrcID/GID	Assoc Type	Pct Along	Status	Error Code	Error Message	IsContent Visible	GDB From	GDB IsDelete

Error model

- When an edge or junction objects has an error identified during validate
  - We will not have an error object table
  - Insert/update a dirty area and add 5 (ObjectError) to the DirtyArea.Status
  - Update the association record, setting the object error code in the Association.ErrorCode
- A tool for the Pro UI will be developed to support common user workflows when managing errors
- A tool will be needed to resynthesize the physical error tables (standalone) to support QA/QC workflows (supported by partial validate)

Error model

- When cleaning up the error objects, the dirty area traverser will also expose the capability to identify all the unmodified but still existing object errors (ErrorsNotModified(), DESC only)
  - This is used to control resetting the DirtyArea.Status(5: ObjectError) bit

- In the DirtyArea.Status, the transition from TTTXXX to FFFXXX indicates that a validation has occurred
  - This is useful to allow a user to understand when a validate has occurred and the index is in a locally consistent state

Esri confidential – contents subject to change

Error model

schema version 3

- Errors can be generated in:
  - Enable Network Topology
  - Validate Network Topology
  - Update Subnetwork
- Errors are stored in internal UN error tables
  - Point
  - Line
  - Polygon
- To view errors you must look at each table individually

Error model

schema version 4

- Introduced nonspatial objects (edge and junction objects, and their errors)
- Consolidated the dirty and error information together on same table
  - Dirty Area Table – Spatial Features
  - Association Table – Nonspatial Objects
- Removed the old error tables

How are errors stored?

- Both the DirtyArea table and Association table added fields:
  - ErrorCode(long)
  - ErrorMessage(String)
- Nonspatial Errors are stored on the association records in which it is a child (to side of association)
- ErrorCode is a bit encoded value
  - Merges all errors to 1 location
    - Dirty Area (Spatial)
    - Association (Nonspatial)
  - (Schema3) ErrorCode => (Schema4) 2^ErrorCode
- ErrorMessage contains only contextual information on the errors

How do I view my errors?

- Dirty areas will by symbolized based on error type
- The ErrorCode shows all errors for that feature/object
- The dirty area/association row and a static lookup table is all that is needed to see all error messages
- A popup will be provided either as part of core or as a sample to display error information

Error Code Example:

Row Value: Decimal: 10

Bit encoding: 0b1010

Errors: 1, 3

How do I view my errors?

- The ErrorCode shows all errors for that feature/object
- The dirty area/association row and a static lookup table is all that is needed to see all error messages.
- A popup can be easily written using Arcade to allow uses to see their errors again.

Error Code Example:

Row Value: Decimal: 10

Bit encoding: 0b1010

Errors: 1, 3

Error model

summary

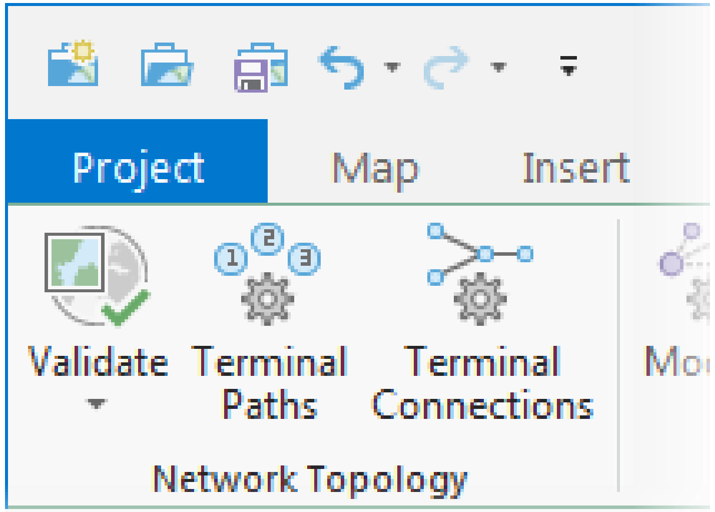
- When an object is found in error, the error is marked on all associations for which it is child
- When the error is on the association itself, the error is marked only on the association in error – this is independent of whether or not the child object has two or more parents
- If an error is on a single feature, the error is marked only on the dirty area – this is independent of any associations present
- If an error is between two features:
  - The two dirty areas associated with the two features are marked
  - If there is an association between the two features, the association is marked
- If an error is between a feature and an associated object, the association is marked and the dirty area is marked
  - It may also be the case that the object is the parent in the association

New errors

- To accommodate midspan junction object connectivity; midspan nonspatial errors at the same % along can be identified
- Only accept $0 < x < 100\%$ percent along midspan

Build algorithm

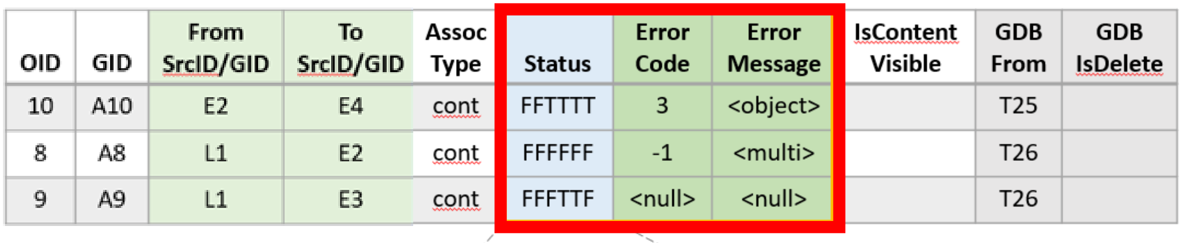
Validate network topology



Build

summary of changes

- Edits and errors to the nonspatial objects will be recorded in the associations table during editing by setting a status field
  - No physical error objects are generated (optionally this can be turned back on)
  - Errors on the nonspatial object will be reported on the associations in which it is a **child**
  - Errors on the association itself will be reported on the associations
  - First error gets specific error code and message
    - Two or more errors get error code -1 (multiple errors found)
- At the end of build, updates will be made to the association's STATUS field



OID	GID	From SrcID/GID	To SrcID/GID	Assoc Type	Status	Error Code	Error Message	IsContent Visible	GDB From	GDB IsDelete
10	A10	E2	E4	cont	FFTTTT	3	<object>		T25	
8	A8	L1	E2	cont	FFFFFF	-1	<multi>		T26	
9	A9	L1	E3	cont	FFFTTF	<null>	<null>		T26	

Status bits:
1 – Deleted association
2 – Deleted FROM feature/object
3 – Deleted TO feature/object
4 – Modified association
5 – Modified FROM feature/object
6 – Modified TO feature/object

When are nonspatial objects validated?

- From Ribbon/GP Tool (standard full/partial validate)
  - All modified nonspatial objects that are traversable in the associations table from the dirty areas will be consumed
    - This is done to keep the performance of validation – we will not consume unreachable edge and junction objects (as is done with the network dataset)

- New Validate Selected Objects GP tool
  - Given a selection of features/objects, validate them
  - Scorched earth policy, reestablish index for selection, no eid reuse

Error codes and messages

Error code	Error message
8	Invalid connectivity - No junction edge rule
26	During update subnetwork, invalid device feature which is not in the tier definition was discovered
27	During update subnetwork, invalid parent subnetwork discovered
28	During update subnetwork, disjoint subnetwork discovered
29	During update subnetwork, inconsistent subnetwork name on multiple subnetwork controllers in the same subnetwork discovered
30	During update subnetwork, inconsistent subnetwork name on multiple parent subnetwork controllers in the same subnetwork discovered
31	Association record is missing one of the endpoints.
34	Feature or object in unsupported containment relationship
35	Feature or object in unsupported structural attachment relationship
36	Self-intersecting edge object
37	Stacked junction objects

Additional tools for detecting unlocatable objects

overview

- Subnetwork Manager/Trace can optionally detect anomalous conditions
  - Consistent -> no dirty areas/associations
  - Unlocatable -> no derivable spatial location
  - Unparented network elements
  - Unparented network elements whose objects have been deleted
- Spatial containment check (selection set or whole)
  - For each object determine if it is within a spatial cover
  - Very expensive calculation for a full dataset
- Identify floating edge and junction objects
  - AssocStatus = 0 on junction/edge tables

Additional tools for detecting unlocatable objects

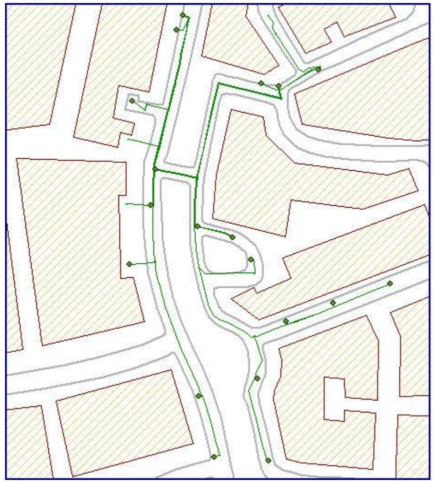
overview

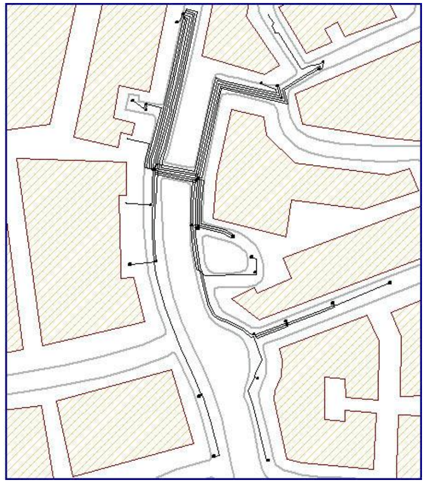
- Build on selection set (sourceIDs, globalIDs)
  - Takes as an input a selection of source ids and global ids rather than rebuild extent
  - Uses the association traversal object to perform downward and upward traversal
  - Uses a scorched earth element management policy

Tracing

Motivation

- Support urban underground (electric)
  - Avoid the need to digitize a large number of features (expensive) while still providing the ability to trace based on network attributes (e.g., phase)
  - Address gap of not having layout tools for auto-generating underground features





- Lay the ground work for supporting telco/fiber

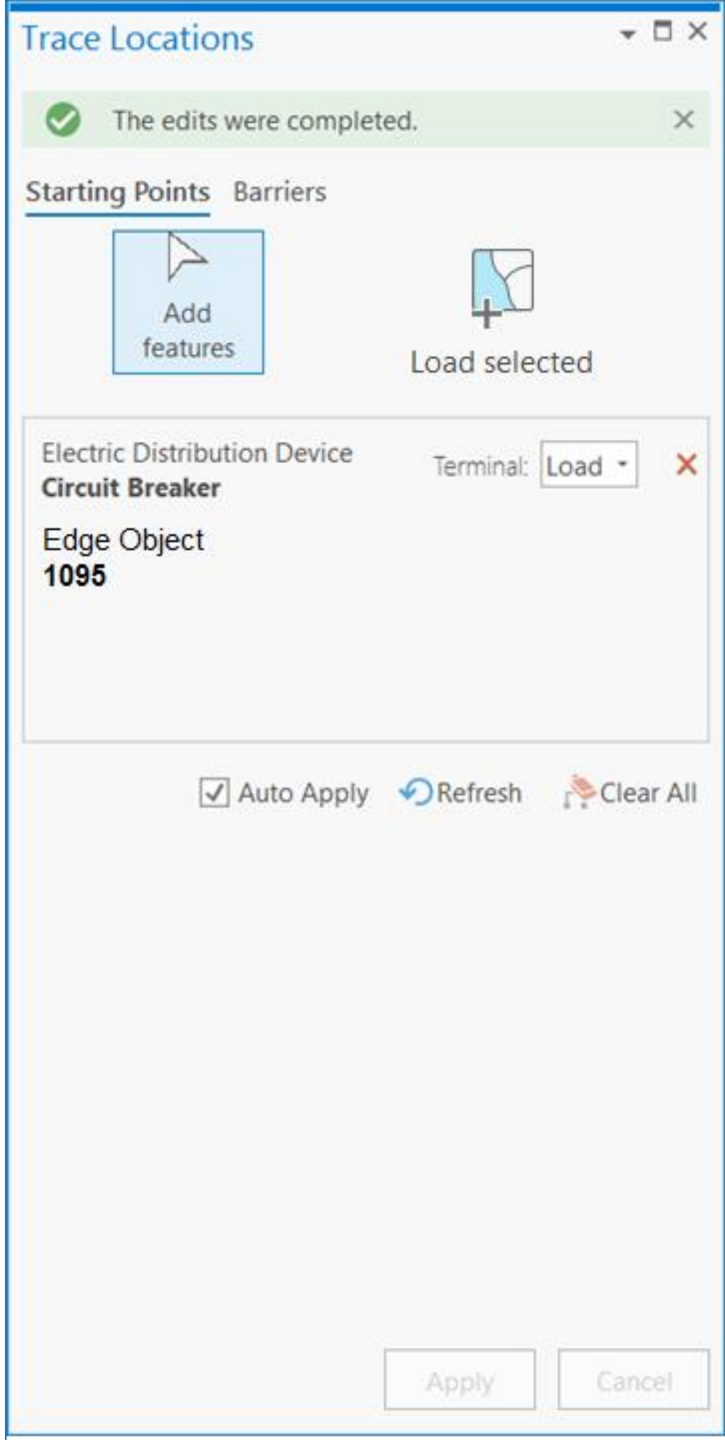
Changes

- Trace locations pane
- Selection
- Validate consistency
- Unlocatable objects

Esri confidential – contents subject to change

Trace locations pane

- Allow rows from standalone tables for junction and edge objects to be used as trace locations
  - Select row in table and then use “Load selected” command
- Graphics for nonspatial trace locations will be based on container geometry
- Change label
  - Load selected features → Load

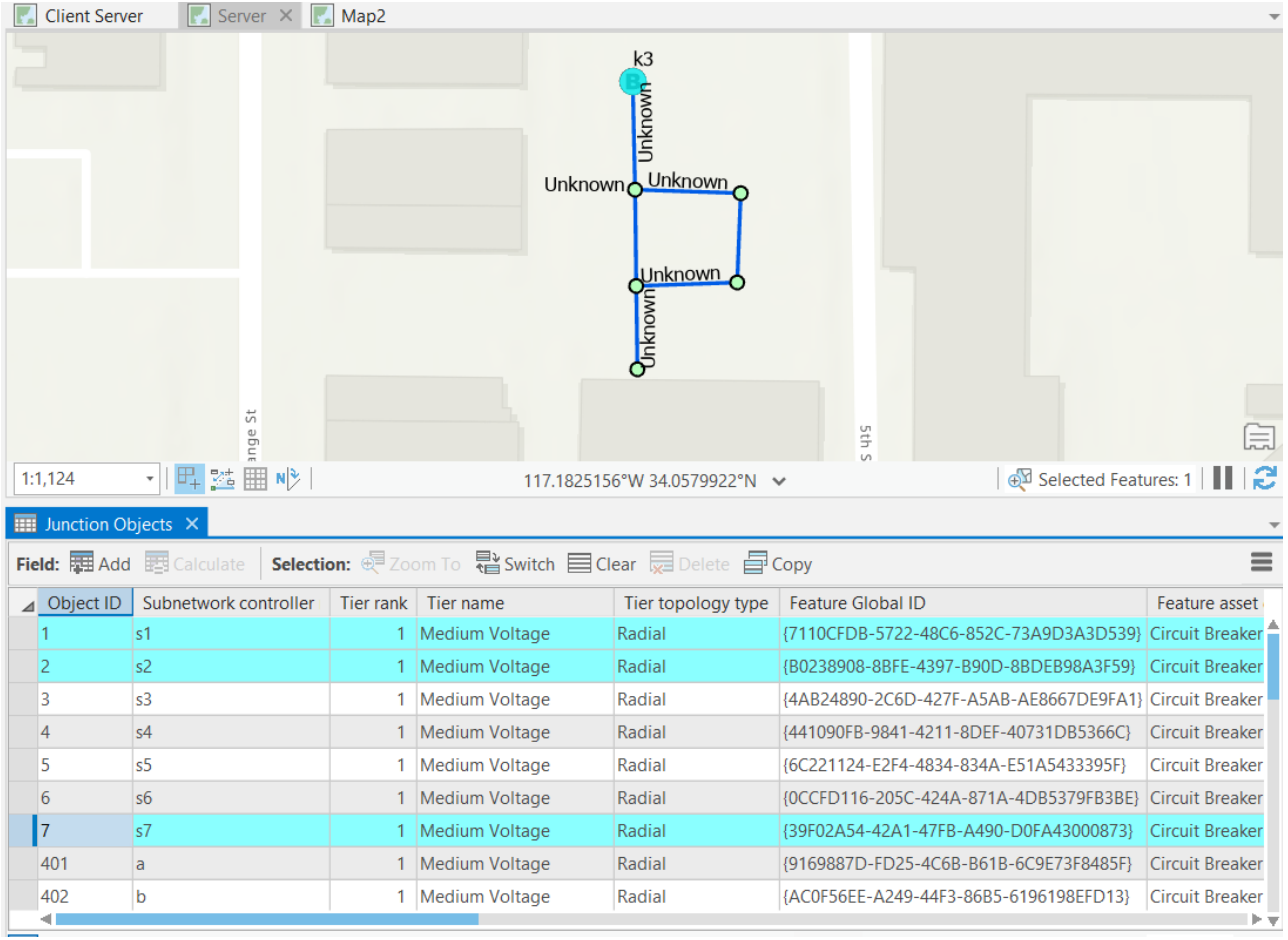


Electric Distribution Device
Circuit Breaker

Edge Object
1095

Trace results selection

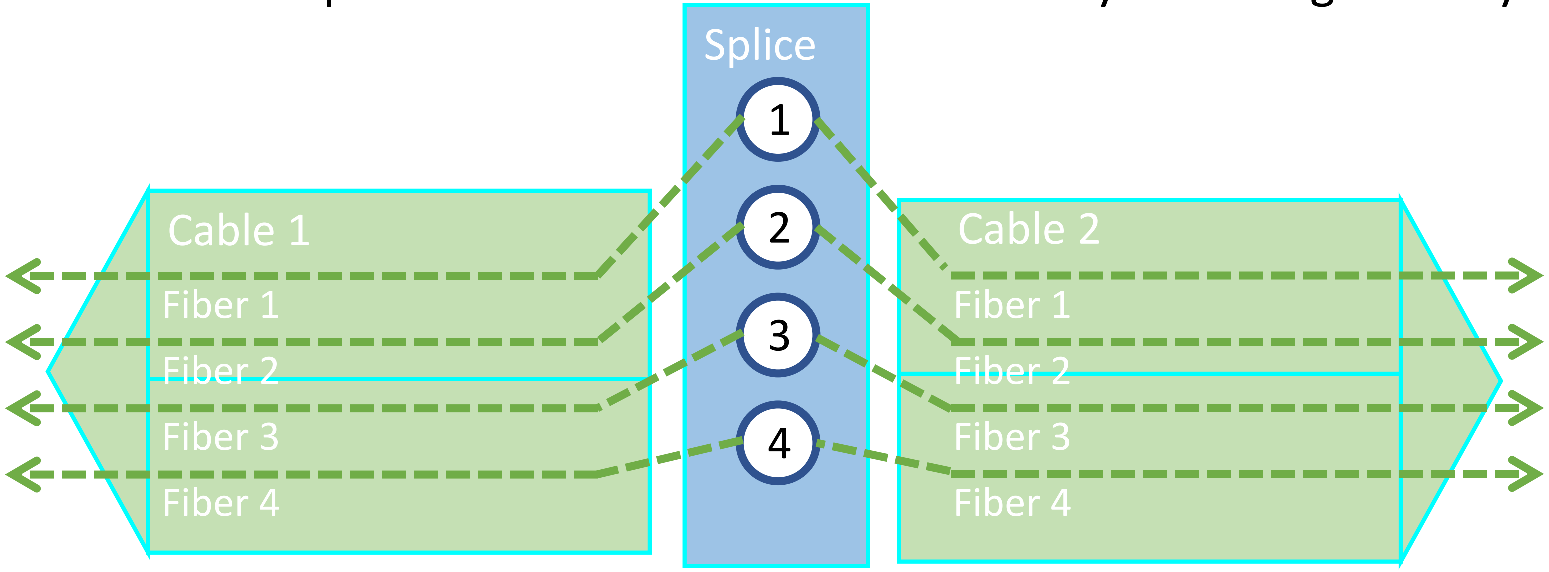
- Support trace results selection in standalone tables for junction and edge objects

