

ArcGIS® 9.3

ArcGIS Schematics Designer Tutorial III: Working With the XML Builder



Copyright © 1999-2007 ESRI

All rights reserved

Copyright information

The information contained in this document is the exclusive property of ESRI and its licensor(s). This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

U.S. Government Restricted/Limited Rights

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternatives I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI Trademarks

@esri.com, 3D Analyst, ACORN, ADF, AML, ArcAtlas, ArcCAD, ArcCatalog, ArcCOGO, ArcData, ArcDoc, ArcEdit, ArcEditor, ArcEurope, ArcExplorer, ArcExpress, ArcGIS, ArcGlobe, ArcGrid, ArcIMS, ARC/INFO, ArcInfo, ArcInfo Librarian, ArcInfo—Professional GIS, ArcInfo—The World's GIS, ArcLessons, ArcLocation, ArcLogistics, ArcMap, ArcNetwork, *ArcNews*, ArcObjects, ArcOpen, ArcPad, ArcPlot, ArcPress, ArcQuest, ArcReader, ArcScan, ArcScene, ArcSchool, ArcScripts, ArcSDE, ArcSdl, ArcSketch, ArcStorm, ArcSurvey, ArcTIN, ArcToolbox, ArcTools, ArcUSA, *ArcUser*, ArcView, ArcVoyager, *ArcWatch*, ArcWeb, ArcWorld, ArcXML, Atlas GIS, AtlasWare, Avenue, Business Analyst Online, BusinessMAP, Community, CommunityInfo, Data Automation Kit, Database Integrator, DBI Kit, EDN, ESRI, ESRI—Team GIS, ESRI—The GIS Company, ESRI—The GIS People, ESRI—The GIS Software Leader, FormEdit, GeoCollector, Geographic Design System, ESRI BIS, Geography Matters, Geography Network, GIS by ESRI, GID Data ReViewer, GIS Day, GIS for Everyone, GISData Server, JTX, MapBeans, MapCafé, MapData, MapObjects, Maplex, MapStudio, ModelBuilder, MOLE, NetEngine, PC ARC/INFO, PC ARCPLOT, PC ARCSHELL, PC DATA CONVERSION, PC STARTER KIT, PC TABLES, PC ARCEDIT, PC NETWORK, PC OVERLAY, PLTS, Rent-a-Tech, RouteMAP, SDE, Site Reporter, SML, Sourcebook•America, Spatial Database Engine, StreetEditor, StreetMap, Tapestry, the ARC/INFO logo, the ArcAtlas logo, the ArcCAD logo, the ArcCAD WorkBench logo, the ArcCOGO logo, the ArcData logo, the ArcData Online logo, the ArcEdit logo, the ArcEurope logo, the ArcExplorer logo, the ArcExpress logo, the ArcGIS logo, the ArcGIS Explorer logo, the ArcGrid logo, the ArcIMS logo, the ArcInfo logo, the ArcLogistics Route logo, the ArcNetwork logo, the ArcPad logo, the ArcPlot logo, the ArcPress for ArcView logo, the ArcPress logo, the ArcScan logo, the ArcScene logo, the ArcSDE CAD Client logo, the ArcSDE logo, the ArcStorm logo, the ArcTIN logo, the ArcTools logo, the ArcUSA logo, the ArcView 3D Analyst logo, the ArcView Data Publisher logo, the ArcView GIS logo, the ArcView Image Analysis logo, the ArcView Internet Map Server logo, the ArcView logo, the ArcView Network Analyst logo, the ArcView Spatial Analyst logo, the ArcView StreetMap 2000 logo, the ArcView StreetMap logo, the ArcView Tracking Analyst logo, the ArcWorld logo, the Atlas GIS logo, the Avenue logo, the BusinessMAP logo, the Community logo, the Data Automation Kit logo, the Digital Chart of the World logo, the ESRI Data logo, the ESRI globe logo, the ESRI Press logo, The Geographic Advantage, the Geography Network logo, the MapCafé logo, the GIS Day logo, the MapObjects Internet Map Server logo, the MapObjects logo, the MOLE logo, the NetEngine logo, the PC ARC/INFO logo, the Production Line Tool Set logo, the RouteMAP IMS logo, the RouteMAP logo, the SDE logo, The Geographic Advantage, The World's Leading Desktop GIS, *Water Writes*, www.esri.com, www.esribis.com, www.geographynetwork.com, www.gis.com, www.gisday.com, and Your Personal Geographic Information System are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions.

Other companies and products mentioned herein may be trademarks or registered trademarks of their respective trademark owners.

Contents

Introducing the ArcGIS Schematics Designer Tutorial III	1
Exercise 1: Creating a schematic dataset	5
Exercise 2: Generating a schematic diagram from a sample XML document	6
Exercise 3: Customizing the XML schematic diagram display	14
Exercise 4: Generating new XML diagrams and updating them	28
Appendix: More about the Schematics XML Builder	31

Introducing the ArcGIS Schematics Designer Tutorial III

Welcome to the ArcGIS Schematics Designer Tutorial III.

This tutorial has been designed to help you understand the Schematics XML Builder. XML Builder is used with external applications such as industry applications (for example, Network Engineer), Enterprise Resource Planning (such as, SAP), Product LifeCycle Management (such as, Matrix One) or analysis packages (such as, CYME). It allows you to avoid developing specific interfaces that are costly and difficult to maintain. You can generate schematic diagrams based on eXtensible Marker Language (XML) documents; these documents are built according to the XMLBuilderDiagram XML Schema Definition (XSD) file.

This tutorial is based on XML documents generated from features contained in the Schematics_Designer_Tutorial geodatabase. You will learn to generate the associated XML diagrams and customize the displayed way they are.

Introducing the sample tutorial data

When the ArcGIS® Schematics tutorials were installed on your computer, the \ArcGIS\ArcTutor\Schematics\Designer folder should have been created. This directory contains the Schematics_Designer_Tutorial file geodatabase on which this tutorial is based. The features contained in this geodatabase are organized into a geometric network. Although it would be easier to generate schematic diagrams based on these features by using the predefined Schematics Standard Builder, in this tutorial, you will consider this feature dataset a simple set of feature classes from which XML documents are built. The generated XML documents will be used as input data to generate diagrams using another type of builder, the Schematics XML Builder.

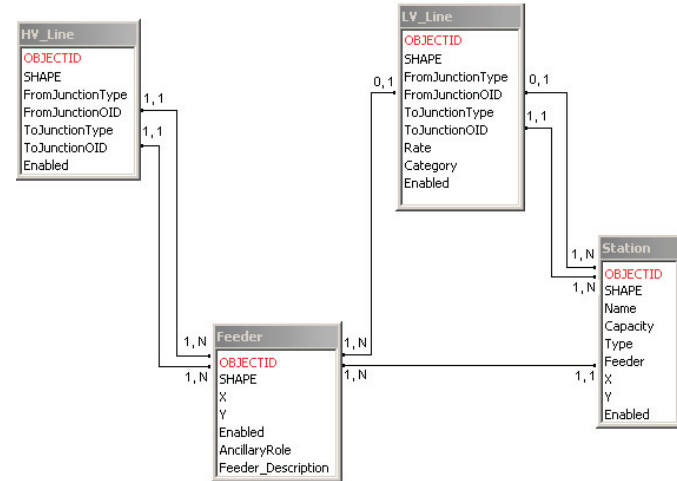
About the geodatabase data from which the XML documents are built

The Schematics_Designer_Tutorial geodatabase contains an electric network composed of two parts, a high-voltage part and a low-voltage part, that connect to each other:

- High-voltage network: Feeder nodes comprise a high-voltage network. In the high-voltage network, feeder nodes are connected to each other through high-voltage links (HV_Line).
- Low-voltage network: Station nodes comprise a low-voltage network. All station nodes depend on one unique feeder node that could be considered a root node for part of the low-voltage network.

Low-voltage links connect a station to another station or a station to a feeder. The link that connects the station to the feeder connects the low-voltage part of the network to the high-voltage part.

The data model can be presented as follows:



About the XML documents built from these feature class tables

An applicative component has been developed to generate XML documents from these feature class tables. The generated XML documents respect the XMLBuilderDiagram XSD file stored in your \ArcGIS\Schematics\XMLSchema folder.

- Feeder point features => Feeders node element in the XML documents

When a feature point stored in the Feeder feature class is extracted from the Schematics_Designer_Tutorial geodatabase and pulled into the generated XML documents, a NodeElement data type for which the ElementTypeName tag is Feeders, is created.

This element is extracted with some attributes coming directly from the feature fields or built from information stored in the geodatabase tables: the ID of the element (ExternalUniqueID), the unique-classID (UCID) and OBJECTID (UODI) of the associated feature, the X- and Y-coordinates. Following is a sample of a Feeder feature in the geodatabase and its associated element in the generated XML document.

OBJECTID *	SHAPE *	Y	X	Enabled	AncillaryRole	Feeder Description
1	Point	2715	-3888	True	0	Feeder 1

The associated Feeders element in the XML document generated by the applicative component:

```
<NodeElement>
  <ElementTypeName>Feeders</ElementTypeName>
  <ExternalUniqueID>Feeder-1</ExternalUniqueID>
  <DatasourceName>XMLDataSource</DatasourceName>
  <UCID>3</UCID>
  <UOID>1</UOID>
  <InitialX>-3888</InitialX>
  <InitialY>2715</InitialY>
</NodeElement>
```

- Station point features => StationsFeeder1, StationsFeeder2 or StationsFeeder3 node elements in the XML documents

For a feature point stored in the Station feature class, depending on the value of its Feeder field (1, 2, or 3), a NodeElement data type for which the ElementTypeName tag is StationsFeeder1, StationsFeeder2, or StationsFeeder3, is created in the built XML documents.