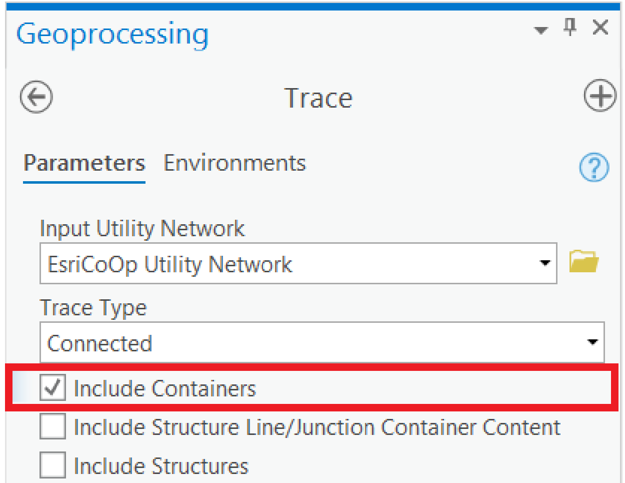


Object ID	Subnetwork controller	Tier rank	Tier name	Tier topology type	Feature Global ID	Feature asset
1	s1	1	Medium Voltage	Radial	{7110CFDB-5722-48C6-852C-73A9D3A3D539}	Circuit Breaker
2	s2	1	Medium Voltage	Radial	{B0238908-8BFE-4397-B90D-8BDEB98A3F59}	Circuit Breaker
3	s3	1	Medium Voltage	Radial	{4AB24890-2C6D-427F-A5AB-AE8667DE9FA1}	Circuit Breaker
4	s4	1	Medium Voltage	Radial	{441090FB-9841-4211-8DEF-40731DB5366C}	Circuit Breaker
5	s5	1	Medium Voltage	Radial	{6C221124-E2F4-4834-834A-E51A5433395F}	Circuit Breaker
6	s6	1	Medium Voltage	Radial	{0CCFD116-205C-424A-871A-4DB5379FB3BE}	Circuit Breaker
7	s7	1	Medium Voltage	Radial	{39F02A54-42A1-47FB-A490-D0FA43000873}	Circuit Breaker
401	a	1	Medium Voltage	Radial	{9169887D-FD25-4C6B-B61B-6C9E73F8485F}	Circuit Breaker
402	b	1	Medium Voltage	Radial	{AC0F56EE-A249-44F3-86B5-6196198EFD13}	Circuit Breaker

Viewing trace results in a map

- By selecting “Include Containers”, the trace framework will select all container features on the map for the trace results
  - Entire containment hierarchy will be traversed and all containers selected
  - Possible future enhancement: provide user with warning/info if results have nonspatial objects and include containers isn’t checked
  - Another possible future enhancement is to synthesize geometry for nonspatial objects





Input Utility Network
EsriCoOp Utility Network

Trace Type
Connected

☑ Include Containers
☐ Include Structure Line/Junction Container Content
☐ Include Structures

Validate consistency

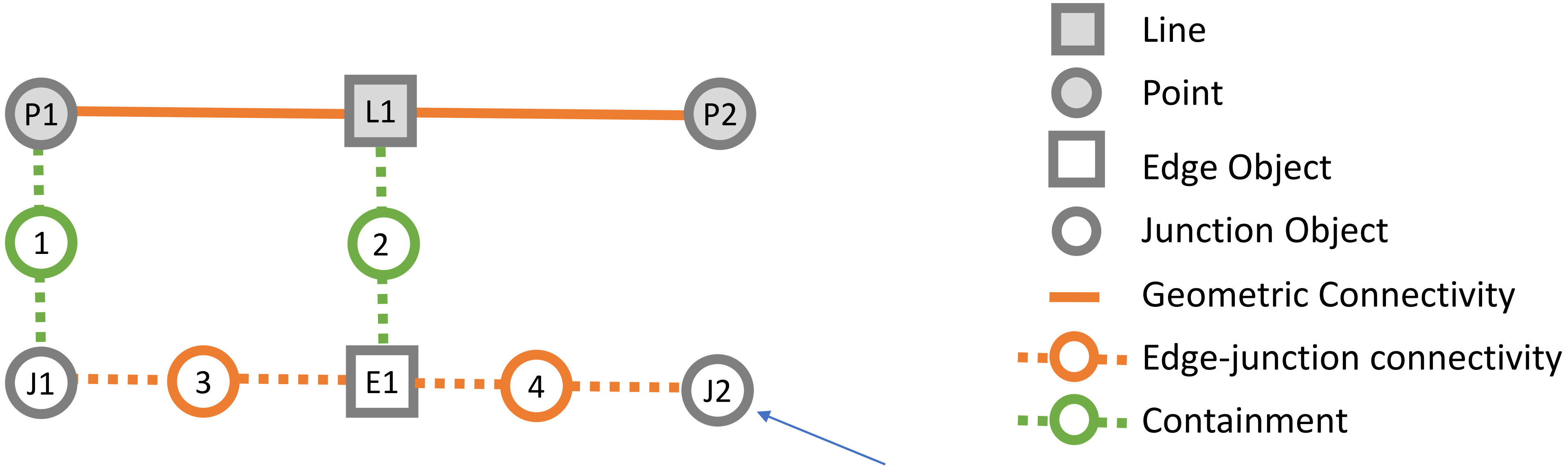
- Validate consistency will be augmented to support nonspatial objects

- Current behavior
  - If the global ID of any feature traced exists in the dirty areas table, isConsistent is set to false

- New behavior
  - isConsistent is set to false if either
    - If the global ID of any feature traced exists in the Dirty Areas table <u>or</u>
    - If the global ID of any nonspatial object has a dirty status bit set in the Associations table (i.e., the association, “from”, or “to” feature/object cannot be dirty or deleted)

Unlocatable objects

- Unlocatable objects are nonspatial objects that lack a transitive containment relationship back to a feature
- A new trace configuration option will be added to discover unlocatable objects
  - This can be used to clean problems associated with unlocatable objects



Junction object J2 is unlocatable because there is no transitive containment relationship back to a feature

Nonspatial urban underground examples

trace dense urban regions with nonspatial objects

- Ductbank and duct elements are dense and challenging to maintain across dense urban regions
  - Nonspatial objects will ease this burden by removing the need to digitize many features (which is expensive) and addressing the gap of not having layout tools for auto-generating underground features
  - Supporting starting points/barriers as well as reporting nonspatial trace results is important for supporting analysis on this type of model

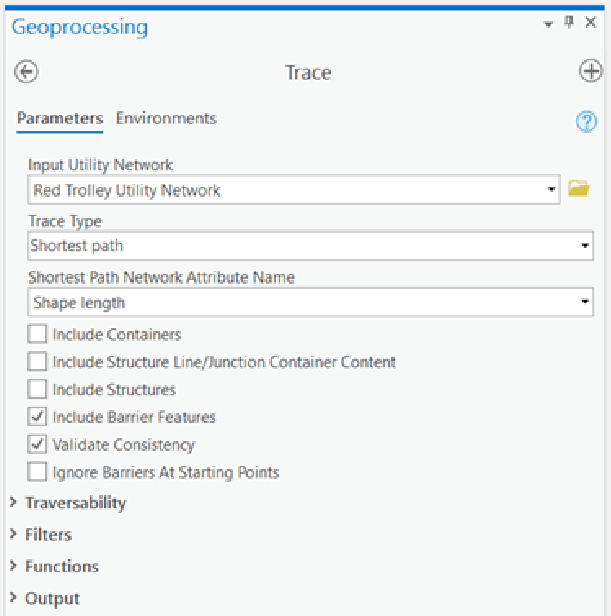
Underground Data

Duct Cells and Circuits Downtown

Nonspatial urban underground examples

shortest path based on availability

- Identify the shortest path between two starting points but only paths which have available capacity and support a specific type, such as medium voltage or 3”-12” distribution duct banks
  - Add condition barrier to shortest path trace to force the path to only traverse Duct Banks which have duct availability
  - Add an output condition to only return features with the category Medium Voltage Duct



Geoprocessing

Trace

Parameters Environments

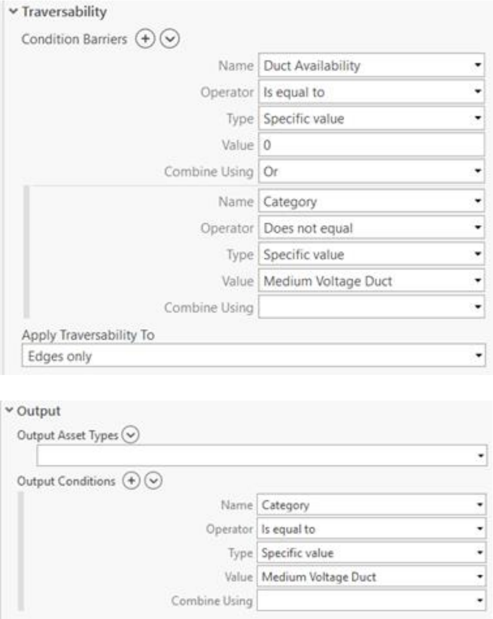
Input Utility Network: Red Trolley Utility Network

Trace Type: Shortest path

Shortest Path Network Attribute Name: Shape length

□ Include Containers
□ Include Structure Line/Junction Container Content
□ Include Structures
☑ Include Barrier Features
☑ Validate Consistency
□ Ignore Barriers At Starting Points

› Traversability
› Filters
› Functions
› Output



˅ Traversability

Condition Barriers

Name	Duct Availability
Operator	Is equal to
Type	Specific value
Value	0
Combine Using	Or
Name	Category
Operator	Does not equal
Type	Specific value
Value	Medium Voltage Duct
Combine Using	

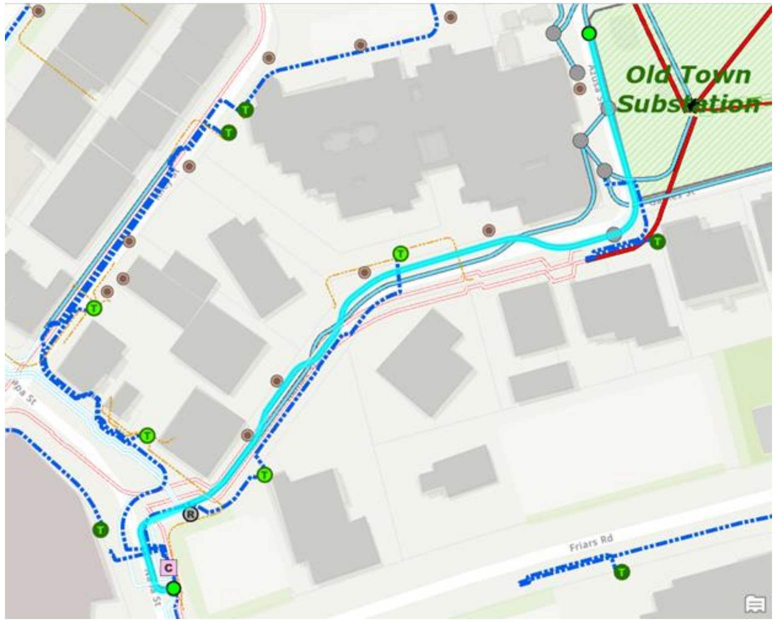
Apply Traversability To: Edges only

˅ Output

Output Asset Types

Output Conditions

Name	Category
Operator	Is equal to
Type	Specific value
Value	Medium Voltage Duct
Combine Using	



Analytic capabilities for telco/fiber

Analytic Capability	Notes
Trace to determine businesses or residences within a certain proximity of a cable (serviceability)	Connected trace with function barrier and output filters
Trace to determine amount of cable (footage) with a government jurisdiction for tax reporting purposes	Connected trace with a function for adding cable length
Trace to determine location of a cable cut based on specified distance or hops	Connected trace with function barrier
Trace downstream from the headend (ISP transmission equipment port level, distribution frame, etc.) to show all connected customers (commercial – residential) and OSP devices (SAIs, FIC, splices, terminals, etc.)	Downstream trace with output filters
Trace upstream from customer or device to the headend	Upstream trace
Trace a copper pair to determine the length of loop makeup and connected devices (often used for DSL pre-qualification)	Loops trace with function for adding loop length
Given information from OTDR (optical time domain reflectometer), identify the location of a fault	Connected trace with function barrier and include barriers. Result is where the trace stops.
Trace to determine cables with unused fibers or copper pairs	Trace based on network attribute that indicates whether fiber is used
Trace to determine network diversity in and out of a central office or customer location (redundancy)	*Not Supported*

Changes
summary

- Issue 1: Modify the trace locations pane to allow junction and edge objects to be used as trace locations
- Issue 2: Enhance the trace GP tool so that trace results in standalone tables for junction and edge objects can be selected
- Issue 3: Augment validate consistency to support nonspatial objects
- Issue 4: Provide option to detect unlocatable objects

Subnetwork management

Changes

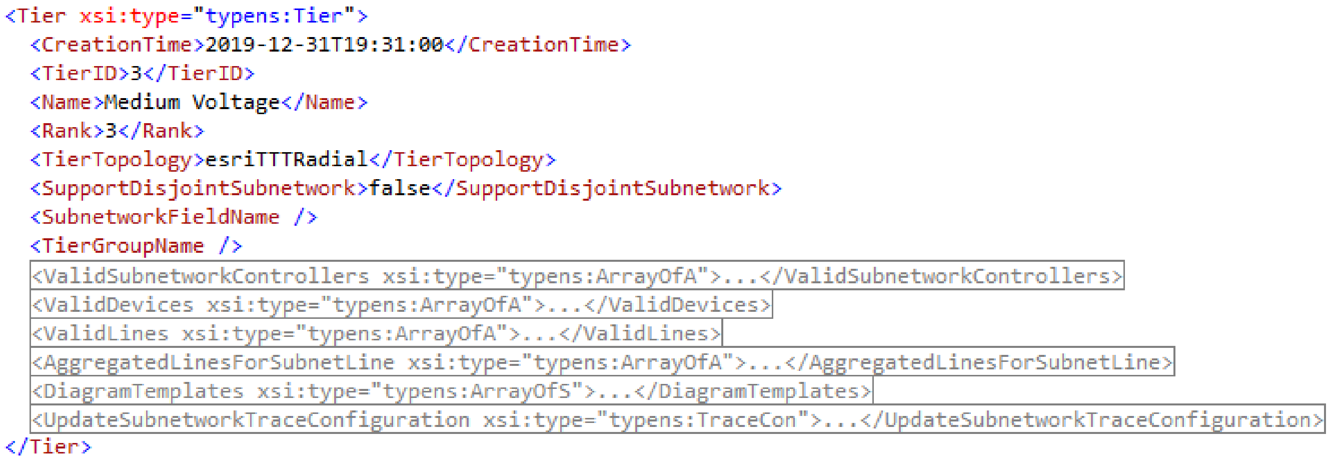
- Tier definition
  - Add definition for valid items
  - Add new options for *Update Subnetwork*

- Additional changes in Pro 2.6

Tier definition

add definitions for valid items

- Add setting for the following three to detect invalid AG/AT in those source classes during Update Subnetwork
  - Valid Junctions
  - Valid Junction Objects
  - Valid Edge Objects



```xml
<Tier xsi:type="typens:Tier">
  <CreationTime>2019-12-31T19:31:00</CreationTime>
  <TierID>3</TierID>
  <Name>Medium Voltage</Name>
  <Rank>3</Rank>
  <TierTopology>esriTTTRadial</TierTopology>
  <SupportDisjointSubnetwork>false</SupportDisjointSubnetwork>
  <SubnetworkFieldName />
  <TierGroupName />
  <ValidSubnetworkControllers xsi:type="typens:ArrayOfA">...</ValidSubnetworkControllers>
  <ValidDevices xsi:type="typens:ArrayOfA">...</ValidDevices>
  <ValidLines xsi:type="typens:ArrayOfA">...</ValidLines>
  <AggregatedLinesForSubnetLine xsi:type="typens:ArrayOfA">...</AggregatedLinesForSubnetLine>
  <DiagramTemplates xsi:type="typens:ArrayOfS">...</DiagramTemplates>
  <UpdateSubnetworkTraceConfiguration xsi:type="typens:TraceCon">...</UpdateSubnetworkTraceConfiguration>
</Tier>
```

- Modify Set Subnetwork Definition GP tool accordingly

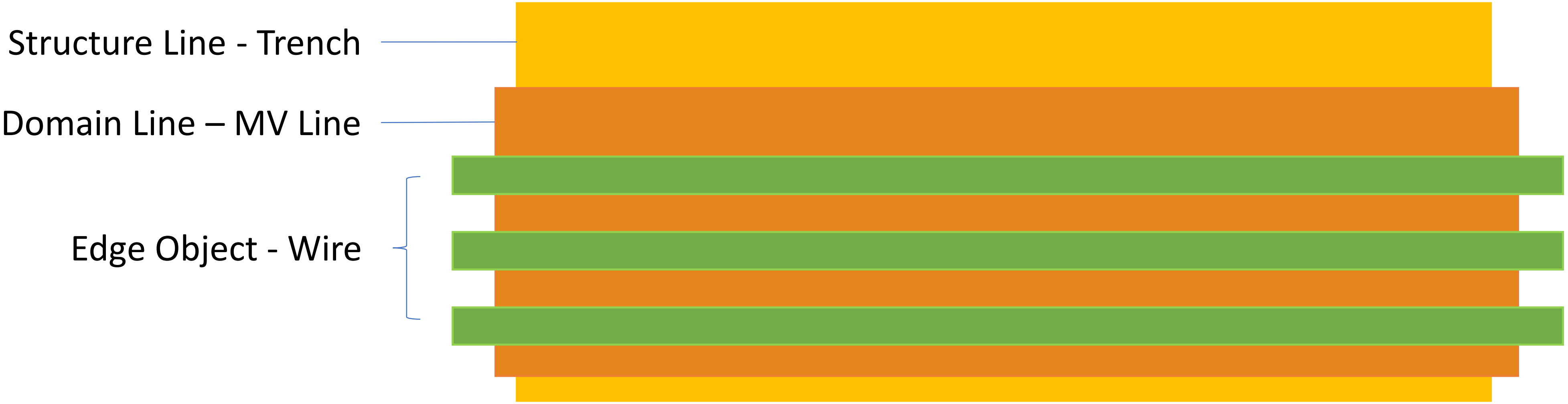
- Show added properties in utility network properties page

- Upgrade will set the new ValidJunctions to “all”; users will then reconfigure as appropriate

Update subnetwork

option: update containers in domain network classes

- Multiple levels of containment, 1 in domain network, and 1 in structure network
- A trace on a wire requesting the return of “containers” will give back the line and the trench (all levels)



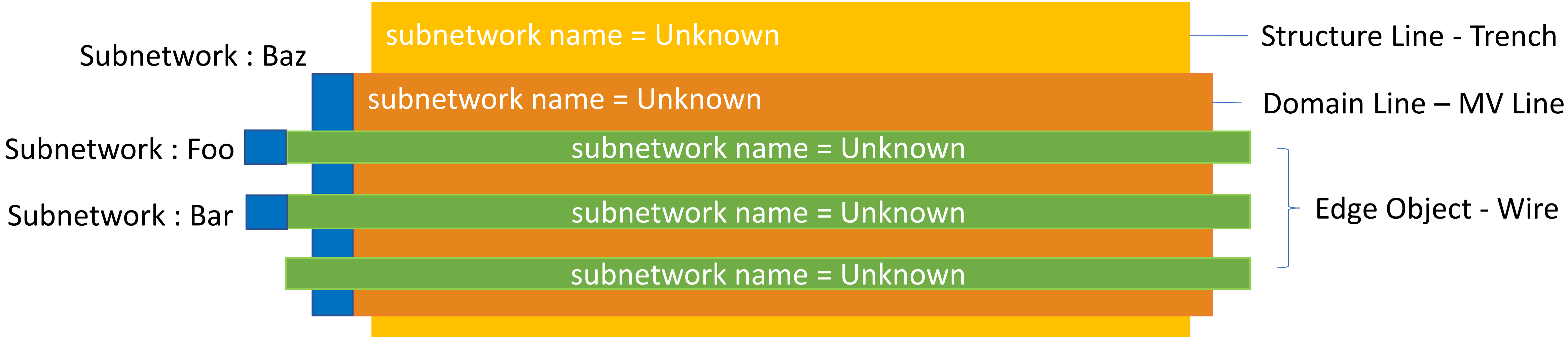
Wires are contained in Line, Line is contained in Trench

Note: Domain line class is allowed to be a container from 2.5

Update subnetwork

option: update containers in domain network classes

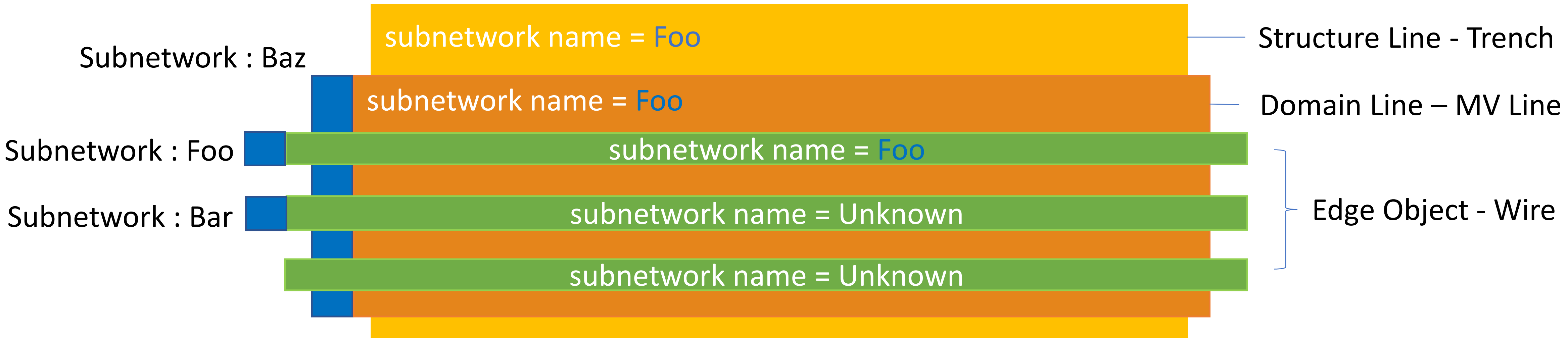
- Let's say we have three different subnetworks
    - Two, Foo and Bar, at the lowest level
    - One, Baz, at the second lowest level
- Subnetwork name is Unknown for all these linear features/objects at the moment


Subnetwork : Baz
Subnetwork : Foo
Subnetwork : Bar
subnetwork name = Unknown
subnetwork name = Unknown
subnetwork name = Unknown
subnetwork name = Unknown
subnetwork name = Unknown
Structure Line - Trench
Domain Line – MV Line
Edge Object - Wire

Update subnetwork

option: update containers in domain network classes

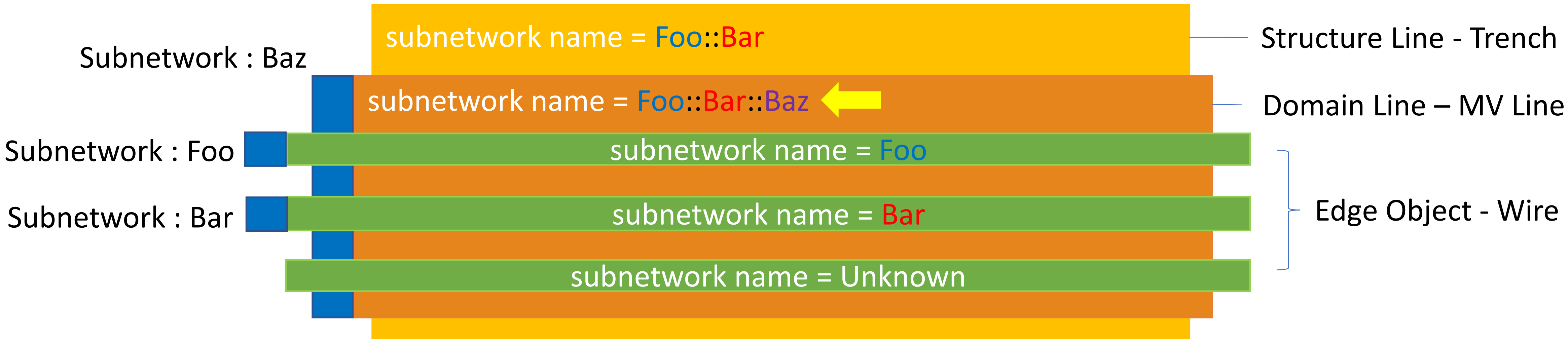
- Update Subnetwork on subnetwork Foo



Update subnetwork

option: update containers in domain network classes

- Update Subnetwork on subnetwork Baz

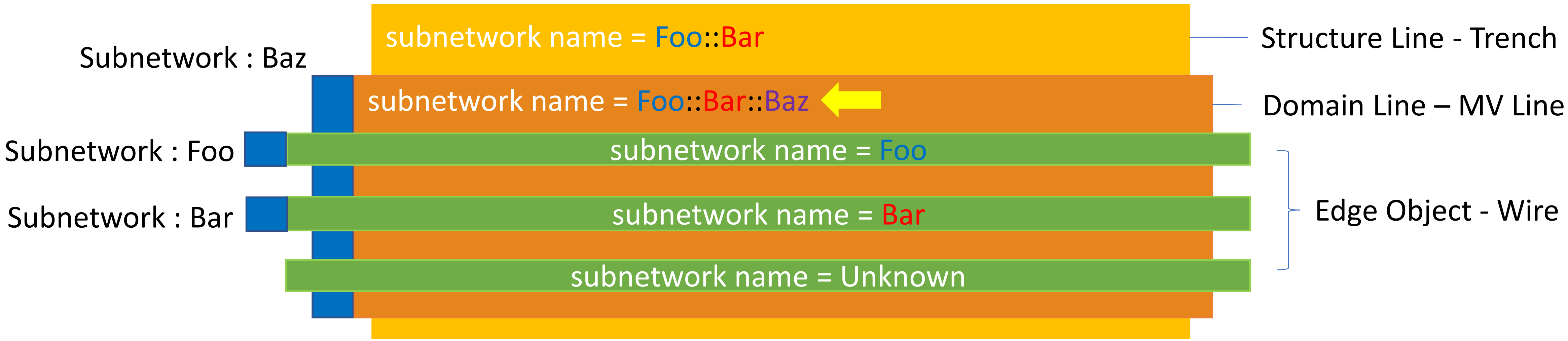


Requires a new mechanism to specify which Domain features returned from trace are *containers*
- new API needs to be exposed in trace framework

Update subnetwork

option: update containers in domain network classes

- Update Subnetwork on subnetwork Baz



Requires a new mechanism to specify which Domain features returned from trace are *containers*
- new API needs to be exposed in trace framework

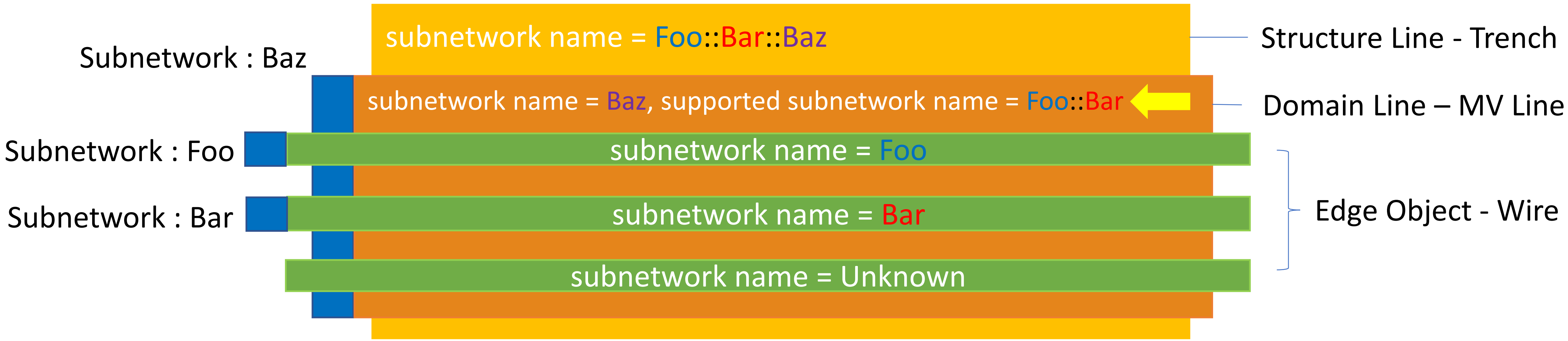
Update subnetwork

concern: subnetwork name field is overloaded at V3

Q1. Can users immediately know which subnetwork is the source of MV line?

No, subnetwork name field is overloaded

→ add a new field named SupportedSubnetworkName to disambiguate this as shown below



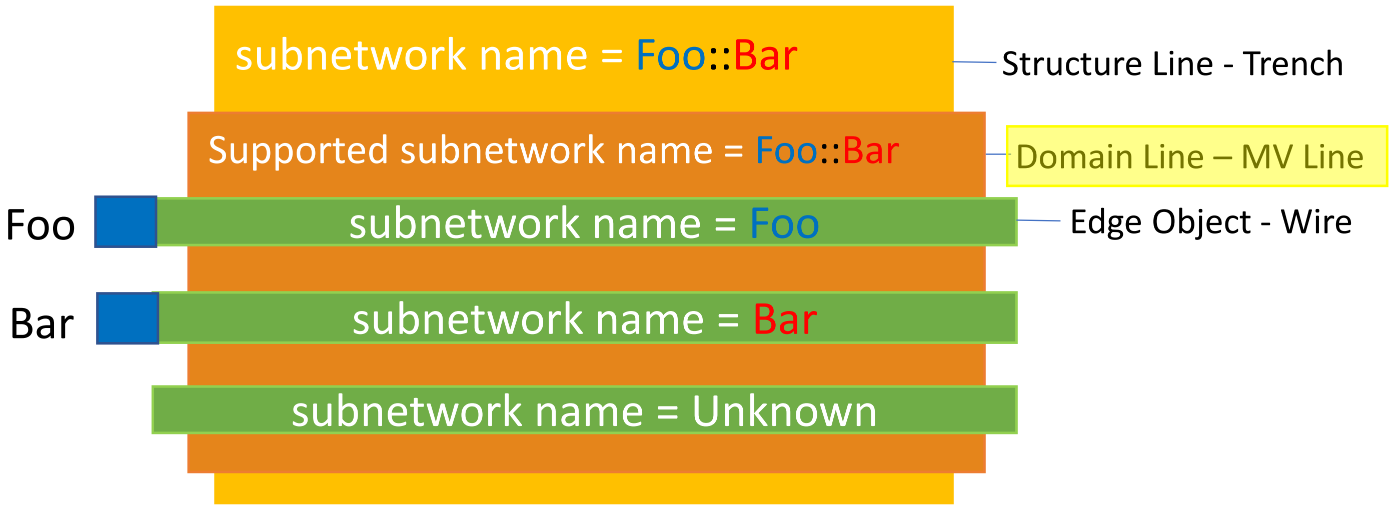
Note, we don't add SupportedSubnetworkName field to structure classes because they always *supports* subnetwork, never *participate in* a subnetwork – we change only the alias name to `Supported subnetwork name`

Update subnetwork

concern: supported subnetwork name field on containers

Q2. What would happen if we have containment levels like shown below on the right side in telco?

The supported subnetwork name field on containers in domain network classes would easily be broken – too many subnetworks in containers



subnetwork name = Foo::Bar

Structure Line - Trench

Supported subnetwork name = Foo::Bar

Domain Line – MV Line

Foo

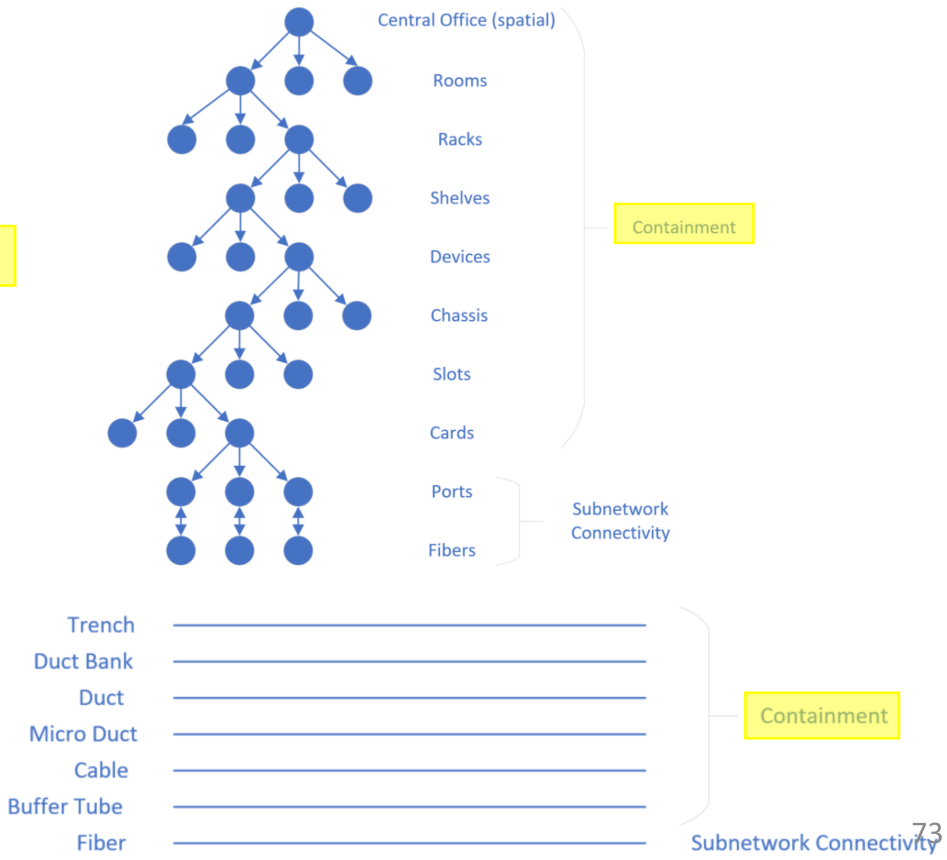
subnetwork name = Foo

Edge Object - Wire

Bar

subnetwork name = Bar

subnetwork name = Unknown



Esri confidential – contents subject to change

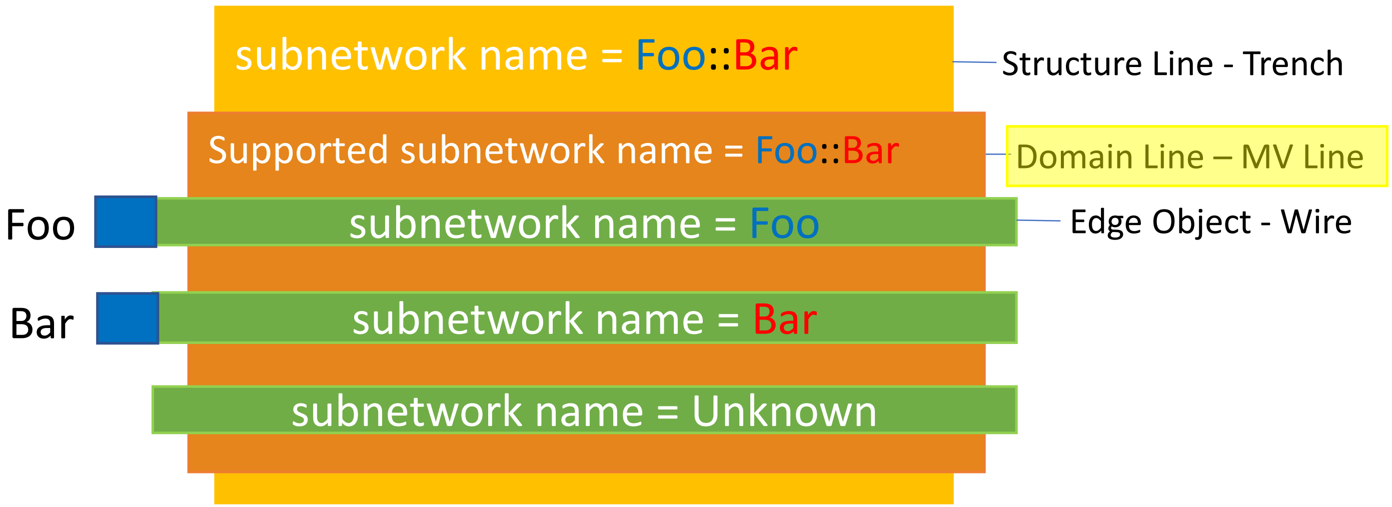
Update subnetwork

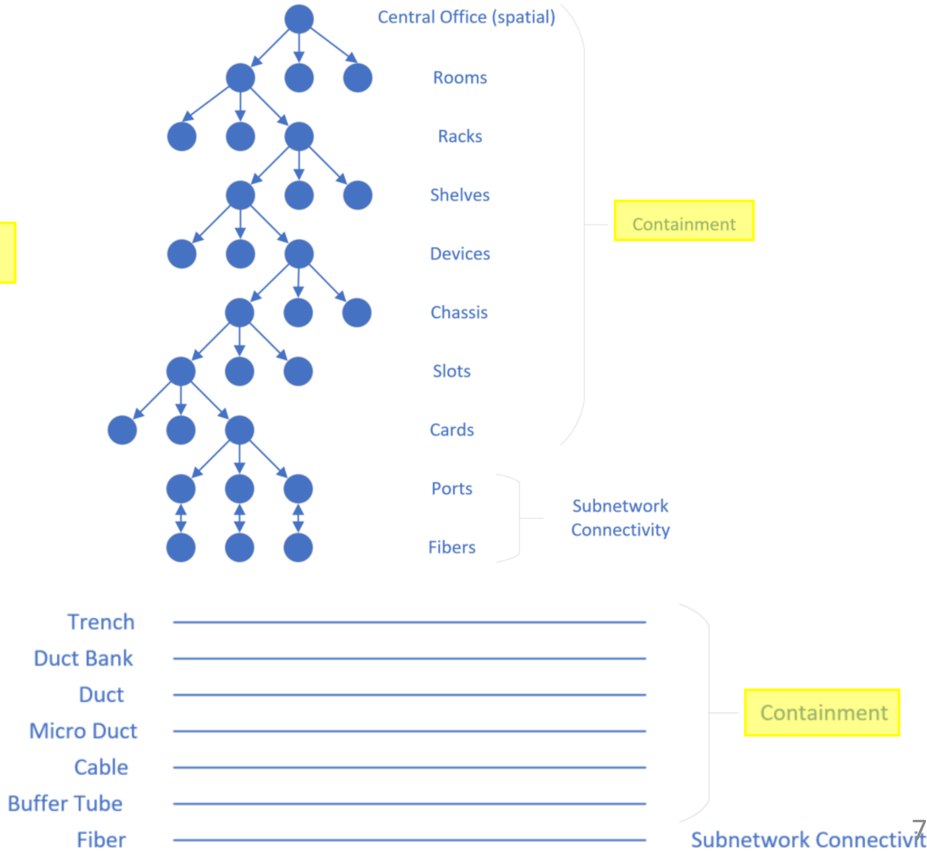
option: configure updating container features

Q3. Would telco companies really care the concatenated subnetwork names in container classes?