This element is extracted with the same attributes: the ID of the element (ExternalUniqueID), the unique classID (UCID) and OBJECTID (UOID) of the associated feature, the ID of its related container (RelatedContainerID), which depends on the Feeder field value of the element; and the X- and Y-coordinates. Some properties directly coming from the fields in the feature class are also extracted: Feeder, Name, Capacity, and Type.

Following is a sample of a Station feature in the geodatabase and its associated element in the generated XML document.

Station 45 in the Schematics_Designer_Tutorial feature table:

OBJECTID *	SHAPE *	X	Y	Feeder	Name	Capacity	Type	Enabled
45	Point	510	-3299	3	ARRY	5	D	True

The associated StationsFeeder3 element in the XML document generated by the applicative component:

```
<NodeElement>
  <ElementTypeName>StationsFeeder3</ElementTypeName>
  <ExternalUniqueID>Station-45</ExternalUniqueID>
  <DatasourceName>XMLDataSource</DatasourceName>
  <UCID>1</UCID>
  <UOID>45</UOID>
  <RelatedContainerID>Container-3</RelatedContainerID>
  <InitialX>510</InitialX>
  <InitialY>-3299</InitialY>
  <PropertySet>
    <PropertyArray>
      <PropertySetProperty>
        <Key>Feeder</Key>
        <Value>3</Value>
      </PropertySetProperty>
      <PropertySetProperty>
        <Key>Name</Key>
        <Value>ARRY</Value>
      </PropertySetProperty>
      <PropertySetProperty>
        <Key>Capacity</Key>
        <Value>5</Value>
      </PropertySetProperty>
      <PropertySetProperty>
        <Key>Type</Key>
        <Value>D</Value>
      </PropertySetProperty>
    </PropertyArray>
  </PropertySet>
</NodeElement>
```

- HV_Line polyline features => HV_Lines link elements in the XML documents

For a polyline feature stored in the HV_Line feature class in the Schematics_Designer_Tutorial geodatabase, a LinkElement data type, for which the ElementTypeName tag is HV_Lines, is created in the built XML documents. This element is extracted with some attributes coming directly from the feature fields or built from information stored in the geodatabase tables: the ID of the element (ExternalUniqueID), the unique classID (UCID) and OBJECTID (UOID) of the associated feature, and the ID of the origin and extremity schematic element nodes (FromNode and ToNode).

Below is a sample of an HV_Line feature in the geodatabase and its associated element in the generated XML document:

HV_Line 1 in the Schematics_Designer_Tutorial feature table:

OBJECTID	SHAPE	Enabled	FromJunctionType	FromJunctionOID	ToJunctionType	ToJunctionOID	SHAPE_Length
1	Polyline	True	Feeder	1	Feeder	2	2367.0570778316

The associated HV_Lines element in the XML document generated by the applicative component:

```
<LinkElement>
  <ElementTypeName>HV_Lines</ElementTypeName>
  <ExternalUniqueID>HVLine-1</ExternalUniqueID>
  <DatasourceName>XMLDataSource</DatasourceName>
  <UCID>4</UCID>
  <UOID>1</UOID>
  <FromNode>Feeder-1</FromNode>
  <ToNode>Feeder-2</ToNode>
</LinkElement>
```

- LV_Line polyline features => LV_Lines link elements in the XML documents

For a polyline feature stored in the LV_Line feature class in the Schematics_Designer_Tutorial geodatabase, a LinkElement data type for which the ElementTypeName tag is LV_Lines, is created in the built XML documents. This element is extracted with its ID

(ExternalUniqueID), the unique classID (UCID) and OBJECTID of the associated feature, the ID of its origin and extremity schematic element nodes (FromNode and ToNode), and the list of their vertices (Vertices). Then, some properties directly coming from the fields in the feature class are also extracted: Rate, and Category.

Below is a sample of an LV_Line feature in the geodatabase and its associated element in the generated XML document.

OBJECTID	SHAPE	FromJunctionType	FromJunctionOID	ToJunctionType	ToJunctionOID	Rate	Category	ListPoints	SHAPE_Length	Enabled
1	Polyline	Feeder		1	Station	1	123456	Small	3;-3388.0108;1748.67013904	True

The associated LV_Lines element in the XML document generated by the applicative component:

```
<LinkElement>
  <ElementTypeName>LV_Lines</ElementTypeName>
  <ExternalUniqueID>LVLine-1</ExternalUniqueID>
  <DatasourceName>XMLDataSource</DatasourceName>
  <UCID>5</UCID>
  <UOID>1</UOID>
  <FromNode>Feeder-1</FromNode>
  <ToNode>Station-1</ToNode>
  <Vertices>
    <Vertex>
      <X>-3783.59998</X>
      <Y>2093.72369</Y>
    </Vertex>
    <Vertex>
      <X>-4461.75271</X>
      <Y>2828.38914</Y>
    </Vertex>
  </Vertices>
  <PropertySet>
    <PropertyArray>
      <PropertySetProperty>
        <Key>Rate</Key>
        <Value>123456</Value>
      </PropertySetProperty>
      <PropertySetProperty>
        <Key>Category</Key>
        <Value>S</Value>
      </PropertySetProperty>
    </PropertyArray>
  </PropertySet>
</LinkElement>
```

- An other data type in the XML documents, the Containers drawing elements

A DrawingElement data type, for which the ElementTypeName tag is Containers, can also be found in the XML documents built by the applicative component. This type of elements is associated with Feeder features in the Schematics_Designer_Tutorial database. Such elements are related to Stations elements, for which the RelatedContainerID tag specified is the Containers ID. They will display as containers around schematic stations in the generated diagrams.

Below is a sample of Containers DrawingElement in the generated XML documents:

```
<DrawingElement>
  <ElementTypeName>Containers</ElementTypeName>
  <ExternalUniqueID>Container-1</ExternalUniqueID>
  <DatasourceName>XMLDataSource</DatasourceName>
  <UCID>3</UCID>
  <UOID>1</UOID>
</DrawingElement>
```

Specifications for the tutorial schematic project

The purpose of this tutorial is to create a diagram type to generate schematic diagrams from any XML document containing node, link, and drawing elements based on the data types described in the previous sections. The schematic project conception is organized into four exercises:

- In the first exercise, you will create a schematic dataset that contains all the data related to your schematic project.
- During the second exercise, you will create the schematic diagram type that will be used to generate schematic diagrams from the Feeders, StationsFeeder1, StationsFeeder2, StationsFeeder3, HV_Lines, LV_Lines, and Containers elements that are expected in the generated XML documents. This diagram type will work with the XML Builder. The SampleNetworkData.xml file stored in the \ArcGIS\ArcTutor\Schematics\Schematics folder will be used as a sample for the first parts of the tutorial.
- The purpose of the third exercise is to learn how to customize the schematic diagrams display according to the attributes contained in the XML documents.

- In the fourth exercise, you will generate new schematic diagrams and learn how to use the Update Diagram command on XML diagrams. You will also learn how to finalize your parameterization so your schematic dataset is ready to be provided to the final users.

At the end of the tutorial, you will find sections in the appendix that cover some specifics of the XML Builder:

- Generally, the XML documents used to generate and update XML Builder diagrams are built by an external component that extracts all the required information from a database or several databases and pulls it into XML documents. In the first section of the appendix, you will learn more about how such a component can be developed to be automatically triggered when you request a generation/update of an XML diagram.
- In the second section, you will learn about different ways to manage associations between features and schematic elements contained in XML Builder diagrams.
- Next, you will learn more about the type of attributes usually specified on element types managed by the XML builder diagram type.
- Finally, you will learn how to use the Import From Feature Layers command for diagram types managed by the XML Builder.

Exercise 1: Creating a schematic dataset

In this exercise, you will create a new schematic dataset for your diagrams.

Creating a schematic dataset

1. Start ArcCatalog™.
2. Click the Tools menu and click Extensions.
3. Check the Schematics check box.
4. Click Close.
5. In your \ArcGIS\ArcTutor\Schematics\Designer folder, create a new file geodatabase and type a new name for this geodatabase (“TutorialIII_SchematicDatabase”, for example).

Note: You can also create a new personal geodatabase or use an existing ArcSDE® geodatabase.

6. Right-click this geodatabase and point to New, then click Schematic Dataset.
A new schematic dataset appears in the Catalog tree.
7. Type a new name for this schematic dataset (“TutorialIII”, for example).

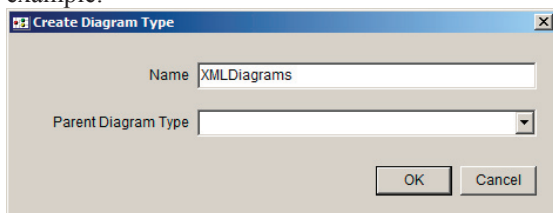
Note: An ArcEditor™ or ArcInfo® license is required for schematic dataset creation.

Exercise 2: Generating a schematic diagram from a sample XML document

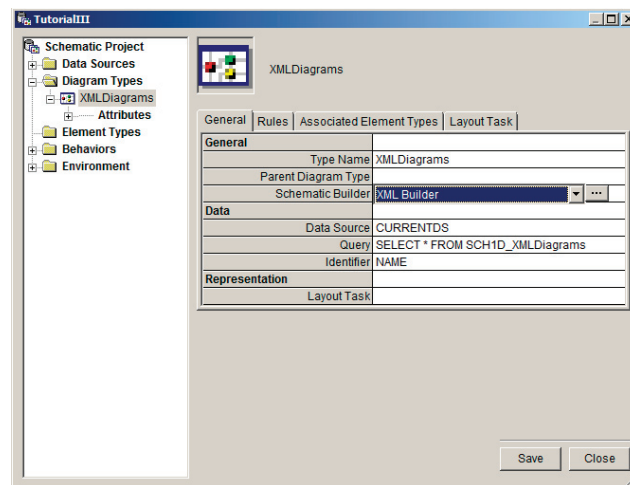
The purpose of this exercise is to create a new schematic diagram type that works with the XML Builder to generate schematic diagrams in ArcMap™ from XML data contained in XML documents. The SampleNetworkData.xml document stored in your \ArcGIS\ArcTutor\Schematics\Schematics folder can be considered a model document for the schematic project setting.

Creating the XML Builder diagram type

1. Start ArcCatalog and activate the Schematics extension.
2. Browse to the geodatabase that contains the TutorialIII schematic dataset you created in exercise 1.
3. Right-click this schematic dataset and click Edit Project. Designer starts and your schematic project opens. Since no schematic diagram type already exists in this schematic dataset, the Diagram Type Assistant automatically opens.
4. Click Cancel.
5. Right-click the Diagram Types entry in the Designer tree and click Create. The Create Diagram Type dialog box opens.
6. Type a name for your schematic diagram type (“XMLDiagrams”, for example).



7. Click OK.
8. Click the General tab and click XML Builder in the Schematic Builder drop-down list.
The General tab content now appears as follows:



Creating the element types used in the XMLDiagrams diagram type

Now, you will create the element types used to generate and update diagrams of this type. Regarding the specifications of these XML documents, there are

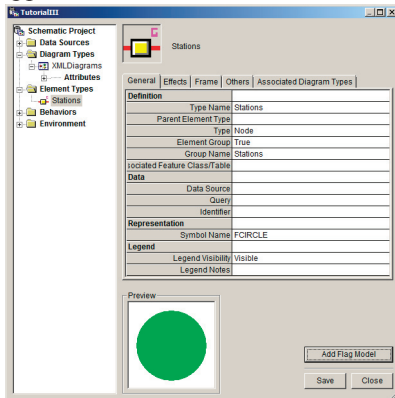
- Four possible types of node elements: Feeders, StationsFeeder1, StationsFeeder2, and StationsFeeder3
- Two possible types of link elements: LV_Lines and HV_Lines
- One type of drawing element: Containers

Creating the node element types

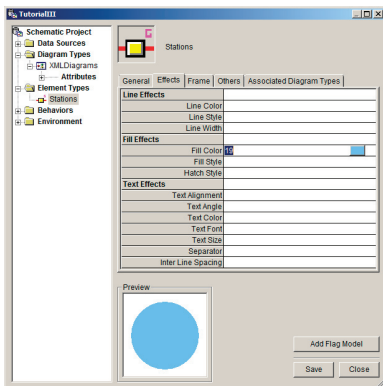
If you open the SampleNetworkData.xml document (stored in your \ArcGIS\ArcTutor\Schematics\Schematics folder) and analyze the NodeElement items for which ElementTypeName are StationsFeeder1, StationsFeeder2 or StationsFeeder3, you can see that all their attributes and PropertySet properties are exactly the same. The only difference concerns their related feeder. All StationsFeeder1 elements have a Feeder PropertySet property value of 1; for StationsFeeder2, the Feeder is 2; and for StationsFeeder3, it is 3. Instead of creating three different element types, you can manage a unique Stations node element type in the schematic project and avoid creating the same properties on different

node element types.

1. Right-click the Element Types entry in the Designer tree and click Create. The Create Element Type dialog box opens.
2. Type “Stations” in the Name text box, keep Node selected in the Type drop-down list, then click OK.
3. Click the Symbol Name field and click the Select Symbol button to open the CGM Symbol Browser.
4. Choose the FCIRCLE CGM symbol (the third symbol in the list) and click OK. The Stations node element type’s General tab content now appears as follows:

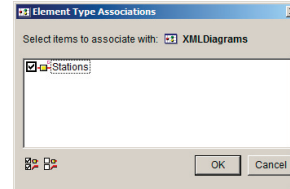


5. Click the Effects tab, click the Fill Color field, then click the Select Color button to open the Colors editor.
6. Click the desired color and click OK.



You must associate this Stations node element type with your diagram type.

7. Right-click the XMLDiagrams entry in the Designer tree and click Edit Associations. The Element Type Associations dialog box opens.
8. Check the Stations check box.

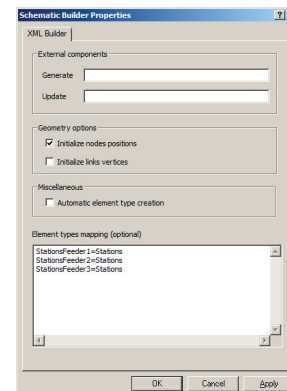


9. Click OK.

You need to parameterize the diagram type builder so that when it detects StationsFeeder1, StationsFeeder2, or StationsFeeder3, it matches it with the Stations element type.

10. Click the General tab, click the Schematic Builder parameter, then click the Schematic Builder Properties button. The XML Builder tab appears.
11. The Element types mapping section is the area in which such mapping rules must be specified. Type the three following lines in that section as follows:
 “StationsFeeder1=Stations”
 “StationsFeeder2=Stations”
 “StationsFeeder3=Stations”

The XML Builder tab on the Schematic Builder Properties dialog box appears as follows:



Note: Conversely, if the XML documents only contained a Stations node type, you could match several element types in the schematic project. In that case, you would have to use a particular property value to match the ElementTypeName found in the XML document and those you would want to have in your schematic project. For example, in the SampleNetworkData.xml document, replace the StationsFeeder1, StationsFeeder2, and StationsFeeder3 node types with a unique Stations node type, specifying the following rules in the Element type mapping section:

“Stations=StationsFeeder1 (Feeder=”1”)”

“Stations=StationsFeeder2 (Feeder=”2”)”

“Stations=StationsFeeder3 (Feeder=”3”)”

This will cause the StationsFeeder1, StationsFeeder2, and StationsFeeder3 element types to match the Stations nodes found in the XML document for which the Feeder attribute value is 1, 2, or 3. In this case, instead of creating a unique Stations node element type in your schematic project, you would create the StationsFeeder1, StationsFeeder2, and StationsFeeder3 element types.

12. Click OK to close the Schematic Builder Properties dialog box.
13. Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.

You must now create the element type that will be used to display the Feeders elements contained in the XML documents.

14. Right-click Element Types entry in the Designer tree and click Create. The Create Element Type dialog box opens.
15. Type “Feeders” in the Name text box, keep Node selected in the Type drop-down list, then click OK.
16. Click the Symbol Name field and click the Select Symbol button to open the Symbol Browser.
17. Choose the FSQUARE CGM symbol (the eighth symbol in the list) and click OK.
18. If you want to modify the default color of this symbol, click the Effects tab, click the Fill Color field, click the Select Color button to open the Colors editor; click the desired color, then click OK.

Next, associate this new node element type with the diagram type.

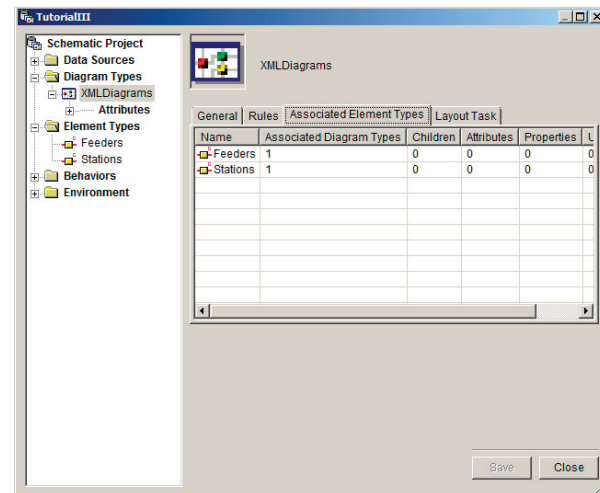
19. Right-click the XMLDiagrams entry in the Designer tree and click Edit Associations.
The Element Type Associations dialog box opens.

20. Check Feeders and click OK.

Note: Feeders is the name of the element type you have just created in your project to manage the Feeders node elements that will be found in the XML documents. Because there is an exact correspondence between the element type name in the schematic project and the ElementTypeName item in the XML documents, no specification needs to be set in the Element types mapping section on the Schematic Builder Properties dialog box.

21. Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.

If you click the Associated Element Types tab related to your XMLDiagrams diagram type, the content now appears as follows:

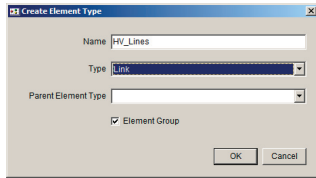


Creating the link element types

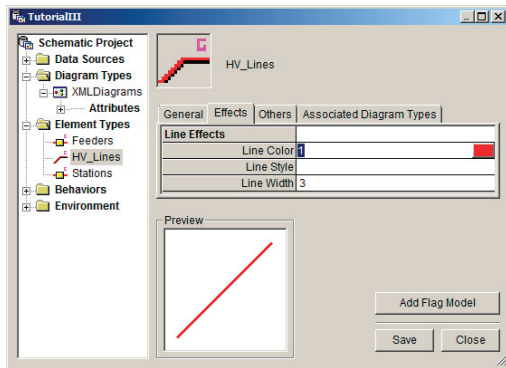
Two different ElementTypeName items are possible in XML documents, HV_Lines and LV_Lines. You must create the associated element types in your schematic project.

1. Right-click the Element Types entry in the Designer tree and click Create.
The Create Element Type dialog box opens.
2. Type “HV_Lines” in the Name text box.

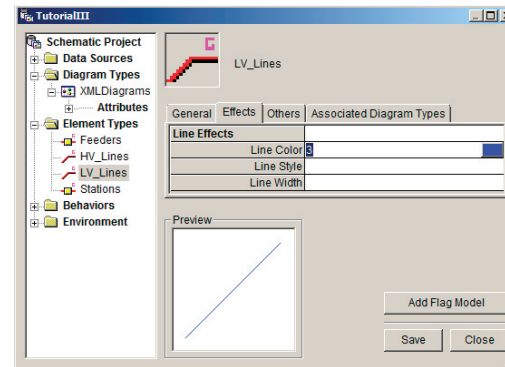
- Click the Type drop-down arrow and click Link.