Maybe not

→ provide an option to not update container features in domain network classes





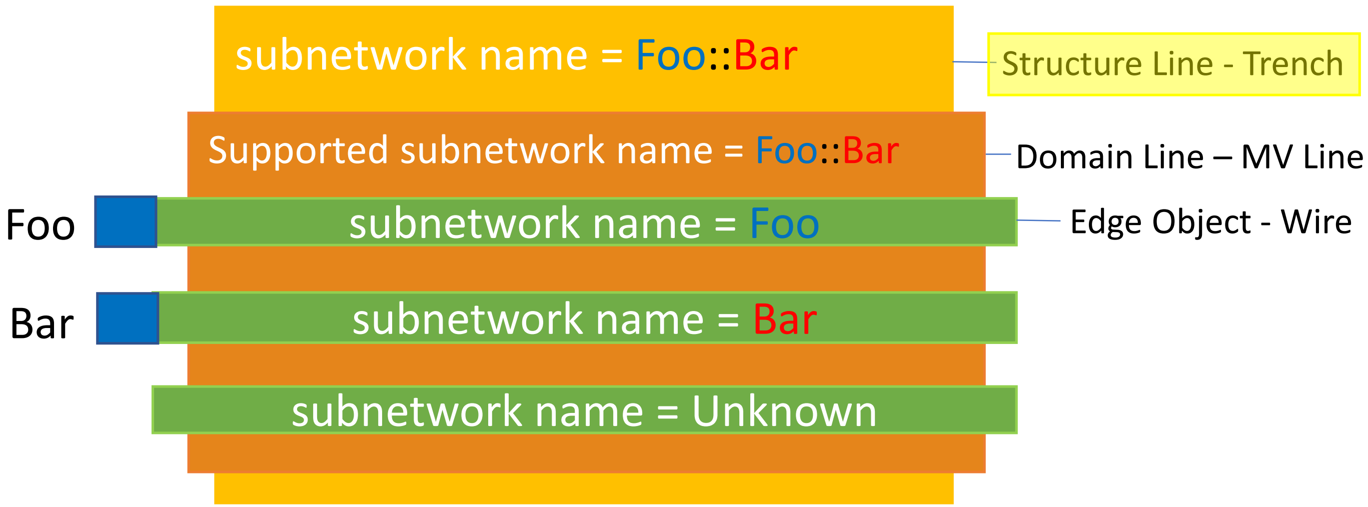
Update subnetwork

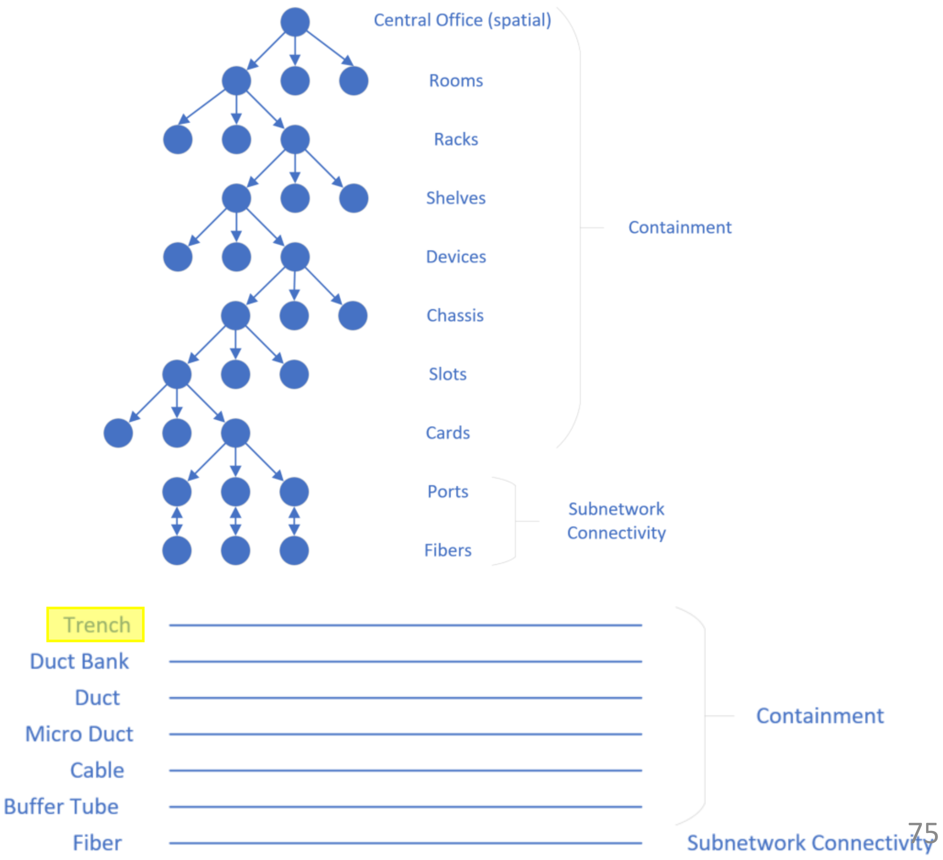
option: configure updating structure features

Q4. If that's the case for containers in domain network classes, what about structure classes?

Telco companies might not care about subnetwork names in structure class as well

→ provide an option to not update structure features





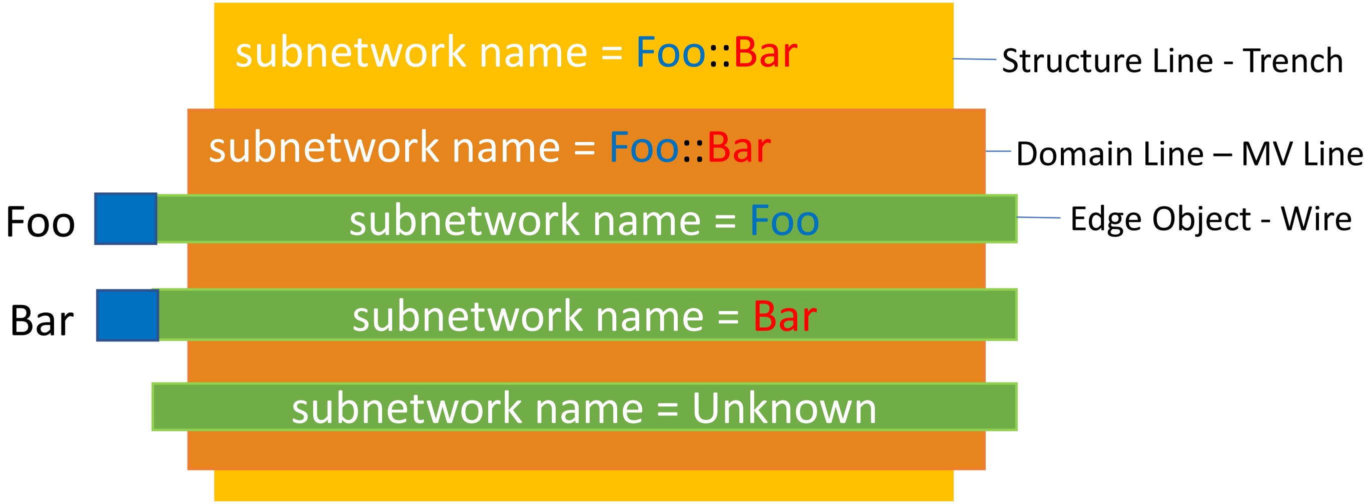
Update subnetwork

option: relax ‘valid lines’ and ‘aggregated lines for subnetLine’

Q5. Would telco companies really care the geometry of subnetworks at the bottom of containment level?

Maybe not, also now subnetworks could be totally nonspatial

→ relax ‘valid lines’ and ‘aggregated lines for subnetLine’ to optional parameter in gp tool



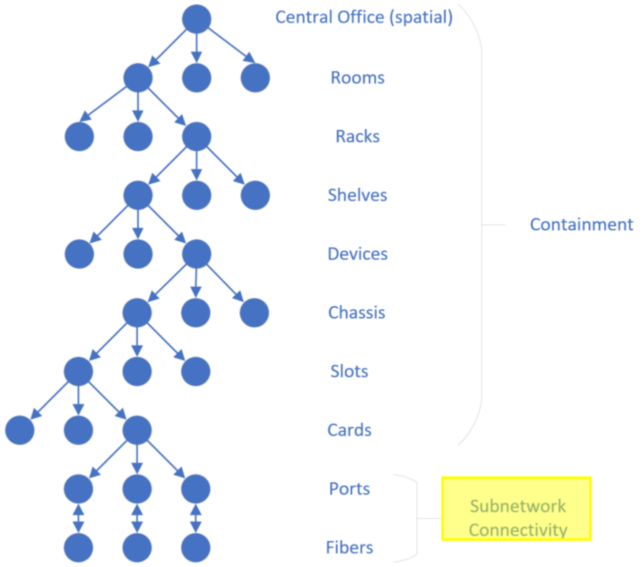
subnetwork name = Foo::Bar — Structure Line - Trench

subnetwork name = Foo::Bar — Domain Line – MV Line

Foo — subnetwork name = Foo — Edge Object - Wire

Bar — subnetwork name = Bar

subnetwork name = Unknown



Central Office (spatial)

Rooms

Racks

Shelves

Devices

Chassis

Slots

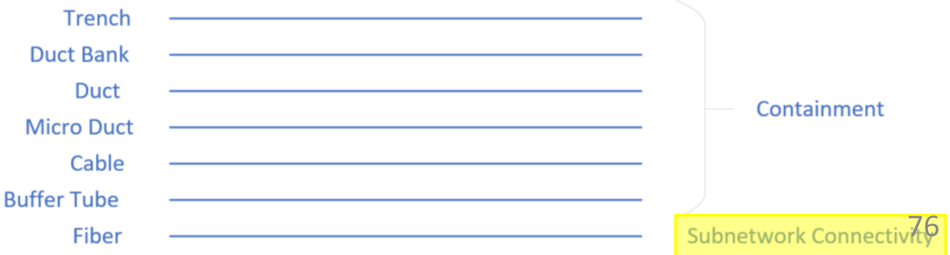
Cards

Ports

Fibers

Containment

Subnetwork Connectivity



Trench

Duct Bank

Duct

Micro Duct

Cable

Buffer Tube

Fiber

Containment

Subnetwork Connectivity

Customer requirements

motivation

- A customer wants to be able to see changes to the subnetwork name as well as the propagation burn-in field during *Update Subnetwork* in a version
  - Why? In order to verify edits made are correctly reflected in the system
    - Initial response - can be verified using Trace (subnetwork trace, trace by phase, etc.)
  - However, they still have to deal with printing paper maps for the field crews
    - Eventually could be dealt with by using Trace in the field once core apps can work with versions and UN functions

- A customer wants events to be triggered in default and versions based on changes to subnetwork name or propagated values
  - E.g., annotation updates

Update subnetwork

without-eventing vs. with-eventing, both in default and child version

Version	Default		Child	
Strategy	Without-Eventing (current)	With-Eventing	Without-Eventing (current)	With-Eventing
Subnetwork name, Propagated attribute	✓	✓	✓ (only on edited)	✓
Eventing (e.g. annotation)	X	✓	X	✓
Attribute Rule on source classes	X	✓	X	✓
Attribute Rule on Subnetline class	✓	✓	✓	✓
Undo/Redo	X	X	X	✓

New at 2.6

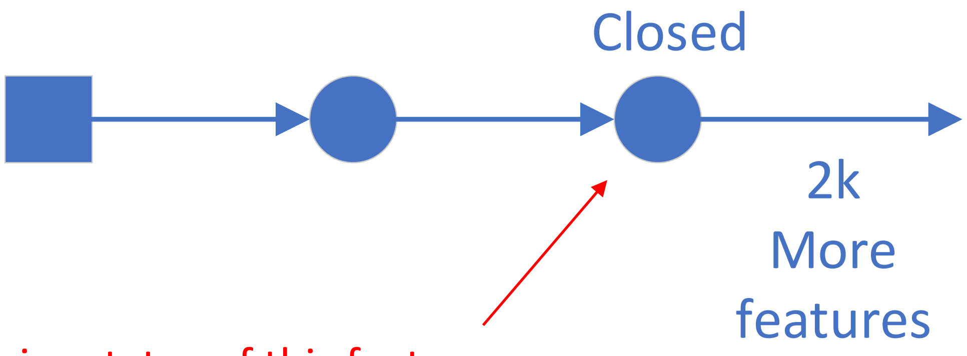
New at 2.6

* For File GDB, no options would be provided, the behavior would be same as update column in child version

Update subnetwork

concern: with-eventing mode will slow down the system

- Scenario 1



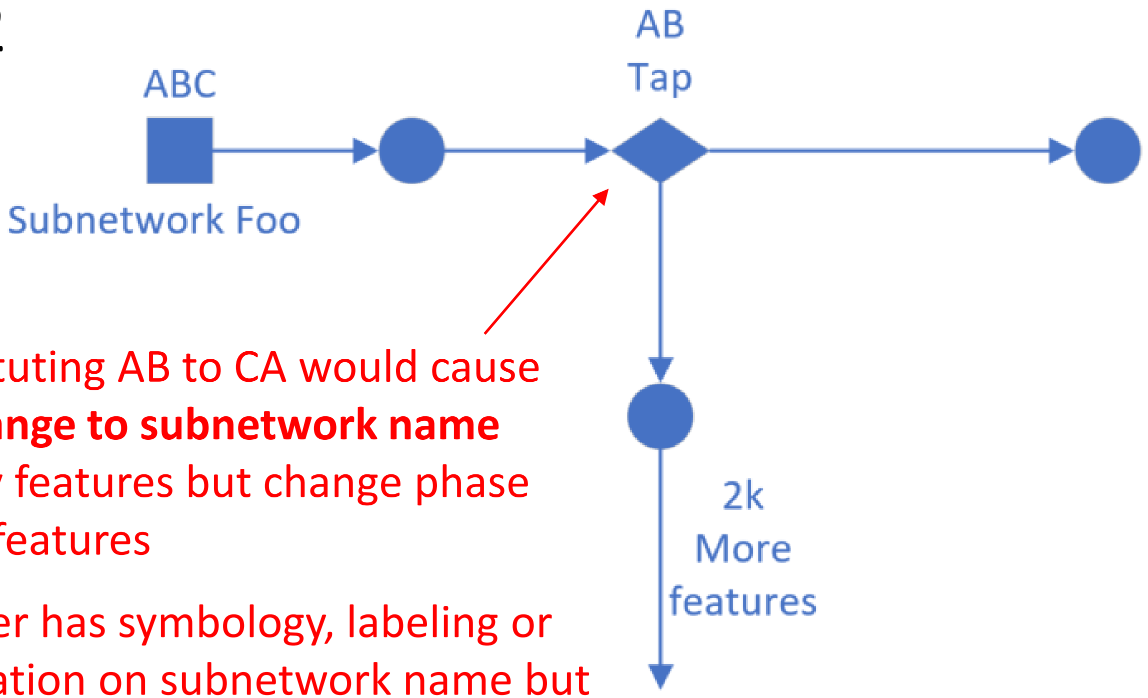
Changing device status of this feature from Closed to Open would cause updates on ~2k features downstream

By providing an option to choose which classes to update, we can reduce the number of features that get updated – i.e., if a user is interested in only line features and there are 600 line features among 2k, the number would be reduced from 2k to 600

Update subnetwork

concern: with-eventing mode will slow down the system

- Scenario 2



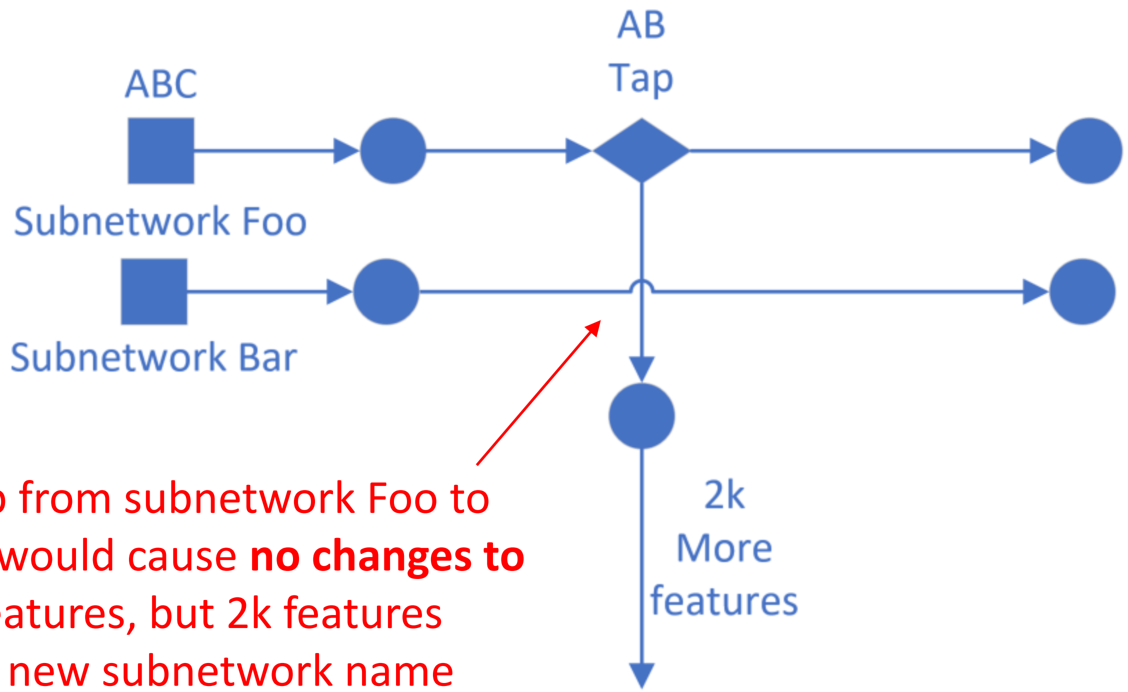
Substituting AB to CA would cause **no change to subnetwork name** on any features but change phase on 2k features

If a user has symbology, labeling or annotation on subnetwork name but not on propagated attribute, we don't need to make useless 2k updates

Update subnetwork

concern: with-eventing mode will slow down the system

- Scenario 3



Changing the tap from subnetwork Foo to subnetwork Bar would cause **no changes to phases** on any features, but 2k features would now have new subnetwork name

If a user has symbology, labeling or annotation on a propagated attribute but not on subnetwork name, we don't need to make the unnecessary 2k updates

By providing **option** to choose which attribute to update, we can reduce the number of features got updated