- Click OK.
The new HV_Lines link element type entry appears in the Designer tree.
- Click the Effects tab to set a default symbology for this type of link.
For example:
Click the Line Color field and click the Select Color button to open the Colors editor. Select the desired color and click OK.
Type "3" for the Line Width parameter.

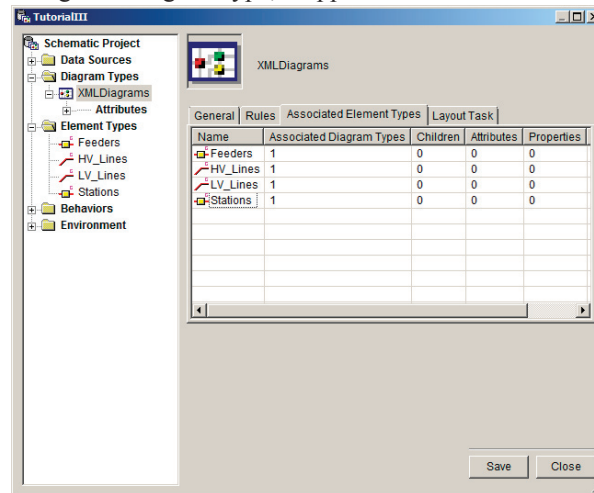


- Associate the newly created HV_Lines element type with the XMLDiagrams diagram type.
- Repeat steps 1 through 4 to create the second LV_Lines link element type that will be used to display any LV_Lines LinkElement found in the XML documents.
- Click the Effects tab to set a default symbology for this second type of link.
For example, click the Line Color field and click the Select Color button to open the Colors editor. Select the desired color and click OK.



- Associate this LV_Lines element type with the XMLDiagrams diagram type.

If you click the Associated Element Types tab related to your XMLDiagrams diagram type, it appears as follows:

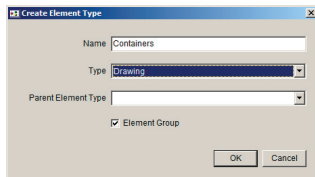


- Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.

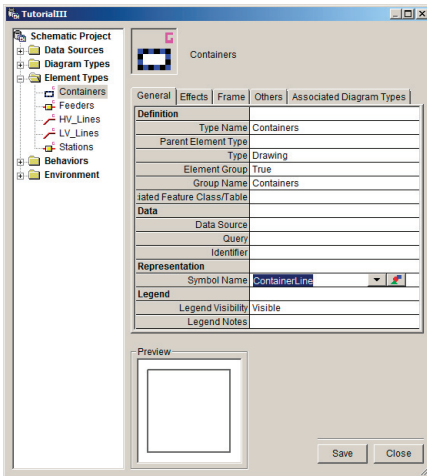
Creating the drawing element type

For the XML document specifications, you need to create one more element type, Containers. The required steps to create are detailed below:

1. Right-click the Element Types entry in the Designer tree and click Create. The Create Element Type dialog box opens
2. Type “Containers” in the Name text box.
3. Click the Type drop-down arrow and click Drawing.



4. Click OK.
The new Containers drawing element type appears in the Designer tree.
5. Click the Symbol Name parameter on the General tab and click the Select Symbol button. The Symbol Browser opens.
6. Click the ContainerLine CGM symbol (the second one in the list) and click OK.



7. Associate the newly created Containers element type with the XMLDiagrams diagram type.

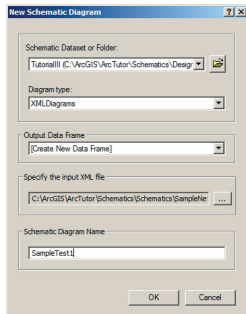
8. Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.

All the possible element types in the XMLDiagrams diagram type are now created. You can test the generation of a first diagram based on the SampleNetworkData.xml document. In that XML document, there are many PropertySet properties that are specified on some elements. To be able to use these property values to customize the schematic diagrams display, you will activate the Automatic Element Type Creation mode from the Schematic Builder Properties dialog box. When this mode is activated, Schematics automatically completes the element types definition in the schematic project by creating PropertySet attributes associated with each PropertySet property found on the elements contained in the XML document.

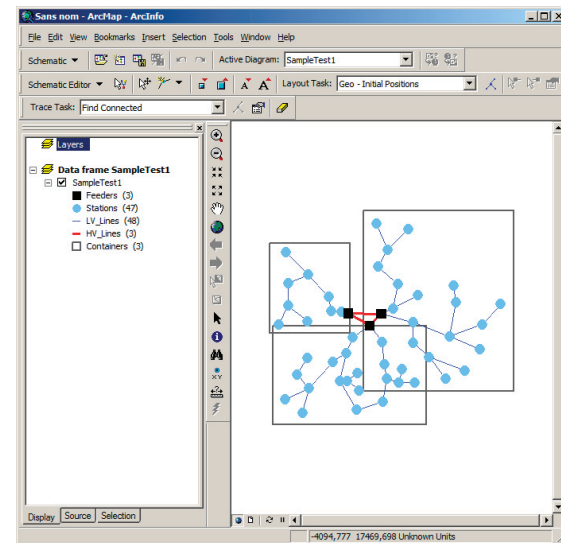
9. Click the XMLDiagrams entry in the Designer tree.
10. Click Schematic Builder on the General tab and click the Schematic Builder Properties button. The Schematic Builder Properties dialog box opens.
11. Check the Automatic element type creation check-box.
12. Click OK.
13. Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.
14. Exit Schematics Designer.

Generating the first diagram test

1. Start ArcMap and activate the Schematics extension.
2. Load the Schematic, Schematic Editor, and Schematic Network Analyst toolbars.
3. Click Generate New Schematic Diagram from the Schematic toolbar. The New Schematic Diagram dialog box opens
4. For the Schematic Dataset or Folder parameter, click the browse button to select the TutorialIII schematic dataset in the TutorialIII_SchematicDatabase database you should created in exercise 1.
5. For the Specify the input XML file parameter, click the ellipsis button and browse to the SampleNetworkData XML file in your \ArcGIS\ArcTutor\Schematics\Schematics folder.
6. Type a name for the diagram that will be generated in the Schematic Diagram Name text box. For example, type "SampleTest1".



7. Click OK.
The diagram appears as follows:



Analyzing the generated diagram and its content

Analyzing the diagram content

The input XML document used to generate the diagram contains elements for which the ElementTypeName tags are Feeders, StationsFeeder1, StationsFeeder2, StationsFeeder3, LV_Lines, HV_Lines and Containers. If the Element types mapping section has been correctly specified during step 11, page 8 and all the element types created manually in the previous steps have been specified with the correct names, five legend entries will be displayed in the table of content: Feeders, Stations, LV_Lines, HV_Lines, and Containers.

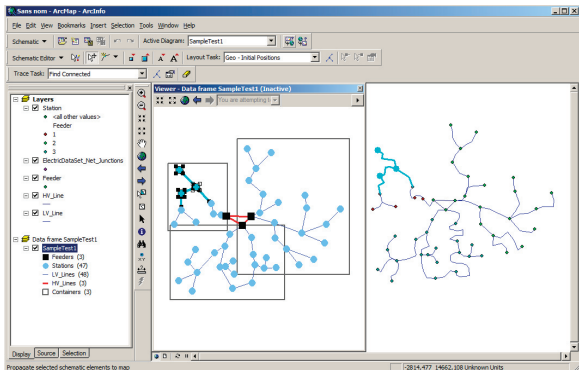
The symbology specified by default for each of these element types should be the one used to display the generated diagram's elements.

Because the input XML document specified the Container-1, Container-2, and Container-3 elements as related containers for some stations, three Containers drawing elements are automatically displayed around their related station nodes. Moving a container causes its related stations to be automatically moved; moving any station causes its container to be automatically redrawn.

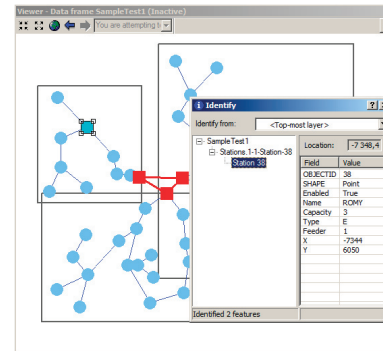
Checking the associations between schematic elements and features

The SampleNetworkData XML document has been generated by an applicative component from the features contained in the Schematics_Designer_Tutorial geodatabase. The DataSourceName, UCID, and UOID items for each element data type contained in the XML document specify the name of the data source that must be used to connect to the geodatabase that contains the associated feature, its classID and ObjectID. The following steps detail how to check that the associations have been correctly reported on the schematic elements contained in the generated diagram.

1. Click Windows and click Viewer. The diagram is displayed in a viewer window.
2. Activate the empty Layers data frame. Click Add Data and browse to the Schematics_Designer_Tutorial geodatabase in your \ArcGIS\ArcTutor\Schematics\Designer folder.
3. Select the ElectricDataset feature dataset and click Add so that all these features are displayed in the Layers data frame.
4. Zoom in to the features and place the Viewer Window so your schematic diagram and the associated features are displayed side by side, with the Layers data frame as the active data frame.
5. Click the Select Schematic Elements button on the Schematic toolbar and select a set of schematic elements in the active schematic diagram.
6. Click the Propagate Schematic Selection To Map button on the Schematic toolbar. The associated features displayed in the geographic active data frame are automatically selected.



7. Click the Select Features tool on the Tools toolbar and define a new selection set of features in the active data frame.
8. Click the Propagate Map Selection To Schematic button on the Schematic toolbar. The associated schematic elements displayed in the generated schematic diagram are automatically selected.
9. Next, click the Identify tool on the Tools toolbar and click one of the schematic elements contained in the active schematic diagram. The Identify dialog box opens and displays properties related to the selected element and those related to its associated feature.