Esri confidential – contents subject to change

Update subnetwork

performance concerns

Version	**Default**		**Child**	
Strategy	Without-Eventing	With-Eventing	Without-Eventing	With-Eventing
Subnetwork name, Propagated attribute	✓	✓ (to keep the Default version trustable, we don't provide the same option in Default)	✓ (only on edited)	□ Device ▣ Line □ Junction □ Junction Object □ Edge Object □ Subnetwork name ▣ Propagated attribute

Subnetwork edits section options

all new options for Pro 2.6

Version	Default		Child	
Strategy	Without-Eventing	With-Eventing	Without-Eventing	With-Eventing
Subnetwork name, Propagated attribute	✓	✓	✓ (only on edited)	□ Device ▣ Line □ Junction □ Junction Object □ Edge Object □ Subnetwork name ▣ Propagated attribute
Containers in Domain Network	□ Update container features in Domain Network classes			
Structure classes	□ Update structure features			

Editing in Pro

Editing in Pro

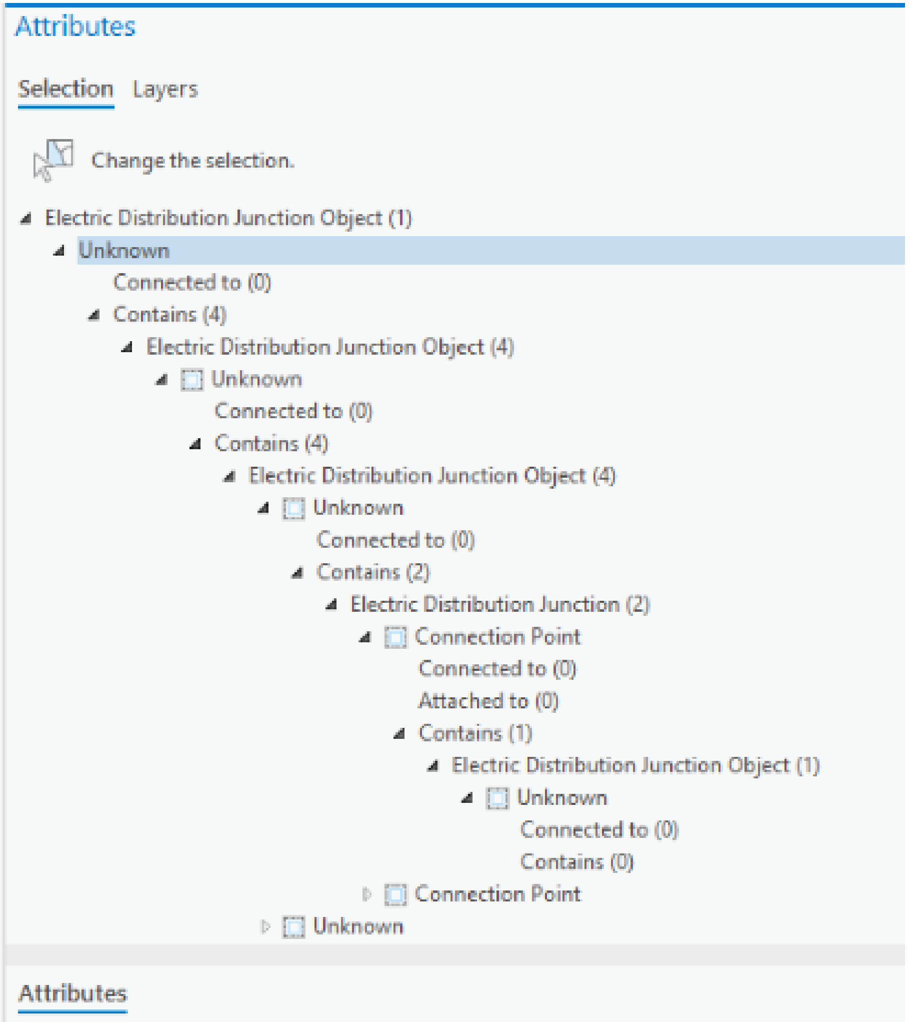
overview

- ArcGIS Pro has been the main editor of utility network features to date
- Going forward we expect Pro to continue to do most of the heavy editing, but more capabilities will become available through other clients (runtime and web)
- Within ArcGIS Pro editing is done using a combination of the Edit and Utility Network/Data ribbons
  - The capabilities from both of these ribbons are being updated with Pro 2.6 to support nonspatial objects

Editing in Pro

edit ribbon – attributes pane

- The Attributes pane is being enhanced to support nonspatial objects as both a selected object and a participant in utility network associations (connectivity, containment, and structural attachment)
- The Attributes pane will also support going deeper into containment relationships when examining a feature or object in order to show the full hierarchy of containment
- Support removing an object or feature from a containment association from the Attributes pane



Attributes

Change the selection.

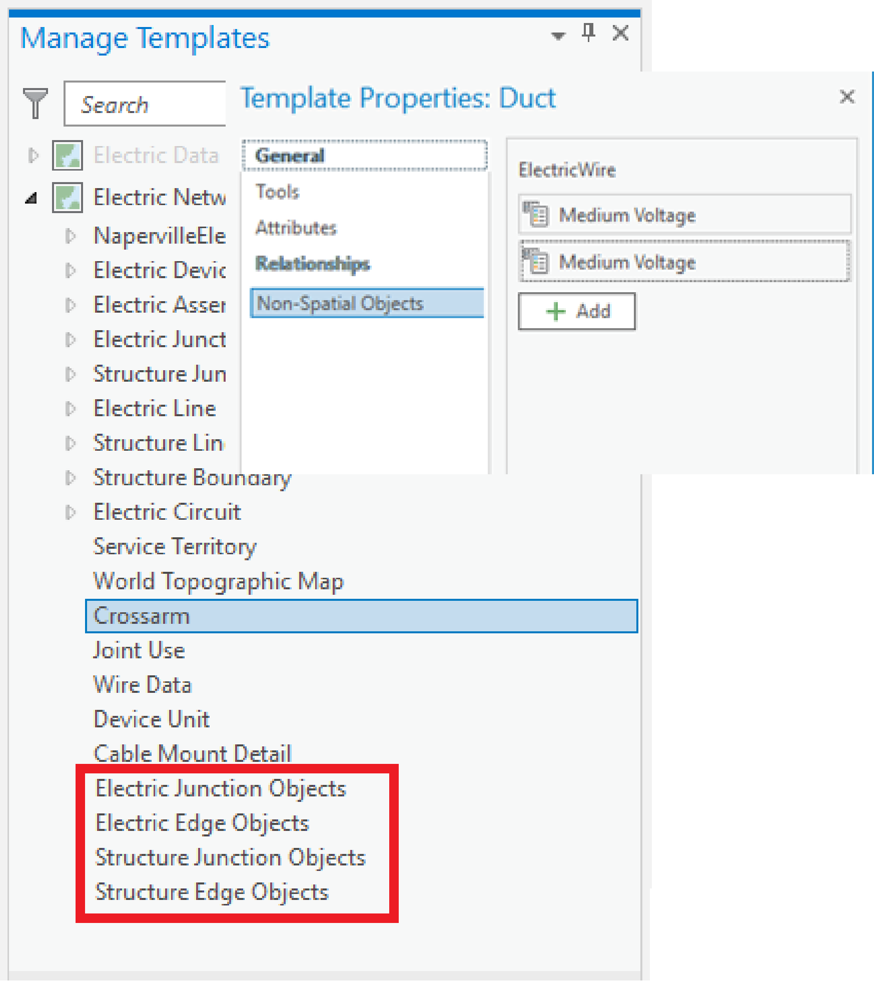
```
◢ Electric Distribution Junction Object (1)
  ◢ Unknown
      Connected to (0)
    ◢ Contains (4)
      ◢ Electric Distribution Junction Object (4)
        ◢ □ Unknown
            Connected to (0)
          ◢ Contains (4)
            ◢ Electric Distribution Junction Object (4)
              ◢ □ Unknown
                  Connected to (0)
                ◢ Contains (2)
                  ◢ Electric Distribution Junction (2)
                    ◢ □ Connection Point
                        Connected to (0)
                        Attached to (0)
                      ◢ Contains (1)
                        ◢ Electric Distribution Junction Object (1)
                          ◢ □ Unknown
                              Connected to (0)
                              Contains (0)
                    ▷ □ Connection Point
              ▷ □ Unknown
```

Esri confidential – contents subject to change

Editing in Pro

edit ribbon – feature/object templates

- Template capabilities will be enhanced to support creating templates for object classes in a manner similar to how tables are supported now with relationships
- For any given template (feature or object) it will be possible to define one or more contained features or objects
  - Through group templates it would then be possible to create the necessary depth of contained objects
  - Example:
    - Create a feature template for the port object as a junction object
    - Create a feature template for the slot object as a junction object
      - Add 10 instances of the port object on the nonspatial objects pane
    - Create a feature template for the device object as a junction object
      - Add 10 instances of the slot object on the nonspatial objects pane
    - Create a feature template for the rack object as a junction object
      - Add 10 instances of the device object on the nonspatial objects pane
    - Create a feature template for the vault feature as a structure junction
      - Add 3 instances of the rack object on the nonspatial objects pane
    - Sketch the vault feature

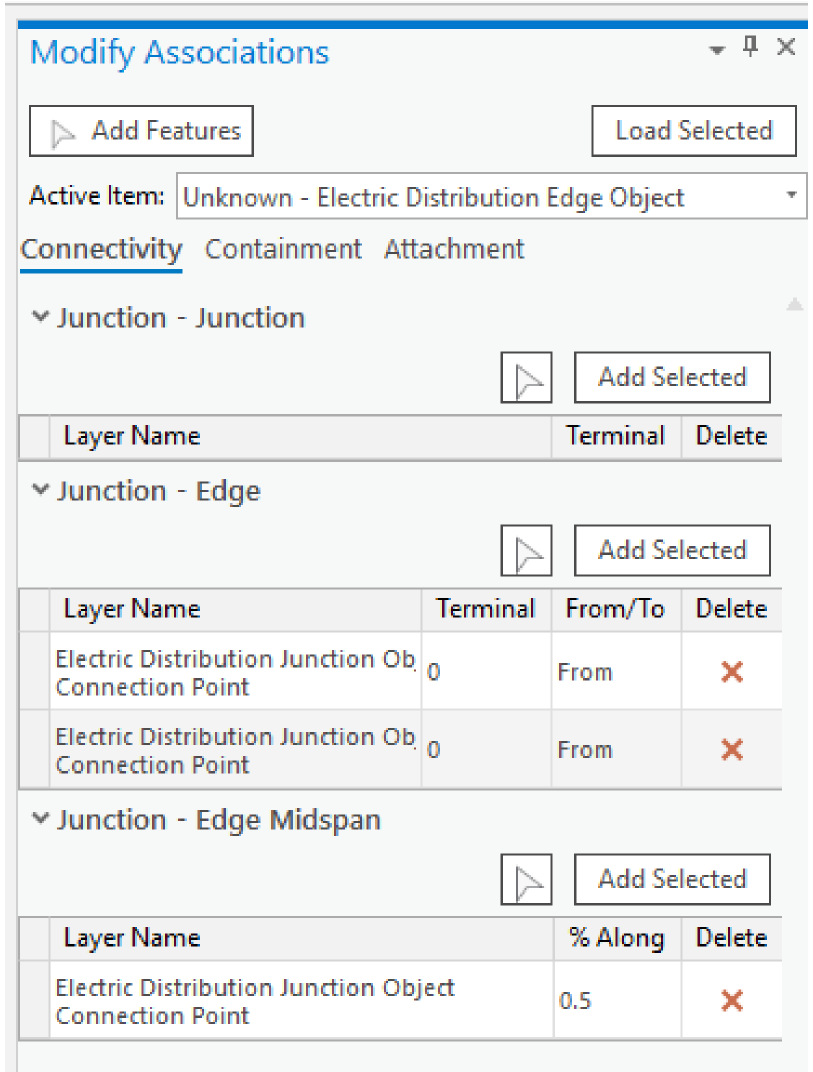


Editing in Pro

utility network/data ribbon – attributes pane

- Tools and commands are being updated appropriately to support nonspatial objects

- One of the main changes is an overhaul of the Modify Associations pane
    - Update pane to provide more information and make it easier to update associations
    - Add Selected options included to bring in nonspatial objects
    - % Along and From/To information included to support new association types



Active Item: Unknown - Electric Distribution Edge Object

Junction - Junction

Layer Name	Terminal	Delete

Junction - Edge

Layer Name	Terminal	From/To	Delete
Electric Distribution Junction Ob Connection Point	0	From	✕
Electric Distribution Junction Ob Connection Point	0	From	✕

Junction - Edge Midspan

Layer Name	% Along	Delete
Electric Distribution Junction Object Connection Point	0.5	✕

Upgrade

Upgrade

policy

- When dataset is to Utility Network Release 4, edge and junction objects are automatically supported in the utility network

- Even if a domain network doesn't utilize edge and junction objects, the object classes are created and incorporated into the information model

Upgrade

workflow

- Stop the UN services
- Upgrade the geodatabase
- Disable the utility network
- Run the Upgrade Dataset GP tool
  - Adds the pairs of object classes to each domain and structure network
- If the user wants to use the new object classes:
  - Add the attributes and perform DDL to the object classes
  - Modify the utility network metadata
  - Load data into the object classes
  - Enable the network topology
- Register the feature dataset as versioned
  - Registers the new junction and edge object classes
- Republish the utility network
- Reconcile all versions (regenerate error features)

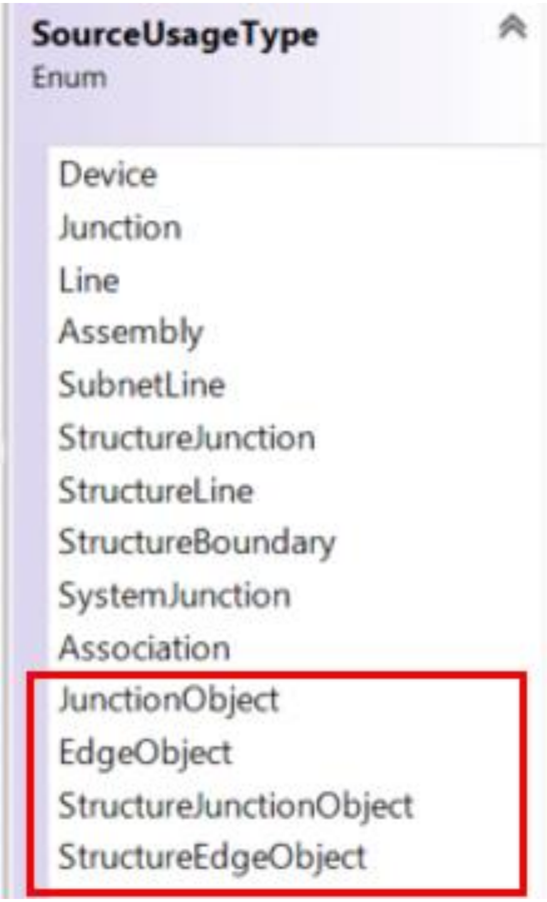
Upgrade 3 to 4

post steps after upgrade

- Manual steps are required by users after upgrade
  - Register the feature dataset as versioned (pick up the object classes)
  - Enable network topology after upgrading succeeded
  - Run reconcile against all versions to regenerate dirty areas based on a new schema

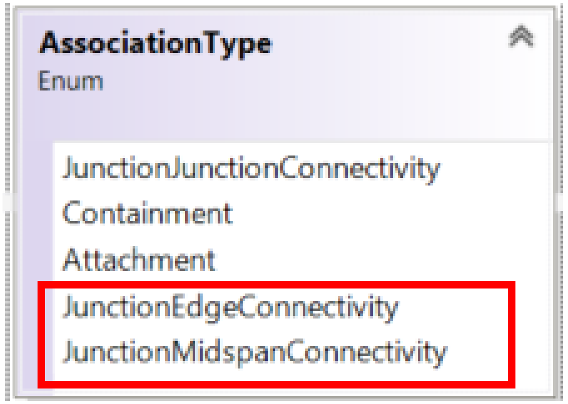
Pro SDK

Augment enumerations



SourceUsageType
Enum

- Device
- Junction
- Line
- Assembly
- SubnetLine
- StructureJunction
- StructureLine
- StructureBoundary
- SystemJunction
- Association
- JunctionObject
- EdgeObject
- StructureJunctionObject
- StructureEdgeObject

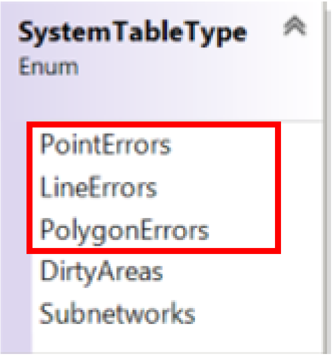


AssociationType
Enum

- JunctionJunctionConnectivity
- Containment
- Attachment
- JunctionEdgeConnectivity
- JunctionMidspanConnectivity

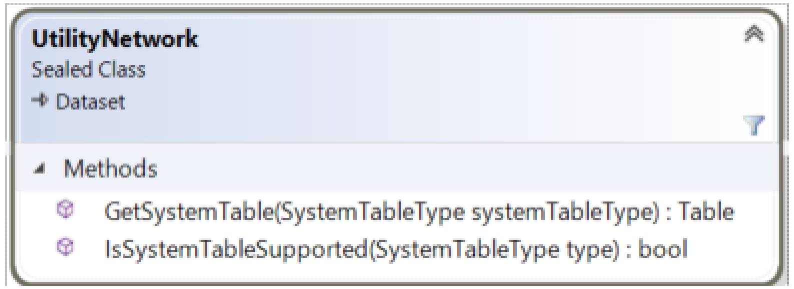
UtilityNetwork class and SystemTableType enum

- Starting UtilityNetworkDefinition.GetSchemaVersion() >= 4, PointErrors, LineErrors, and PolygonErrors tables no longer exist



SystemTableType
Enum

PointErrors
LineErrors
PolygonErrors
DirtyAreas
Subnetworks



UtilityNetwork
Sealed Class
→ Dataset

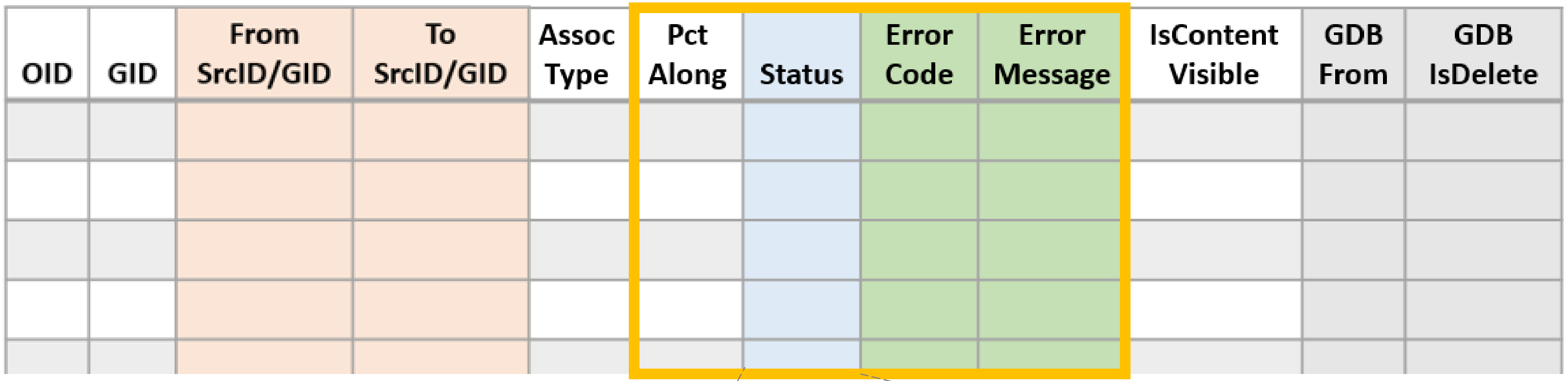
Methods

GetSystemTable(SystemTableType systemTableType) : Table
IsSystemTableSupported(SystemTableType type) : bool