Note: The associations between schematic elements contained in XML Builder diagrams and features work when the three DataSourceName, UCID, and UOID items are correctly specified in the input XML document or when, in the schematic project, there are associated feature classes specified for the element types and, at least, the UOID information is provided through the input XML documents.

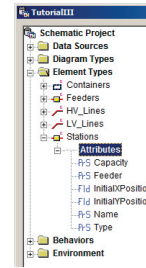
10. Remove the Data frame SampleTest1 from the TOC.
11. Click the Schematic menu and click Options. The Schematic Options dialog box opens. Click the Multi Views tab and check the Automatically open schematic diagrams in a viewer window option.
12. Click Apply and click OK. Each time a schematic diagram is generated or opened, it will automatically be displayed in a viewer window.
13. Type a name for the MXD file ("TutorialIII.mxd", for example), save it, and exit ArcMap.

Analyzing the schematic project parameters automatically added during the diagram generation

1. Edit the TutorialIII schematic dataset within Schematics Designer.
2. Expand the Element Types entry in the Designer tree.

During this diagram generation, the Automatic element type creation running mode was activated. When working in this mode, Schematics automatically creates the schematic element classes and schematic element types associated with each ElementTypeName, of which at least one element is found in the input XML documents, when these element types don't already exist in the schematic project. If the Element types mapping section on the Schematic Builder Properties dialog box has been correctly specified during steps 11, page 8, and all the element types created manually in the previous steps have been specified with the correct names, no new element type should have been created in the schematic project. You should still have five element types in the Designer tree: Containers, Feeders, HV_Lines, LV_Lines, and Stations. On the other hand, PropertySet attributes related to PropertySet properties detected on the elements contained in the input XML document should have been automatically created because the Automatic element type mode is activated.

3. Double-click the Stations entry in the Designer tree and expand Attributes.
 - Four property set attributes—Capacity, Feeder, Name, and Type—are now displayed for the Stations element type. They have been automatically created because PropertySet properties with these key values have been found in the input XML document analyzed during the diagram generation.
 - Moreover, the InitialXPosition and InitialYPosition field attributes have also been created. These attributes are linked to the INITIALX and INITIALY fields in the associated schematic table. These fields are automatically filled during diagram generation when the InitialX and InitialY items are found for node elements contained in the input XML document.



Note: The InitialXPosition and InitialYPosition field attributes have also been created on the Feeders and Containers element types.

4. Double-click the LV_Lines entry and expand Attributes.
 - PropertySet attributes related to the Category and Rate key values of the PropertySet properties specified for the LV_Lines link elements in the input XML document have also been created during the generation.
 - The OriginNode and ExtremityNode field attributes linked to the FROMNODE and TONODE fields in the associated schematic table have also been automatically created. These fields are automatically filled during diagram generation by the FromNode and ToNode items found on each link element contained in the input XML document.
 - The InitialListPoints propertyset attributes have also been created. This attribute is linked to the InitialListPoints PropertySet property stored in the associated schematic table. This PropertySet property is automatically filled during diagram generation when the Vertices item is found for link elements contained in the input XML document.

Note: The OriginNode, ExtremityNode, and InitialListPoints attributes have also been created on the HV_Lines link element types.

5. Double-click Data Sources in the Designer tree. There is a DataSources section in the SampleNetworkData.xml document, and the XMLDataSource is specified in this section. The related XMLDataSource schematic data source has been automatically created during the diagram generation.

Exercise 3: Customizing the XML schematic diagram display

In this exercise, you will learn how to label and symbolize the schematic elements contained in your schematic diagram according to the PropertySet properties imported in the schematic tables during the XML diagrams generation. You will become familiar with property creation and use.

This exercise is organized into three parts:

- In the first part, you will create textual properties for labeling your schematic nodes and links.
- In the second part, you will learn to create discrete properties. You will also learn how to create a composed Computer Graphic Metafile (CGM) symbol to display a node element type and use this symbol to specify interesting subsymbols' effects to represent your discrete filters.
- In the third part, you will learn about bounded properties.

Displaying labels for LV_Lines and Stations

In this part, you will create two textual properties.

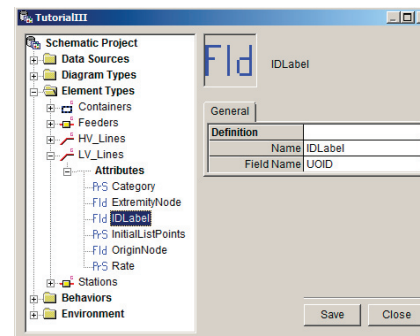
- The Label1 property will display the ObjectID of the associated feature in blue, centered below each LV_Lines schematic link. The ObjectID of the associated feature is stored in the UUID field of the schematic table.
- The Label2 property will display a label for each station node. This label will merge the values of two database fields, Name and Feeder, formatted as follows: Name (F-Feeder). The property label will be centered at the top of the station.

Displaying a label below the schematic links

The Label1 property will use the UUID field stored in the schematic table. Begin by creating an attribute corresponding to this UUID field.

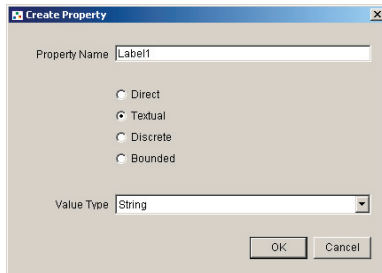
1. Right-click the LV_Lines entry in the Designer tree and click Create Attribute. The Create Element Type Attribute dialog box opens.
2. Type the name of the new attribute in the Name text box ("IDLabel", for example), click Field Attribute in the Type drop-down list, then click OK.
3. The Identifier Editor dialog box opens. Click the UUID field in the Fields list, click the down arrow button, then click OK.

The Identifier Editor and Create Element Type Attribute dialog boxes close. The General tab related to the newly created IDLabel attribute now appears as follows:



Next, you will create the textual property that will manage this attribute display.

4. Right-click the LV_Lines entry and click Create Property. The Create Property dialog box opens.
5. Type a name for the new property in the Property Name text box ("Label1", for example).

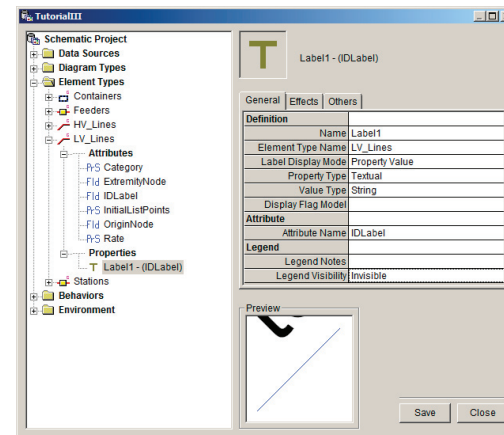


6. Since the default parameters set on this dialog box (Textual property type and String value type) are those you need, click OK. The Create Property dialog box closes.

Under the LV_Lines element type entry, a new Properties entry is automatically created. The newly created Label1 property appears under this main entry.

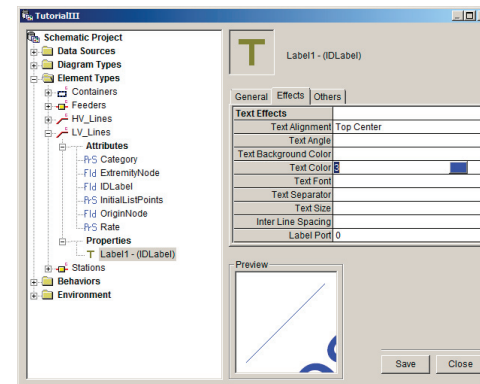
Now, you need to relate the IDLabel attribute with the newly created property.

7. Click the General tab.
8. Click IDLabel in the Attribute Name drop-down list.
9. Click Invisible in the Legend Visibility drop-down list, so this label is not displayed in the Legend subwindow and in the ArcMap TOC (when the diagram is opened in the ArcMap environment). The Label1 textual property's General tab content now appears as follows:



Next, you need to specify how the label will be displayed. The label must be in blue and centered below each link.

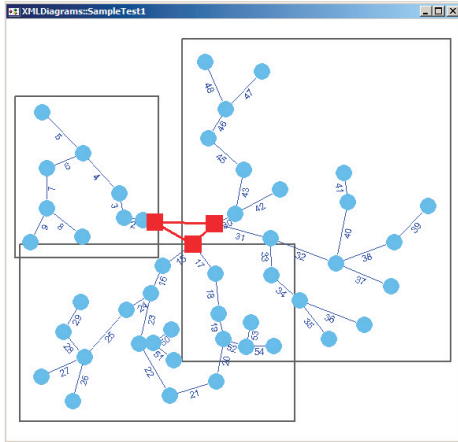
10. Click the Effects tab where the text effects are assembled.
11. Click Top Center in the Text Alignment drop-down list.
12. Click the Text Color field and click the Select color button to open the Color Editor. Choose the desired blue color and click OK.



13. Click Save to save the specified parameters. If the Schematic Project Verification Checklist dialog box opens, click Yes.

If you want to test the Label1 property definition result, click Close. Click File on the main menu and click Diagram Toolbar. On the new toolbar, click Open Schematic Diagram. The Select diagram to open dialog box opens. Select the XML diagram generated during exercise 2.

The result should look similar to the following screen shot:



Displaying a label for the schematic nodes

The Label2 property must use the Name and Feeder PropertySet properties stored in the Stations schematic table as formatted parameters to display a label formatted as follows: Name (F-Feeder).

The Name and Feeder PropertySet attributes have already been created because they exist as PropertySet properties in the SampleNetworkData XML document used to generate the diagram in exercise 2.

You must now create the formatted attribute that will concatenate the Name and Feeder PropertySet attributes to generate values formatted as follows: Name (F-Feeder).