- The new query method IsSystemTableSupported(SystemTableType) returns false for the three affected enums
- If IsSystemTableSupported() returns false, calling GetSystemTable() throws exception
- Add SystemTableType.Associations

Association table

Associations



OID	GID	From SrcID/GID	To SrcID/GID	Assoc Type	Pct Along	Status	Error Code	Error Message	IsContent Visible	GDB From	GDB IsDelete

- New fields
  - Status
  - Error code
  - Error message
  - Percent along

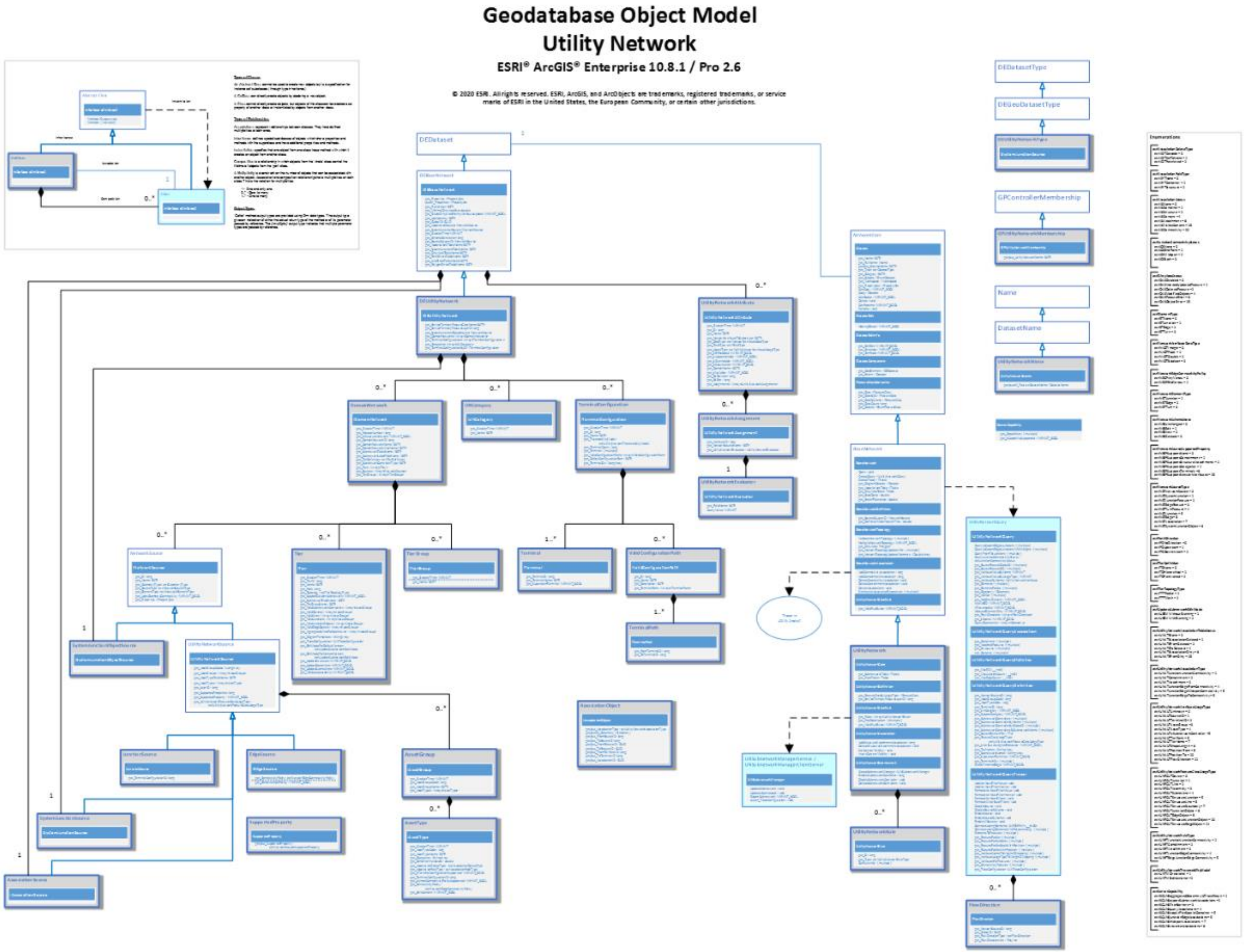
Status bits:
1 – Deleted association
2 – Deleted FROM feature/object
3 – Deleted TO feature/object
4 – Modified association
5 – Modified FROM feature/object
6 – Modified TO feature/object

Enterprise SDK

Additions

un abstractions

- UN interfaces are exposed in Enterprise SDK at 2.6/10.8.1
- This includes support for nonspatial objects in Server Object Extensions (SOEs) and Server Object Interceptors(SOIs)


Geodatabase Object Model
Utility Network
ESRI® ArcGIS® Enterprise 10.8.1 / Pro 2.6
© 2020 ESRI. All rights reserved. ESRI, ArcGIS, and ArcObjects are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions.
DEDataset
DEDatasetType
DEGeoDatasetType
GPControllerMembership
Name
DatasetName
0..*
1
1..*

Esri confidential – contents subject to change

Arcade & Attribute Rules

Esri confidential – contents subject to change

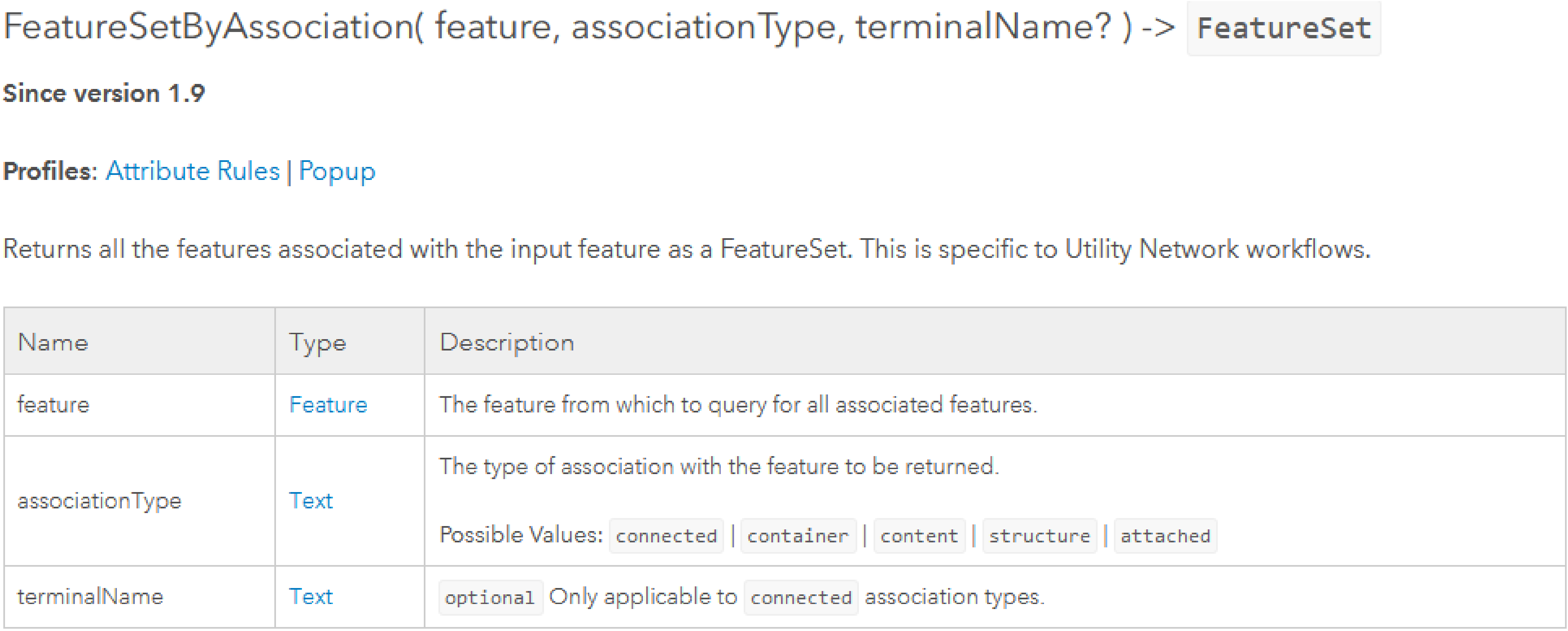
Arcade Changes (2.6 Min-ship)

1. FeatureSetByAssociation (Arcade Team)
   - Ignore logically deleted Associations
   - Return the new association types (junction edge, midspan & percent along)

2. Attribute Rules DML (GDB Team)
   - New DML to create junction/edge & midspan associations

FeatureSetByAssociation

input



FeatureSetByAssociation(feature, associationType, terminalName?) -> `FeatureSet`

Since version 1.9

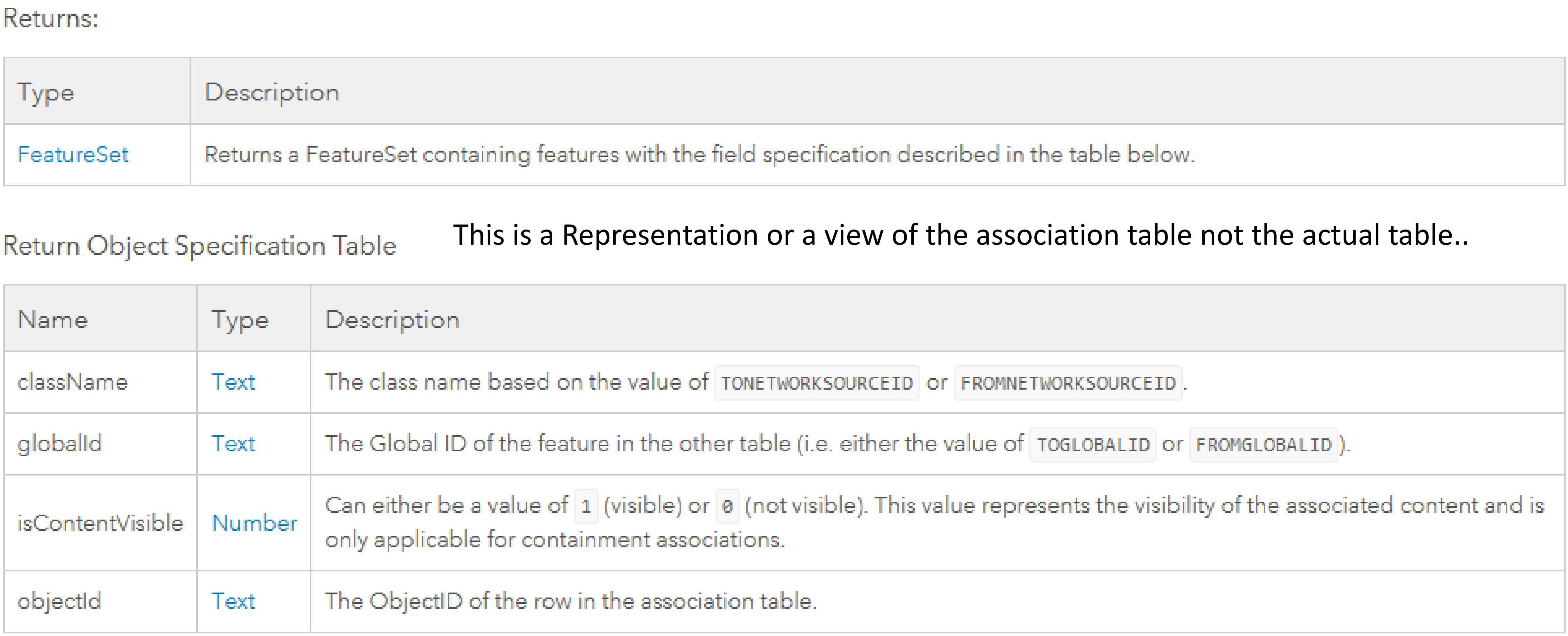
Profiles: Attribute Rules | Popup

Returns all the features associated with the input feature as a FeatureSet. This is specific to Utility Network workflows.

Name	Type	Description
feature	Feature	The feature from which to query for all associated features.
associationType	Text	The type of association with the feature to be returned. Possible Values: `connected` | `container` | `content` | `structure` | `attached`
terminalName	Text	`optional` Only applicable to `connected` association types.

FeatureSetByAssociation

return



Returns:

Type	Description
FeatureSet	Returns a FeatureSet containing features with the field specification described in the table below.

Return Object Specification Table

This is a Representation or a view of the association table not the actual table..

Name	Type	Description
className	Text	The class name based on the value of `TONETWORKSOURCEID` or `FROMNETWORKSOURCEID`.
globalId	Text	The Global ID of the feature in the other table (i.e. either the value of `TOGLOBALID` or `FROMGLOBALID`).
isContentVisible	Number	Can either be a value of `1` (visible) or `0` (not visible). This value represents the visibility of the associated content and is only applicable for containment associations.
objectId	Text	The ObjectID of the row in the association table.

Arcade

FeatureSetByAssociation

- Use case – Ability to not return associations that are deleted
- FeatureSetByAssociation should add an additional filter.
  - Where either of the STATUS BITs are set
  - esriUNATSAssociationDeleted is set
  - esriUNATSFromDeleted is set
  - esriUNATSToDeleted is set
- STATUS in ([1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 52, 53, 54, 55, 57, 58, 59, 60, 61, 62,63)
- New SDE API for bitwise operator (where STATUS & (1,2,4) != 0)
- Arcade should check the version of the UN by querying the data element.
- If UN version is 4 or later add the filter, else use the existing behavior for backward compatibility.

esriUtilityNetworkAssociationTableStatus

esriUNATSNone = 0 Nothing is deleted or dirty
esriUNATSAssociationDeleted = 1 Association itself is deleted
esriUNATSFromDeleted = 2 From side is deleted
esriUNATSToDeleted = 4 To side is deleted
esriUNATSAssociationDirty = 8 Association itself is dirty
esriUNATSFromDirty = 16 From side is dirty
esriUNATSToDirty = 32 To side is dirty

DELETE_STATUS_FILTER

Esri confidential – contents subject to change

Arcade

FeatureSetByAssociation

- Use case – Ability to return:
  - Junction Edge From / To Association
  - Midspan Associations
- FeatureSetByAssociation($feature, “JunctionEdge”)
- FeatureSetByAssociation($feature, “midspan”)

esriUtilityNetworkAssociationType

esriUNATJunctionJunctionConnectivity = 1

esriUNATContainment = 2

esriUNATAttachment = 3

esriUNATJunctionEdgeFromConnectivity = 4 Junction is connected to edge at the ‘from’ side of edge

esriUNATJunctionMidspanConnectivity = 5 Junction is connected to the midspan of the edge

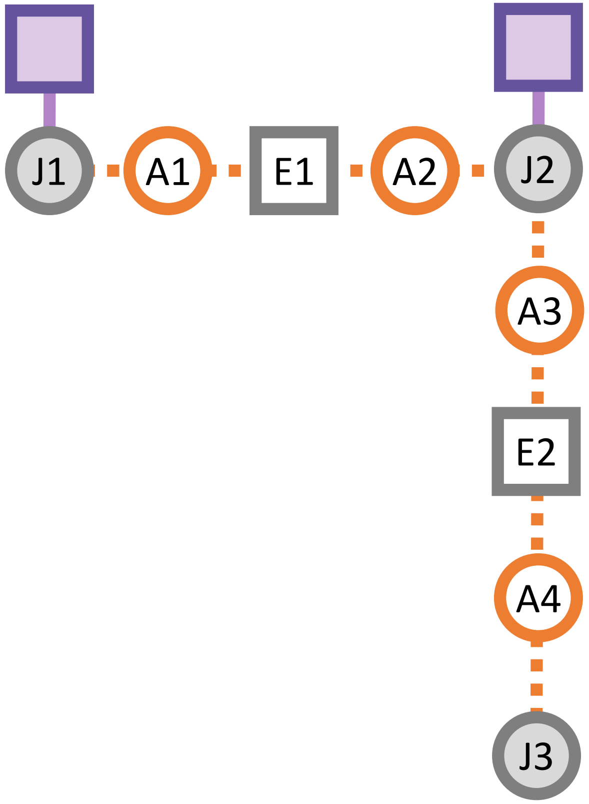
esriUNATJunctionEdgeToConnectivity = 6 Junction is connected to edge at the ‘to’ side of edge

Esri confidential – contents subject to change

FeatureSetByAssociation

JunctionEdge – input is a junction

- FeatureSetByAssociation(“J1”, “JunctionEdge”)
  - WHERE STATUS NOT IN (DELETE_STATUS_FILTER)
  - AND ASSOCIATIONTYPE IN (4, 6)
  - AND (FROMGLOBALID = “J1”)
  - Returns **A1** association
  - Result (1)
    - [{ClassName: EdgeObject, GlobalId: E1 , “Side”: “From”} }]
- FeatureSetByAssociation(“J2”, “JunctionEdge”)
  - WHERE SSTATUS NOT IN (DELETE_STATUS_FILTER)
  - AND ASSOCIATIONTYPE IN (4, 6)
  - AND (FROMGLOBALID = “J2”)
  - Returns **A2,A3** associations
  - Result (2)
    - [{ClassName: EdgeObject, GlobalId: E1 , “Side”: “To”} } ,
    - { ClassName: EdgeObject, GlobalId: E2 , “Side”: “From”} }]

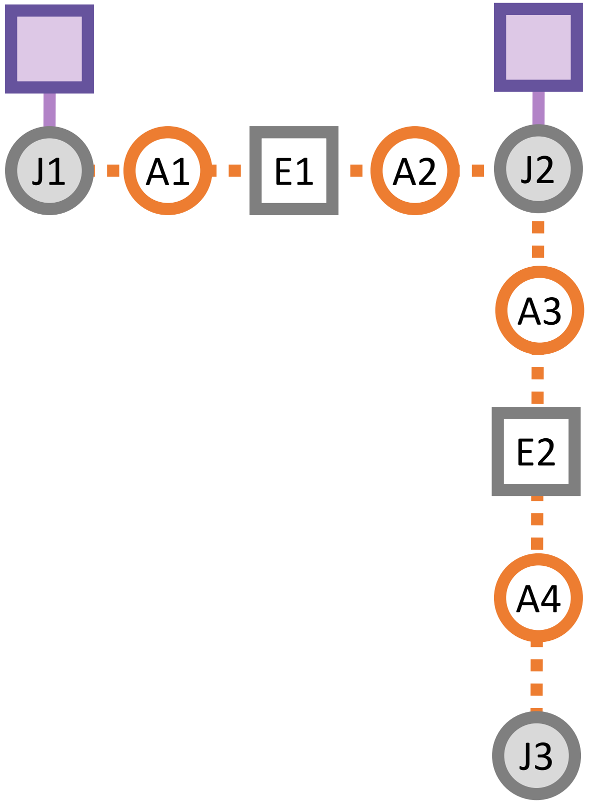


Esri confidential – contents subject to change

FeatureSetByAssociation

JunctionEdge – input is an edge

- FeatureSetByAssociation(“E1”, “JunctionEdge”)
  - WHERE STATUS NOT IN (DELETE_STATUS_FILTER)
  - AND ASSOCIATIONTYPE IN (4, 6)
  - AND (TOGLOBALID = “E1”)
  - Returns **A1, A2** association
  - Result (2)
    - [{ClassName: JunctionObject, GlobalId: J1, “Side”: “From”} ,
    - { ClassName: JunctionObject, GlobalId: J2, “Side”: “To” }]
- FeatureSetByAssociation(“E2”, “JunctionEdge”)
  - WHERE STATUS NOT IN (DELETE_STATUS_FILTER)
  - AND ASSOCIATIONTYPE IN (4, 6)
  - AND (TOGLOBALID = “E2”)
  - Returns **A3,A4** associations
  - Result (2)
    - [{ClassName: JunctionObject, GlobalId: J2, “Side”: “From”} ,
    - { ClassName: JunctionObject, GlobalId: J3, “Side”: “To” }]