1. Right-click the Attributes entry related to the Stations element type and click Create. The Create Element Type Attribute dialog box opens.
2. Type "LabelName" in the Name text box, click the Type drop-down arrow and choose Formatted Attribute, then click OK.
3. On the General tab related to the newly created attribute, you need to specify the format that will be used to build the new attribute value (Format parameter). The format is generally a string combining words or characters in C language-compliant format. For example, if your formatted attribute must generate values by concatenating two attribute values, each being related to a string field, and if these two attribute values must be separated by the ">>" characters,

the format will be: %s >> %s. The %s is used to indicate that the attribute values are string values. Use %e for double and %d for integer.

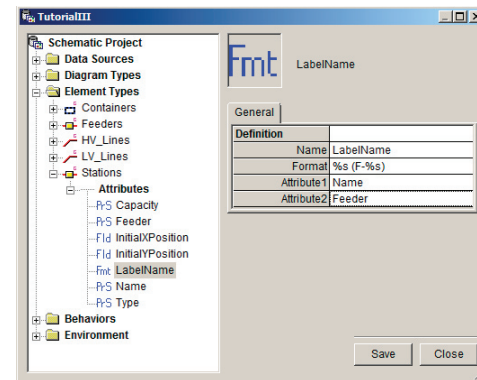
In this example, type "%s (F-%s)" for the Format parameter.

4. Press Enter.

Designer automatically detects that two attributes are needed to build the formatted attribute. The new Attribute1 and Attribute2 parameters are automatically displayed.

5. Click Name in the Attribute1 drop-down list.
6. Click Feeder in the Attribute2 drop-down list.

The LabelName attribute's General tab now appears as follows:



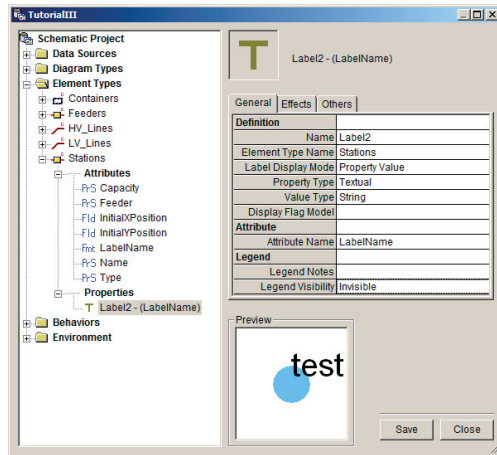
As the label values you want to display for your station nodes are defined, you can now create the textual property that will manage this information display.

7. Right-click the Stations element type entry and click Create Property. The Create Property dialog box opens.
8. Type "Label2" in the Name text box and click OK.

Under the Stations element type entry, a new Properties entry is automatically created. The newly created Label2 property appears under this main entry. Now, you need to relate the LabelName attribute to the newly created property.

9. Click the General tab.
10. Click LabelName in the Attribute Name drop-down list.
11. Click Invisible in the Legend Visibility drop-down list so this label

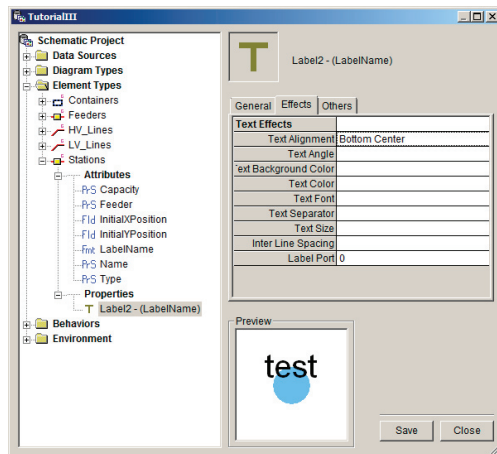
is not displayed in the Legend subwindow and in the ArcMap TOC (when the diagram is opened in the ArcMap environment).



Next, you need to define the property label effects. In this example, the label must be centered at the top of each node.

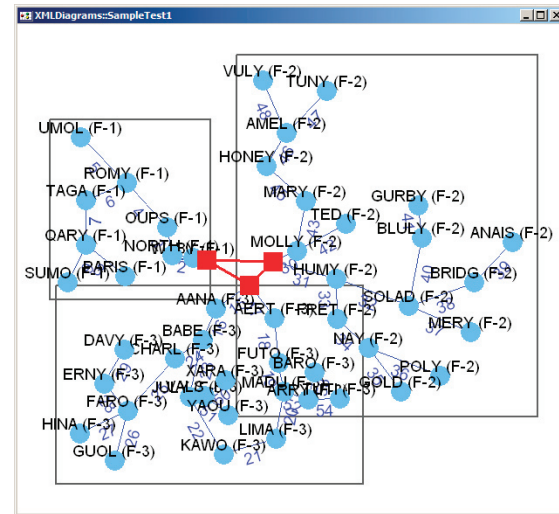
12. Click the Text Alignment drop-down list on the Label2 property's Effects tab and click Bottom Center.

Bottom Center text alignment means that the element node will appear centered at the bottom of the property label; that is, the label will be centered at the top of each node.



13. Click Save to save your schematic project parameters.

Close the Designer Editor window and test your network display. The result should be similar to the following screen shot:



As the size of the displayed labels is too large, you can redefine the Text Size parameter value available from the Schematic Project's Effects tab so all label sizes are decreased.

14. Switch to design mode.
15. Click Schematic Project and click the Effects tab in the Designer tree.
16. Type "0.5" in the Text Size field.
17. Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.
18. Close the Designer Editor window and test the network display again.

Creating discrete properties for Stations and LV_Lines

A discrete property has specified values that belong to a finite set of either numbers or alphanumeric codes. The values for the property are generally returned by an attribute that must be associated with the property.

Two discrete properties will be created:

- You will first symbolize the LV_Lines links display according to the Category PropertySet property values. Each link has S, M, or B for Small, Medium, or Big value respectively. A discrete property will be created, and display rules will be defined for the discrete filters based on these categories.
- After defining a composed CGM symbol to represent the Stations nodes, you will learn how to define subsymbol effects to symbolize the station nodes' display according to the Type PropertySet property values. Since this PropertySet property stores discrete values (A, B, C, D, E, or F), a discrete property with six discrete filters (one for each value) will be created. Subsymbol display rules will then be defined for each discrete filter.

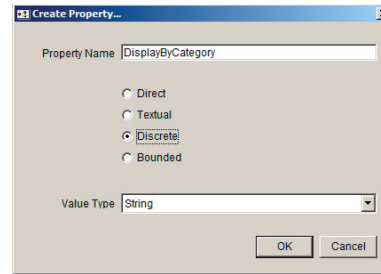
Symbolizing the schematic links according to their category

The Category PropertySet property stored in the schematic table during the diagram generation returns the category of each link. Each link has an S, M, or B category value.

In this section, you need to represent this information graphically. You will create the DisplayByCategory discrete property composed of three discrete filters with each filter corresponding to one category value. This property will display LV_Lines links according to the Category field as follows:

Category value	Filter graphic effect
S	Line width: 1.0
M	Line width: 2.0
B	Line width: 4.0

1. Right-click the Properties entry displayed below the LV_Lines element type and click Create. The Create Property dialog box opens.
2. Type "DisplayByCategory" in the Property Name text box, check the Discrete option, then keep String for the Value Type parameter.



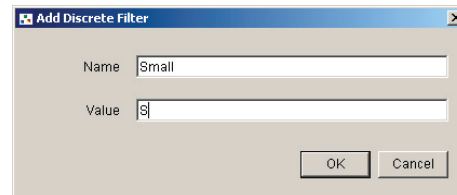
3. Click OK.
The Create Property dialog box closes.

Next, you have to associate this new property with the Category PropertySet attribute that already exists for the LV_Lines element type.

4. On the DisplayByCategory property's General tab, click Category in the Attribute Name drop-down list.

Now, you will create the discrete filters related to each Category field's value.

5. Click the Add Filter button displayed on the lower right corner of the Designer Editor window. The Add Discrete Filter dialog box opens.
6. Type a name for the discrete filter ("Small", for example), and specify the value for the discrete filter ("S" in this example). This is one of the values used by the Category field stored in the LV_Line table.

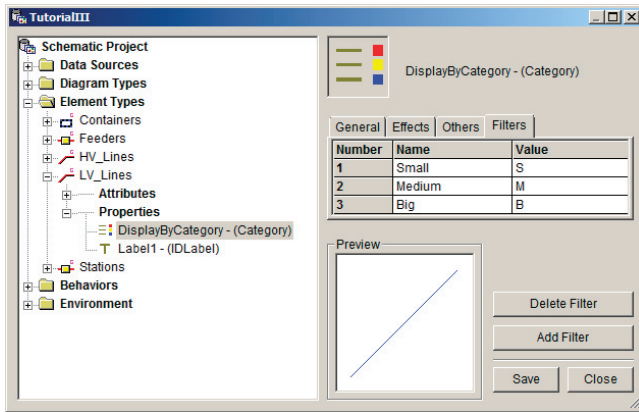


7. Click OK.

The Filters tab appears for the DisplayByCategory property. A line corresponding to the discrete filter you have just created is displayed in this tab.

8. Repeat steps 5 through 7 to create two new discrete filters for the M and B values of the Category PropertySet property attribute's values.

The Filters tab now appears as follows:

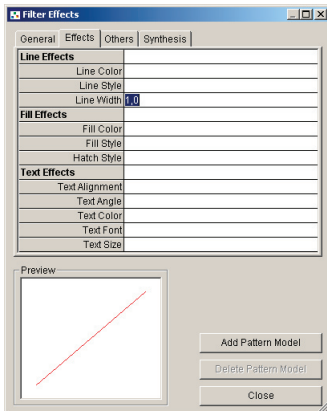


You need to define the graphic effects related to each discrete filter.

- Click the Filters tab, click the first discrete filter parameter (Small), then click the ellipsis button.

The Filter Effects dialog box opens. The links that will verify this Small discrete filter must be displayed with a line width of 1.0.

- Click the Effects tab on the Filter Effects dialog box and type “1,0” in the Line Width field.



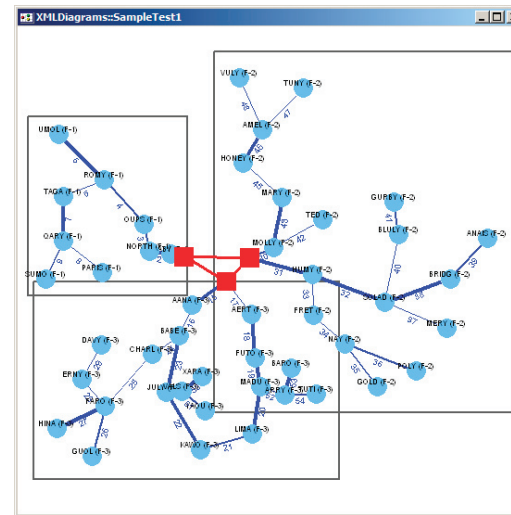
- Click Close to close the Filter Effects dialog box.

- Repeat steps 9 through 11 to define the effects related to Medium and Big discrete filters as follows:

- Type “2,0” for the Line Width parameter related to the Medium filter.
- Type “4,0” for the Line Width parameter related to the Big filter

- Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.

If you open the XML diagram, it should appear as follows:




Symbolizing the schematic nodes according to their type

The Type PropertySet property stored in the Stations schematic table during the diagram generation returns the type of each station. Each station node is either an A, B, C, D, E, or F type. In this section, you will create the DisplayByType discrete property that is composed of six discrete filters, with each filter corresponding to one Type value.

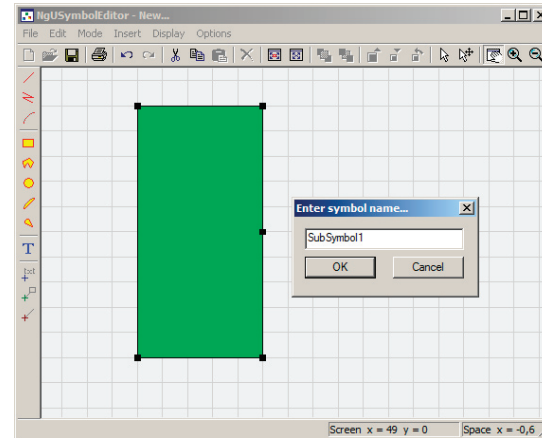
In this example, you will create a CGM symbol based on the default node symbol you created in exercise 2, for which you will add two subsymbols. This composite symbol will be used by default to represent the StationNodes elements. When defining the graphic effects related to the six discrete filters you create, you will have interesting alternatives for representing the filters by specifying subsymbol effects. The following table lists the graphic effects to specify for each discrete filter:

Type value	Filter graphic effects
A	The first subsymbol will be invisible.
B	The second subsymbol will be invisible.
C	The first subsymbol will be invisible; the second one will be rotated 90 degrees, and its fill color will be changed.
D	The two subsymbols will be invisible. The fill color of the basic main symbol will be changed.
E	The two subsymbols will both be rotated 45 degrees.
F	A new CGM symbol will be used to represent this type of station.

Begin by creating the composite CGM symbol. As the Stations element type is represented by the default node symbol, specify this new composed symbol for representing Stations schematic nodes.

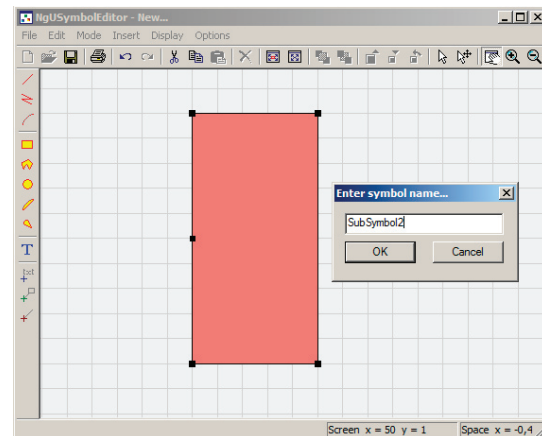
1. Expand the Element Types entry in the Designer tree and click Stations.
2. Click the Stations General tab, click the Symbol Name field, and click the NgUSymbolEditor button  to open the CGM Symbol Editor and Vector Drawing tool.
3. Click File and click New to creating the first subsymbol.
4. Click the Insert Rectangle button and draw a green rectangle as in the following screen shot (width: 5 units, height: 10 units).

Note: To change the rectangle fill color, double-click the newly created rectangle to open the Graphic Attributes dialog box, click the Fill Attributes tab, then change the Color parameter. Click File, click Save, type a name for the first subsymbol (“SubSymbol1”, for example), then click OK.




Next, create the second subsymbol.

5. Click File and click New. Click the Insert Rectangle button and draw an orange rectangle as in the following screen shot. Save this new symbol (“SubSymbol2”, for example).



Next, you will create the composite CGM symbol. This symbol will be based on the FCircle CGM symbol.

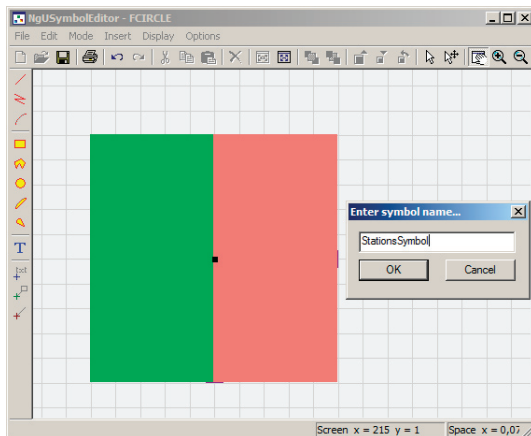
6. Click the Symbol Name parameter on the Stations General tab and click the Symbol Editor button . The NgUSymbolEditor opens.
7. Click File and click Open.
8. The Symbol Browser dialog box opens. Select the FCircle CGM symbol (the third symbol in the list).
9. Click OK. The Symbol Browser dialog box closes.
10. Double-click the circle to open the Graphic Attributes dialog box and change the circle's fill color (choose purple, for example). Click OK.

You will now insert the first subsymbol you have created.

11. Click Insert, point to SubSymbol, then click Internal SubSymbol.
12. Click SubSymbol1 on the Symbol Browser dialog box.
13. Click OK.

The SubSymbol1 is inserted.

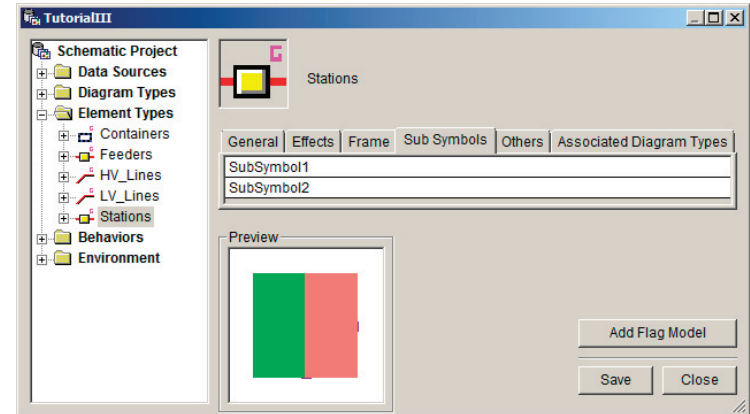
14. Repeat steps 10 through 12 to insert the second subsymbol.
15. Click File, click Save As, then type a name for this new CGM symbol. For example, type "StationsSymbol".



16. Exit the NgUSymbolEditor drawing tool.

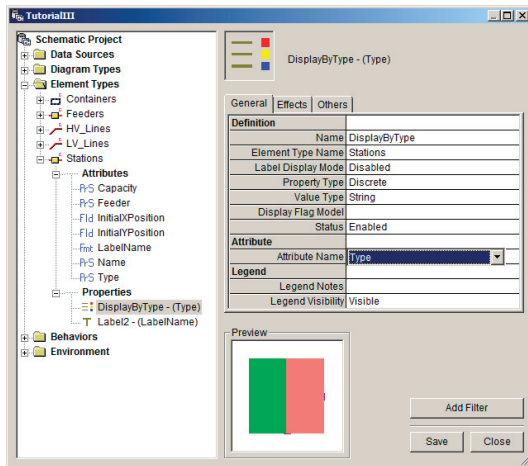
The newly composed symbol is automatically used as the Stations default symbol.

17. Click the Stations element type Effects tab and remove the fill color specified in exercise 2.
18. Click the Sub Symbols tab, which references the subsymbols that comprise this symbol.



Now, you must create the DisplayByType discrete property that will be used to display the Stations schematic nodes according to the Type propertyset attribute.

19. Right-click Properties below the Stations element type and click Create. The Create Property dialog box opens.
20. Type "DisplayByType" in the Property Name text box, check the Discrete option, keep String for the Value Type parameter, then click OK. The Create Property dialog box closes.
21. You must now associate the property with the Type PropertySet attribute. Click Type in the Attribute Name drop-down list on the DisplayByType General tab.



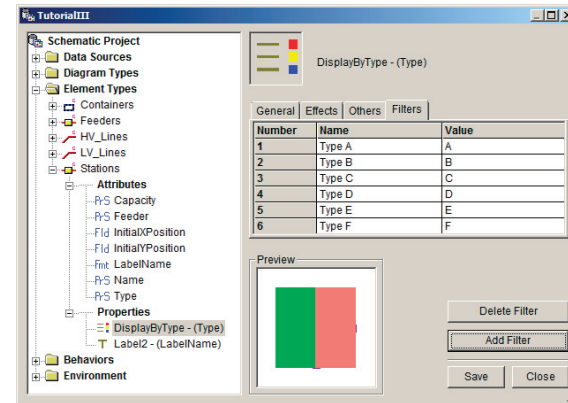
Next, create the discrete filters related to each Type field's value.