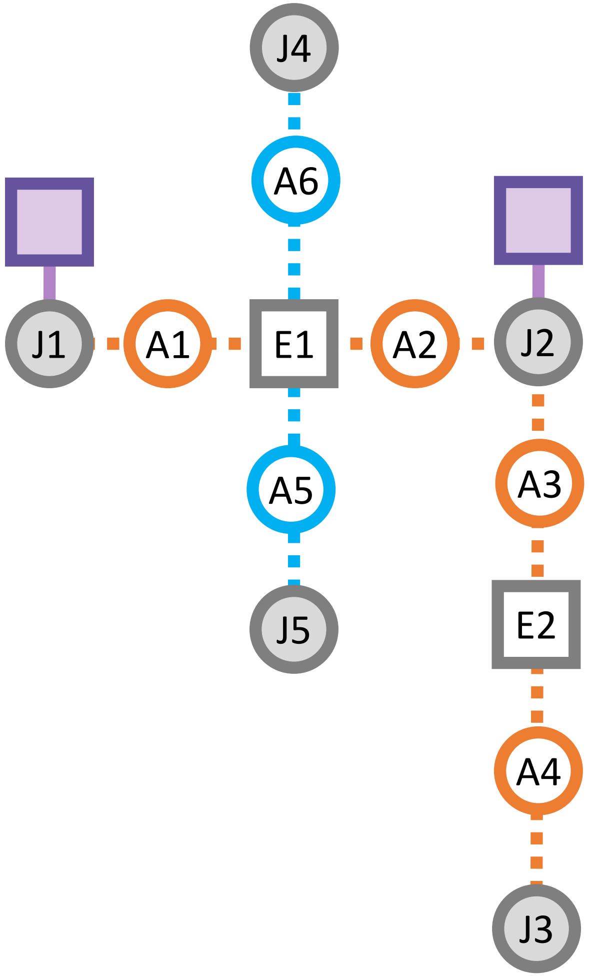


Esri confidential – contents subject to change

FeatureSetByAssociation

midspan – input is a junction

- FeatureSetByAssociation(“J5”, “Midspan”)
  - WHERE STATUS NOT IN (DELETE_STATUS_FILTER)
  - AND ASSOCIATIONTYPE = 5
  - AND (FROMGLOBALID = “J5”)
  - Returns **A5** association
  - Result (1)
    - [{ClassName: EdgeObject, GlobalId: E1 , “PercentAlong”: 0.7} }]

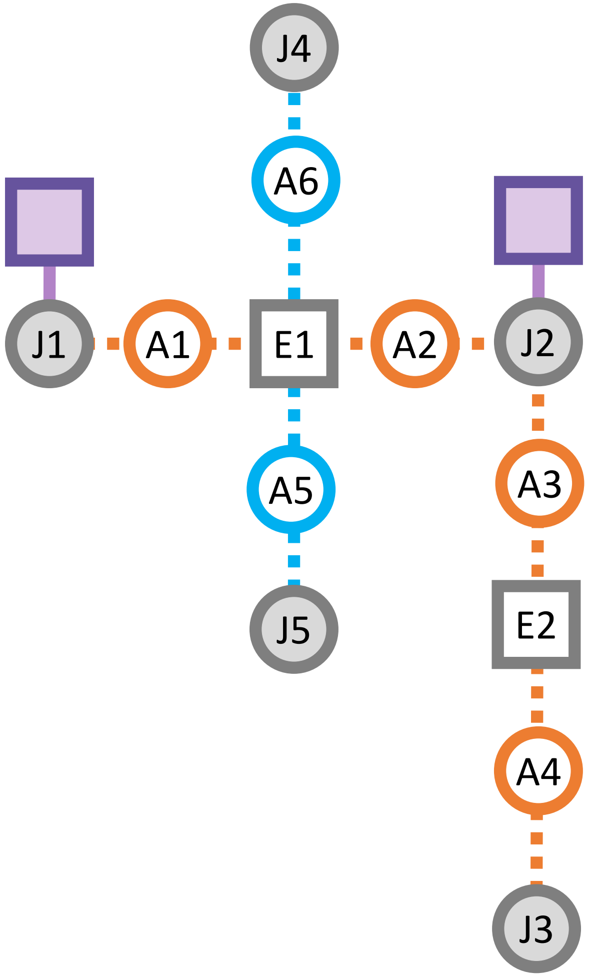


Esri confidential – contents subject to change

FeatureSetByAssociation

midspan – input is an edge

- FeatureSetByAssociation(“E1”, “Midspan”)
  - WHERE STATUS NOT IN (DELETE_STATUS_FILTER)
  - AND ASSOCIATIONTYPE = 5
  - AND (TOGLOBALID = “E1”)
  - Returns **A5,A6** association
  - Result (2)
    - [{ClassName: JunctionObject, GlobalId: J5 , “PercentAlong”: 0.7} } ,
    - {ClassName: JunctionObject, GlobalId: J4 , “PercentAlong”: 0.4} }]



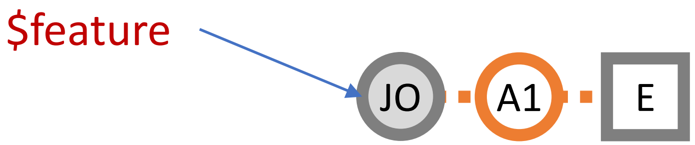
Attribute Rules DML

creating JunctionEdge association - $feature is a junction

- Create Junction-Edge-From connectivity between $feature (JO) and an edgeObject (E) where $feature at the from side

Signature	Example

{
  "className": "EdgeObject",
  "updates": [
    {
      "globalId": "<globalId of the edge object>",
      "associationType": "junction-edge-from"
    }
  ]
}
	{
	  "className": "EdgeObject",
	  "updates": [
	    {
	      "globalId": edgeObject.globalId,
	      "associationType": "junction-edge-from"
	    }
	  ]
	}



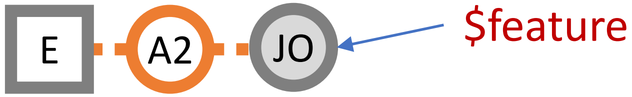
Attribute Rules DML

creating JunctionEdge association - $feature is a junction

- Create Junction-Edge-From connectivity between $feature (JO) and an edgeObject (E) where $feature at the TO side

Signature	Example

{
"className": "EdgeObject",
"updates": [
{
"globalId": "<globalId of the edge object>",
"associationType": "junction-edge-to"
}
]
}
	{
	"className": "EdgeObject",
	"updates": [
	{
	"globalId": edgeObject.globalId,
	"associationType": "junction-edge-to"
	}
	]
	}



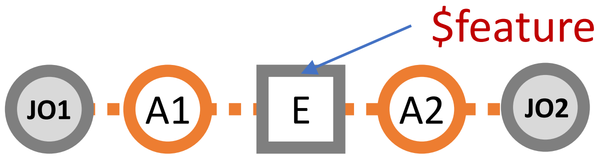
Attribute Rules DML

creating JunctionEdge association - $feature is an edge

- Create Junction-Edge connectivity between $feature (EdgeObject) and two Junction objects JO1 & JO2 at from/to side respectively

Signature	Example

{
"className": “JunctionObject",
"updates": [
{
"globalId": "<globalId of the edge object>",
"associationType": "junction-edge-from"
},
{
"globalId": "<globalId of the edge object>",
"associationType": "junction-edge-to"
}
]
}
	{
	"className": “JunctionObject",
	"updates": [
	{
	"globalId": JO1.globalId,
	"associationType": "junction-edge-from"
	},
	{
	"globalId": JO2.globalId
	"associationType": "junction-edge-to"
	}
	]
	}



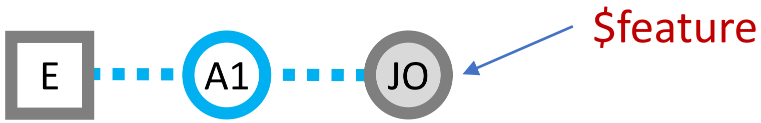
Attribute Rules DML

creating midspan association - $feature is a junction

- Create midspan connectivity between $feature (JO) and an edgeObject (E) at x percent along

Signature	Example

{
"className": "EdgeObject",
"updates": [
{
"globalId": "<globalId of the edge object>",
“percentAlong": x,
"associationType": “midspan"
}
]
}
	{
	"className": "EdgeObject",
	"updates": [
	{
	"globalId": edgeObject.globalId,
	“percentAlong": 0.5,
	"associationType": “midspan"
	}
	]
	}



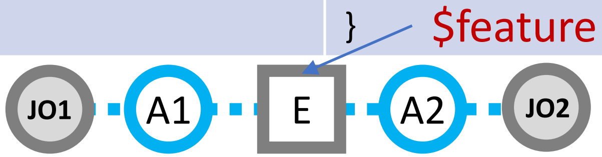
Attribute Rules DML

creating Midspan association - $feature is an edge

- **Create Midspan connectivity between $feature (Edge Object) and Junction Objects (JO1, JO2) x1, x2 percent along respectively**

Signature	Example

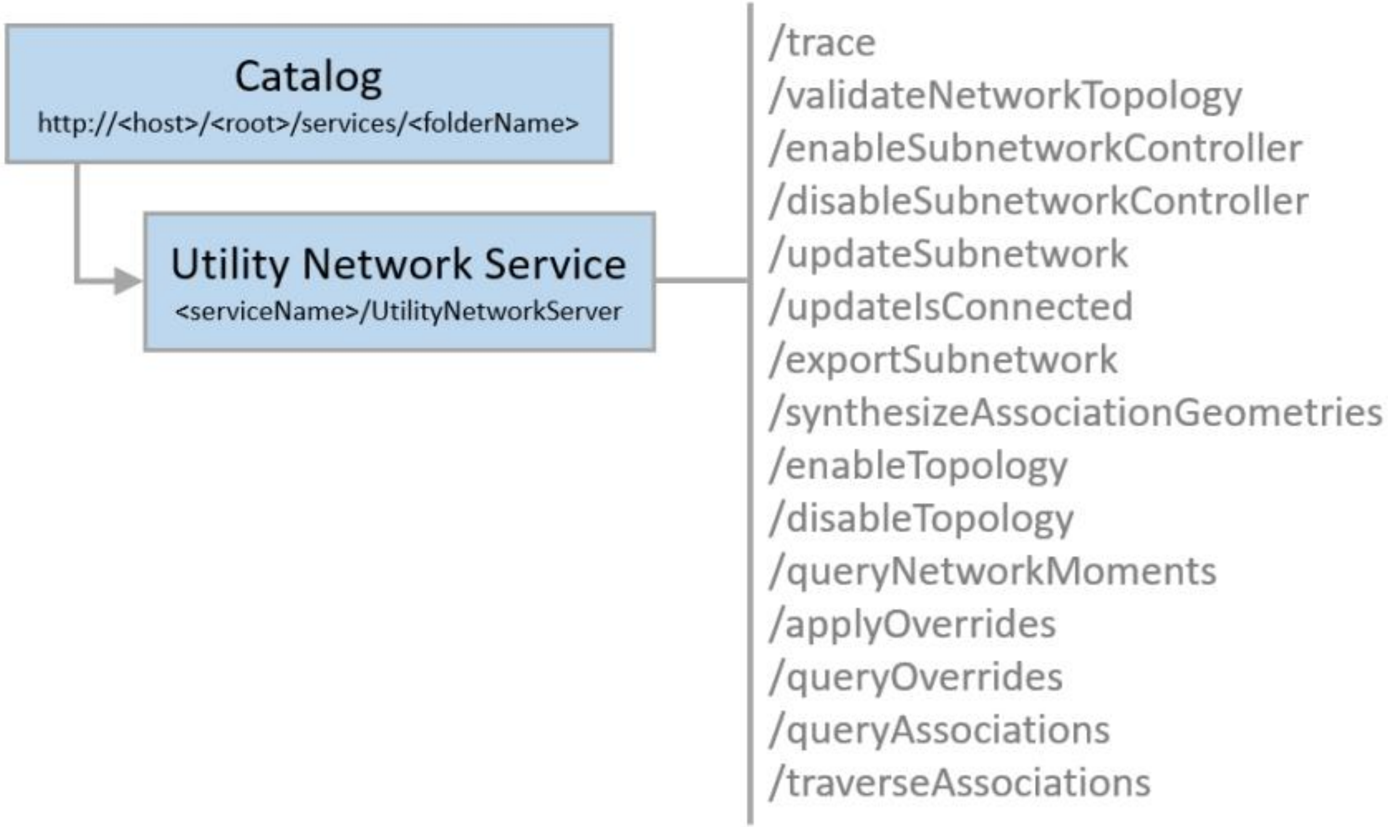
{
"className": “JunctionObject",
"updates": [
{
"globalId": "<globalId of the edge object>",
“percentAlong": x1,
"associationType": “midspan"
},
{
"globalId": "<globalId of the edge object>",
“percentAlong": x2,
"associationType": “midspan"
}
]
}
	{
	"className": “JunctionObject",
	"updates": [
	{
	"globalId": JO1.globalId,
	“percentAlong": 0.7,
	"associationType": “midspan"
	},
	{
	"globalId": JO2.globalId,
	“percentAlong": 0.5,
	"associationType": “midspan"
	}
	]
	}



REST API

Overview

- The Utility Network Server REST API will be augmented to expose a new operation that will allow clients to effectively and performantly traverse the Association table and hierarchy
    - The new operations are queryAssociations and traverseAssociations



Catalog
http://<host>/<root>/services/<folderName>

Utility Network Service
<serviceName>/UtilityNetworkServer

/trace
/validateNetworkTopology
/enableSubnetworkController
/disableSubnetworkController
/updateSubnetwork
/updateIsConnected
/exportSubnetwork
/synthesizeAssociationGeometries
/enableTopology
/disableTopology
/queryNetworkMoments
/applyOverrides
/queryOverrides
/queryAssociations
/traverseAssociations

traverseAssociations

traversal types

- Traversal of associations allows clients to extract and obtain useful information from the associations table
- The supported traversal types are:
  - DirtyAreaExpansion – downward, then upward, with the exit filter on the first spatial feature in each direction
  - FirstContainers – upward on containment associations, with the exit filter on first spatial feature
  - SpatialParents – upward on all association types, with the exit filter on the first spatial feature
  - TopContainers - upward
  - ErrorsNotModified – downward, with the exit filter on the first spatial feature
  - ModifiedObjects – downward, with the exit filter on the first spatial feature

traverseAssociations

- When the exit filter is on the first spatial feature, we will not traverse associations from spatial to nonspatial and then back to spatial.
- It is also possible to configure an association traversal that does not correspond to one of the supported types by specifying other generic parameters:
  - DirtyFilter
  - ErrorFitler
  - StopAtFirstSpatial
  - MaxDepth

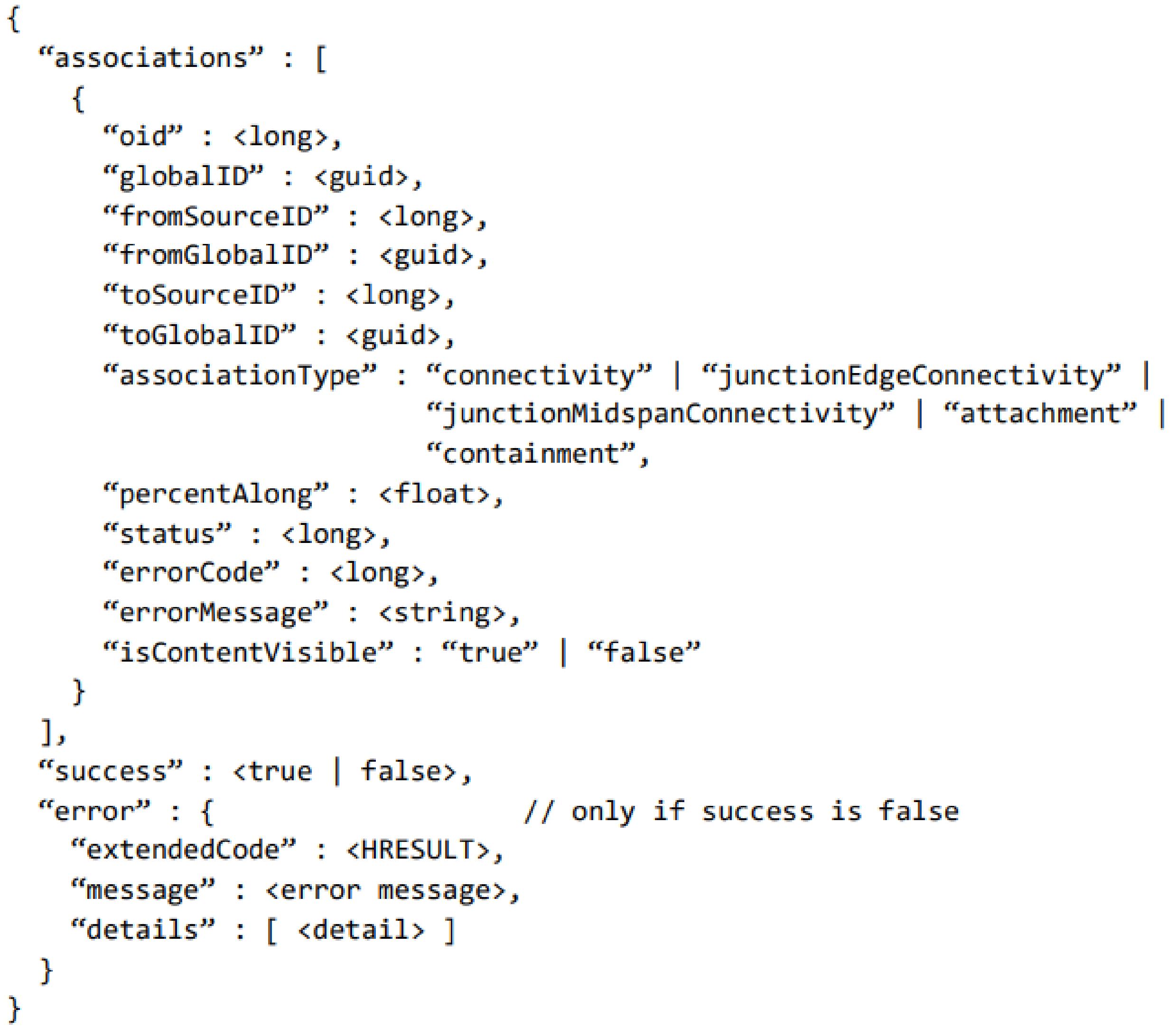
traverseAssociations

parameters

- gdbVersion – geodatabase version (default is DEFAULT)
- moment – the session moment (default is current moment)
- traversalType – the six supported traversal types as described previously
- Direction – the direction (ascending, descending) of the traversal (default is descending)
- dirtyFilter – whether or not to filter based upon association dirty status (default is none)
- errorFilter – whether or not to filter based upon the association error status (default is none)
- stopAtFirstSpatial – whether or not to stop traversal when transitioning from object to feature
- maxDepth – a limit on how many hops to take through the association graph
- elements – feature or object elements from which to initiate the traversal

traverseAssociations

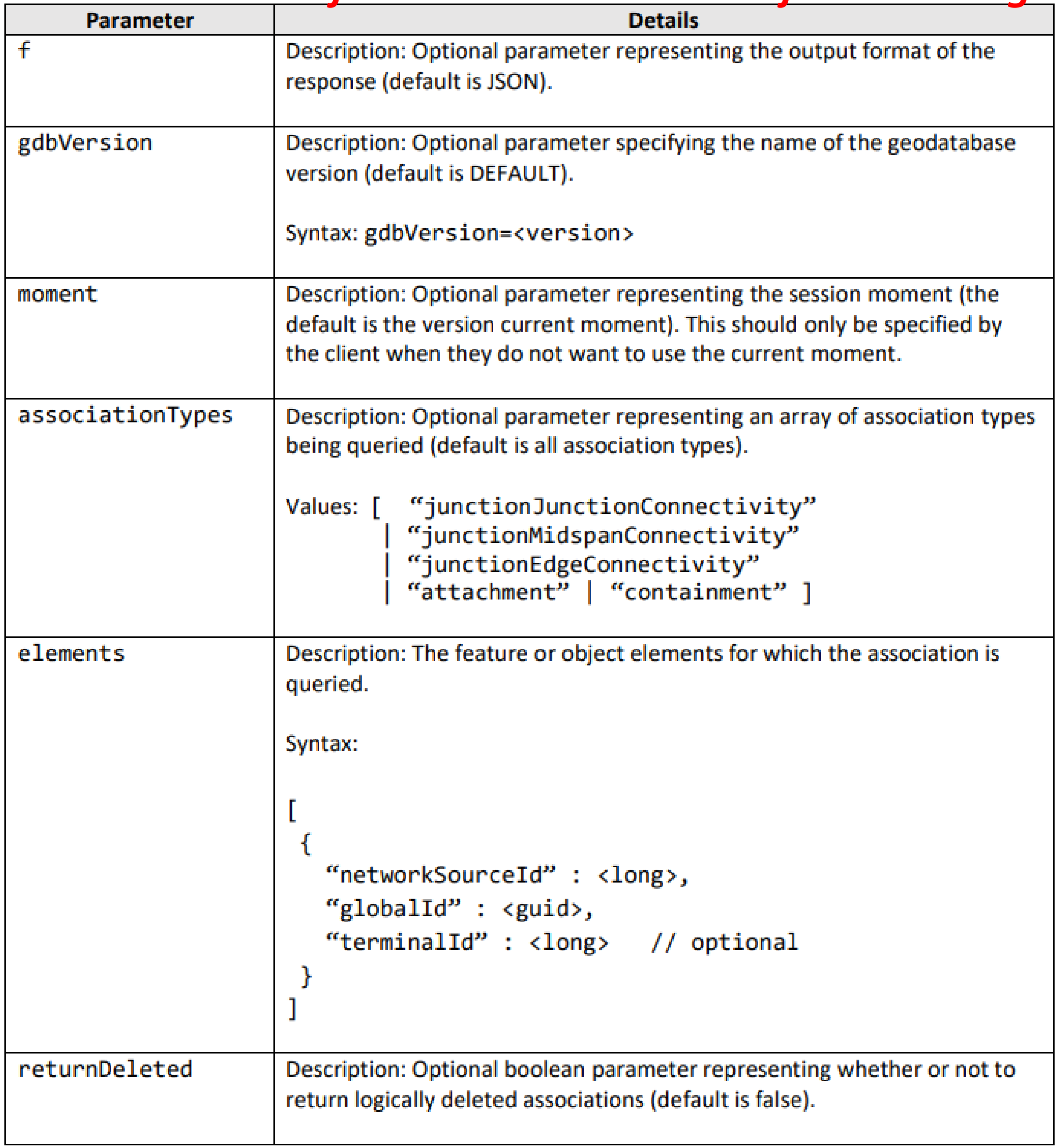
JSON response



```
{
  “associations” : [
    {
      “oid” : <long>,
      “globalID” : <guid>,
      “fromSourceID” : <long>,
      “fromGlobalID” : <guid>,
      “toSourceID” : <long>,
      “toGlobalID” : <guid>,
      “associationType” : “connectivity” | “junctionEdgeConnectivity” |
                          “junctionMidspanConnectivity” | “attachment” |
                          “containment”,
      “percentAlong” : <float>,
      “status” : <long>,
      “errorCode” : <long>,
      “errorMessage” : <string>,
      “isContentVisible” : “true” | “false”
    }
  ],
  “success” : <true | false>,
  “error” : {                        // only if success is false
    “extendedCode” : <HRESULT>,
    “message” : <error message>,
    “details” : [ <detail> ]
  }
}
```

queryAssociations

- Clients can query the association table through this simplified operation; the enhancements made as part of supporting nonspatial objects introduces additional complexity (e.g., logical versus physical deletion)
  - This is in addition to the existing feature server-based direct querying of the association table
  - This operation is backward compatible with older schema generations (e.g., 3 and lower)



Parameter	Details
`f`	Description: Optional parameter representing the output format of the response (default is JSON).
`gdbVersion`	Description: Optional parameter specifying the name of the geodatabase version (default is DEFAULT). Syntax: `gdbVersion=<version>`
`moment`	Description: Optional parameter representing the session moment (the default is the version current moment). This should only be specified by the client when they do not want to use the current moment.

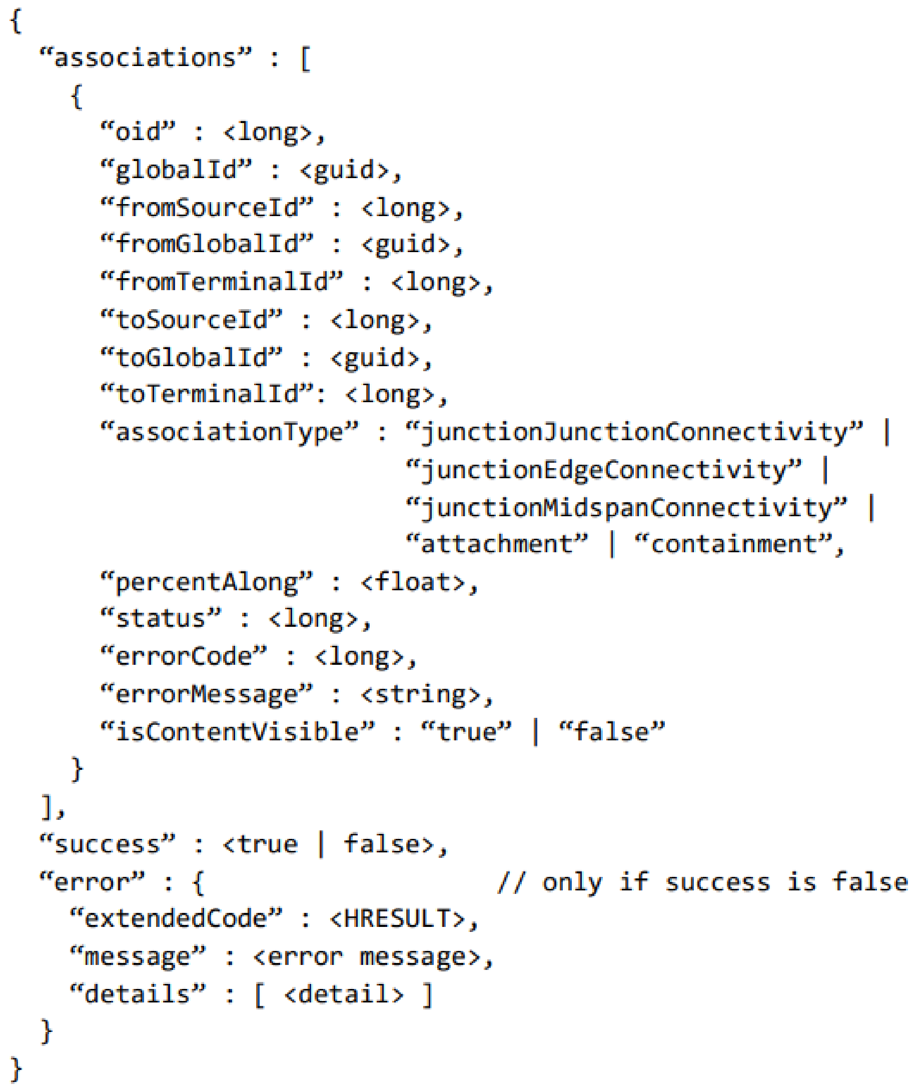
`associationTypes`	Description: Optional parameter representing an array of association types being queried (default is all association types).
	Values: `[ “junctionJunctionConnectivity”`
	`| “junctionMidspanConnectivity”`
	`| “junctionEdgeConnectivity”`
	`| “attachment” | “containment” ]`

`elements`	Description: The feature or object elements for which the association is queried.
	Syntax:
	`[`
	`{`
	`“networkSourceId” : <long>,`
	`“globalId” : <guid>,`
	`“terminalId” : <long>    // optional`
	`}`
	`]`

`returnDeleted`	Description: Optional boolean parameter representing whether or not to return logically deleted associations (default is false).

queryAssociations

JSON response

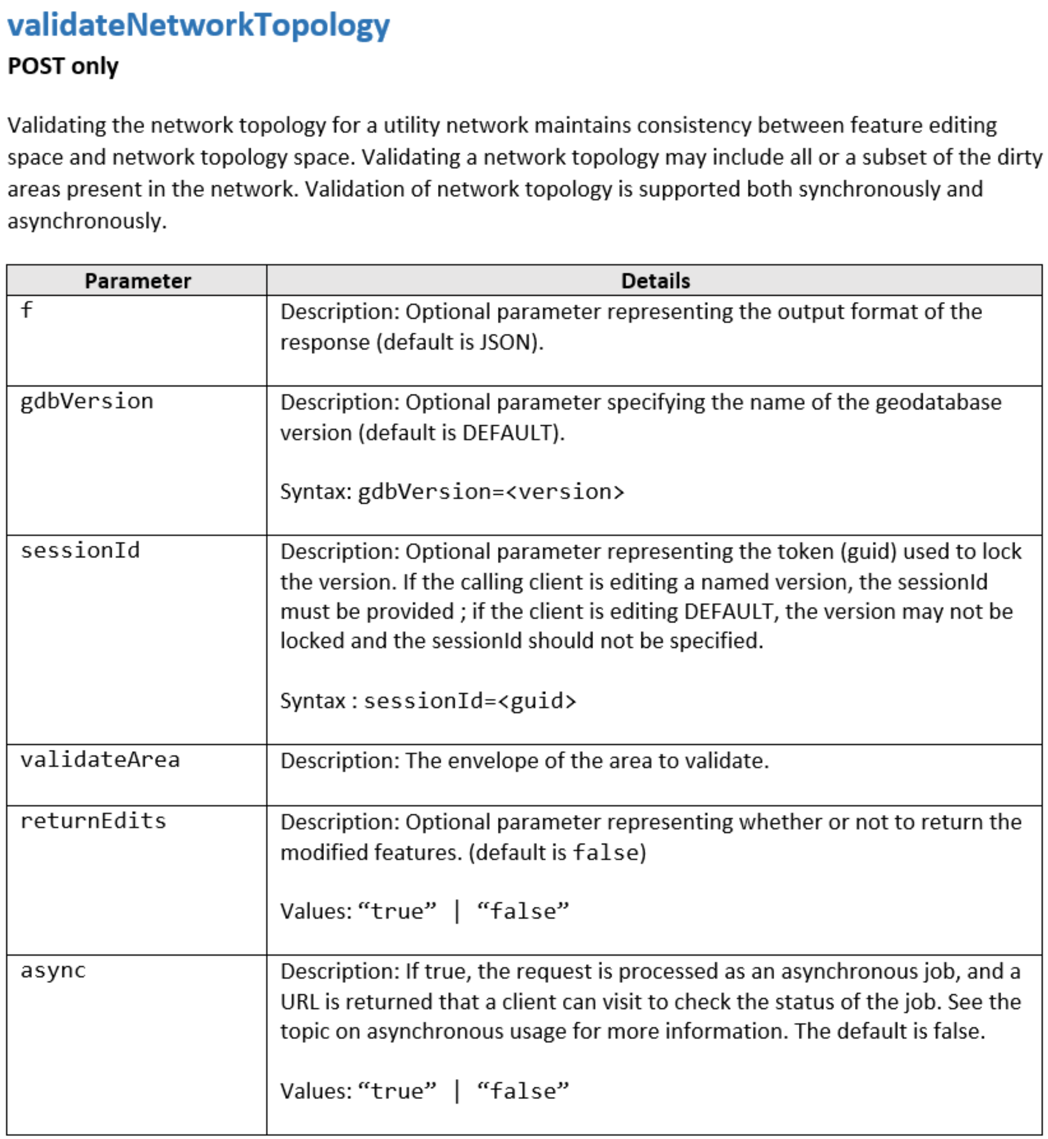


```
{
  “associations” : [
    {
      “oid” : <long>,
      “globalId” : <guid>,
      “fromSourceId” : <long>,
      “fromGlobalId” : <guid>,
      “fromTerminalId” : <long>,
      “toSourceId” : <long>,
      “toGlobalId” : <guid>,
      “toTerminalId”: <long>,
      “associationType” : “junctionJunctionConnectivity” |
                          “junctionEdgeConnectivity” |
                          “junctionMidspanConnectivity” |
                          “attachment” | “containment”,
      “percentAlong” : <float>,
      “status” : <long>,
      “errorCode” : <long>,
      “errorMessage” : <string>,
      “isContentVisible” : “true” | “false”
    }
  ],
  “success” : <true | false>,
  “error” : {                  // only if success is false
    “extendedCode” : <HRESULT>,
    “message” : <error message>,
    “details” : [ <detail> ]
  }
}
```

validateNetworkTopology

additional parameters

- Additional parameters will be added to support workflows where the user want to either:
  - Force a validation of an area, independent of dirty areas and feature/object status; e.g., forceValidation – an optional Boolean
  - Validate a collection of edge and junction objects that are not associated with a containment hierarchy – this can be used to clean the index be removing orphaned network elements; e.g., a set of sourceID,globalID pairs, or a query filter to be used against the association table (TBD)



validateNetworkTopology

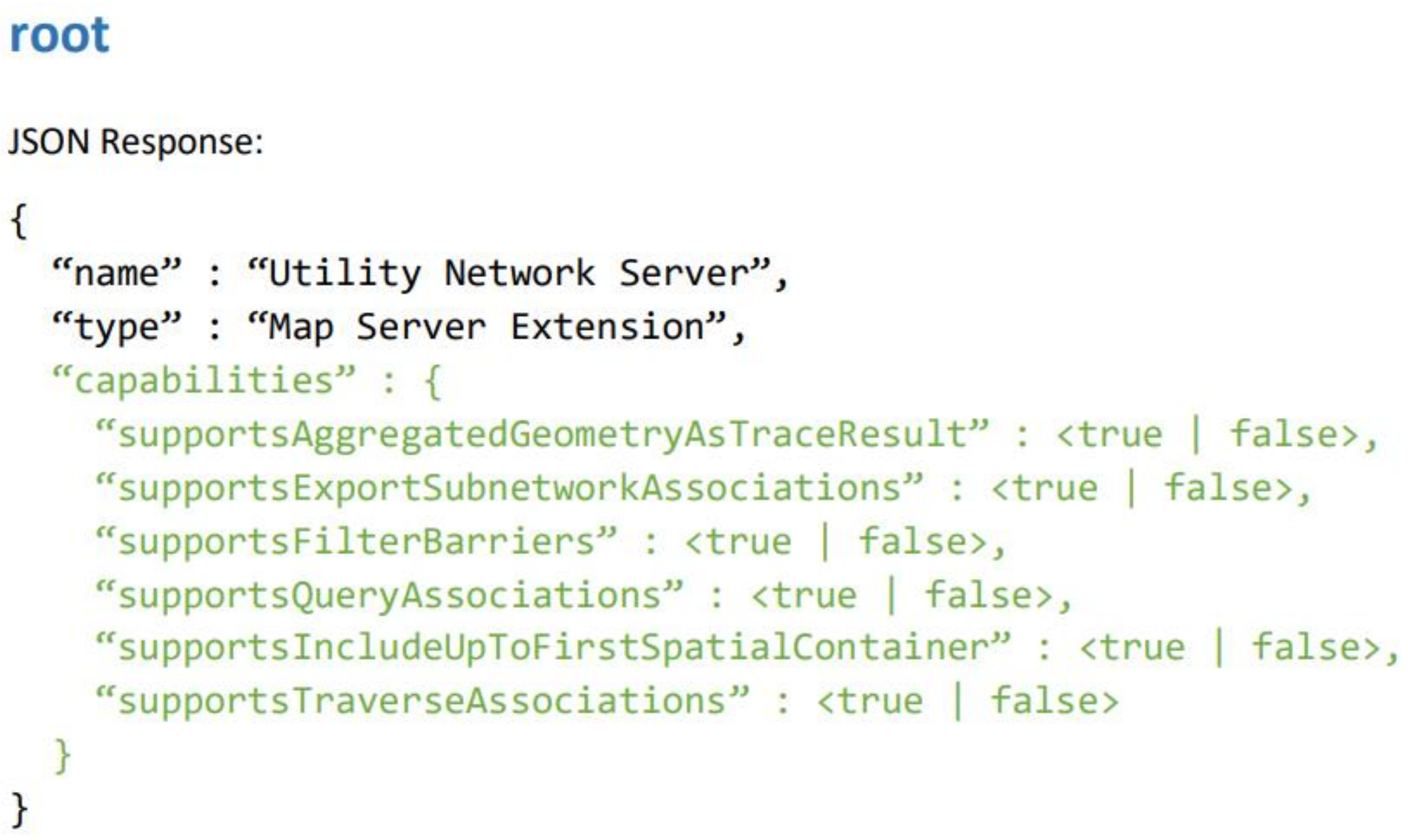
POST only

Validating the network topology for a utility network maintains consistency between feature editing space and network topology space. Validating a network topology may include all or a subset of the dirty areas present in the network. Validation of network topology is supported both synchronously and asynchronously.

Parameter	Details
`f`	Description: Optional parameter representing the output format of the response (default is JSON).
`gdbVersion`	Description: Optional parameter specifying the name of the geodatabase version (default is DEFAULT). Syntax: `gdbVersion=<version>`
`sessionId`	Description: Optional parameter representing the token (guid) used to lock the version. If the calling client is editing a named version, the sessionId must be provided ; if the client is editing DEFAULT, the version may not be locked and the sessionId should not be specified. Syntax : `sessionId=<guid>`
`validateArea`	Description: The envelope of the area to validate.
`returnEdits`	Description: Optional parameter representing whether or not to return the modified features. (default is `false`) Values: “`true`” | “`false`”
`async`	Description: If true, the request is processed as an asynchronous job, and a URL is returned that a client can visit to check the status of the job. See the topic on asynchronous usage for more information. The default is false. Values: “`true`” | “`false`”

Root resource

capabilities



root

JSON Response:

```
{
  “name” : “Utility Network Server”,
  “type” : “Map Server Extension”,
  “capabilities” : {
    “supportsAggregatedGeometryAsTraceResult” : <true | false>,
    “supportsExportSubnetworkAssociations” : <true | false>,
    “supportsFilterBarriers” : <true | false>,
    “supportsQueryAssociations” : <true | false>,
    “supportsIncludeUpToFirstSpatialContainer” : <true | false>,
    “supportsTraverseAssociations” : <true | false>
  }
}
```