22. Click the Add Filter button on the lower right corner of the Designer Editor window. The Add Discrete Filter dialog box opens.
23. Type a name for the discrete filter ("Type A", for example) and specify the first value for the discrete filter ("A" in this example). This value is returned by the Type PropertySet attribute.
24. Click OK.

The Filters tab containing the first created discrete filter appears for the DisplayByType property.

25. Repeat steps 22 through 24 to create five new discrete filters for the B, C, D, E, and F values of the Type field database.

The Filters tab now appears as follows:



Define the graphic effects that will represent each discrete filter.

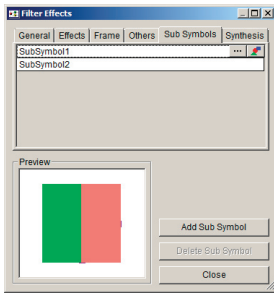
To define graphic effects related to a filter, click the Filters tab and choose the desired discrete filter parameter. Then, by clicking the ellipsis button displayed at the end of this parameter line, the Filter Effects form automatically opens.

In this case, as the station symbol comprises subsymbols, a Sub Symbols tab containing all subsymbols that compose the StationNodes symbol is available on the Filter Effects dialog box.

As the filter effects you need to specify will work on subsymbols, the Sub Symbols tab will be the focus of this exercise.

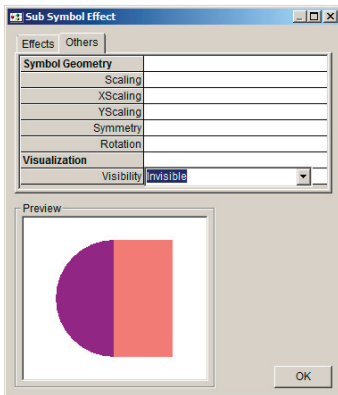
Start with the Type A filter to set the first subsymbol as invisible.

26. Click the Type A discrete filter on the Filters tab and click the ellipsis button to open the Filter Effects dialog box.
27. As the station symbol comprises subsymbols, a Sub Symbols tab containing all subsymbols that compose the Stations symbol is available on the Filter Effects dialog box. Click the Sub Symbol tab.
28. Double-click the line corresponding to SubSymbol1 and click the ellipsis button.



The Sub Symbol Effect dialog box related to SubSymbol1 opens.

29. Click the Others tab and click Invisible in the Visibility drop-down list .

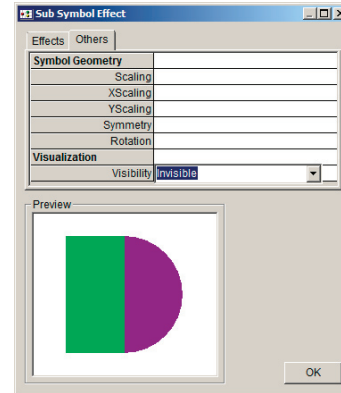


30. Click OK. The Sub Symbol Effect dialog box closes.

31. Close the Filter Effects dialog box.

Next, specify the Type B filter effects to set the second subsymbol as invisible.

32. Click the Type B discrete filter on the Filters tab and click the ellipsis button.
33. Click the Sub Symbols tab on the Filter Effects dialog box
34. Double-click the line corresponding to SubSymbol2 and click the ellipsis button.
35. Click the Others tab and click Invisible in the Visibility drop-down list.

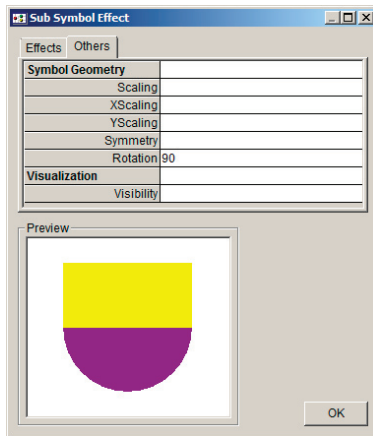


36. Click OK to close the Sub Symbol Effect dialog box.

37. Close the Filter Effects dialog box.

For the Type C discrete filter graphic effects, the first subsymbol must be invisible, and the second subsymbol will be rotated 90 degrees with a different fill color.

38. Open the Filter Effects dialog box corresponding to the third discrete filter (Type C).
39. Click the Sub Symbols tab and open the Sub Symbol Effect dialog box related to SubSymbol1.
40. Click the Others tab, click invisible in the Visibility drop-down list, then click OK so the subsymbol effect related to the first symbol closes.
41. Click the Sub Symbols tab and open the Sub Symbol Effect dialog box related to SubSymbol2.
42. Click the Others tab and type “90” in the Rotation field.
43. Click the Effects tab. Click the Fill Color field and set the color you want.



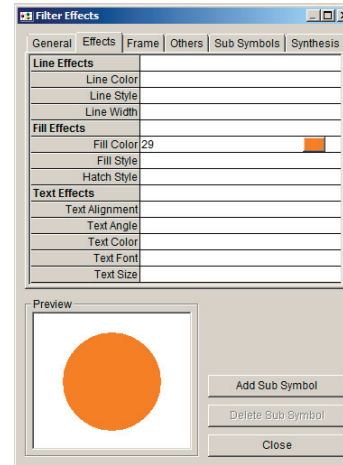
44. Click OK to close the Sub Symbol Effect dialog box.

45. Close the Filter Effects dialog box.

Next, define the filter effects related to the Type D discrete filter. For this filter, you need to set the two subsymbols to invisible and change the color for the basic symbol.

46. Open the Filter Effects dialog box corresponding to the fourth discrete filter (Type D).
47. Click the Sub Symbols tab and open the Sub Symbol Effect dialog box related to SubSymbol1.
48. Click the Others tab, click invisible in the Visibility drop-down list, then click OK so the subsymbol effect related to the first symbol closes.
49. Repeat steps 47 and 48 to deactivate the visibility for SubSymbol2.
50. Click the Effects tab on the Filter Effects dialog box. Click the Fill Color parameter, then set the color you want.

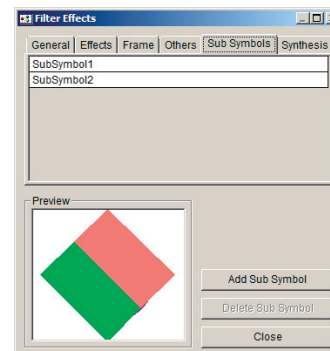
The Preview subwindow on the Filter Effects dialog box now appears similar to the following screen shot:



51. Close the Filter Effects dialog box.

For the Type E discrete filter, the subsymbols must both be rotated 45 degrees:

52. Open the Filter Effects dialog box corresponding to the fifth discrete filter (Type E).
53. Click the Sub Symbols tab and open the Sub Symbol Effect dialog box related to SubSymbol1.
54. Type "45" for the Rotation parameter on the Others tab, and click OK so the subsymbol effect related to the first symbol closes.
55. Repeat steps 53 and 54 for SubSymbol2 to rotate this subsymbol in the same way. The Preview subwindow on the Filter Effects dialog box now appears similar to the following screen shot:



Creating bounded properties for Stations

In this part, you will learn about bounded properties.

A bounded property allows you to specify the graphic effects matching different value ranges defined for this property. The value ranges defined for the property are generally based on an attribute that must be associated with the property. You will animate the station display according to the Capacity propertyset property values. Bounded property and display rules will be defined for each bounded filter, so the size of the symbol used to represent stations varies according to the Capacity field values.

The Capacity PropertySet property stored in the schematic table returns the capacity of each station as an integer value.

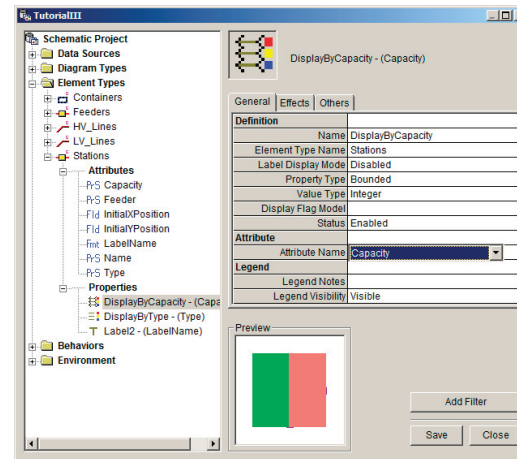
In this section, you will need to represent this information graphically. You will create the DisplayByCapacity bounded property composed of four bounded filters with each filter corresponding to one range of values. This property will display the nodes according to the Capacity field values as follows:

Capacity lower value	Capacity upper value	Filter graphic effects
0	2	Scaling factor: 1.0
3	5	Scaling factor: 2.0
6	8	Scaling factor: 3.0
9	10	Scaling factor: 4.0

1. Right-click Properties under Stations in the Designer tree and click Create. The Create Property dialog box opens
2. Type “DisplayByCapacity” in the Name text box.
3. Check the Bounded option.
4. Click Integer in the Value Type drop-down list.
5. Click OK.

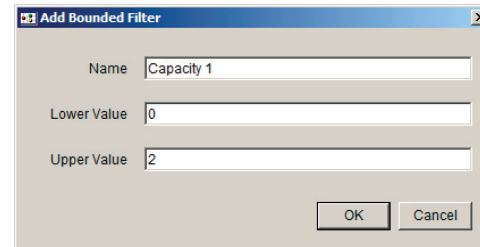
The new DisplayByCapacity property is automatically referenced below the Properties entry.

6. Click the General tab, and associate the property with the Capacity PropertySet attribute by clicking Capacity in the Attribute Name drop-down list.



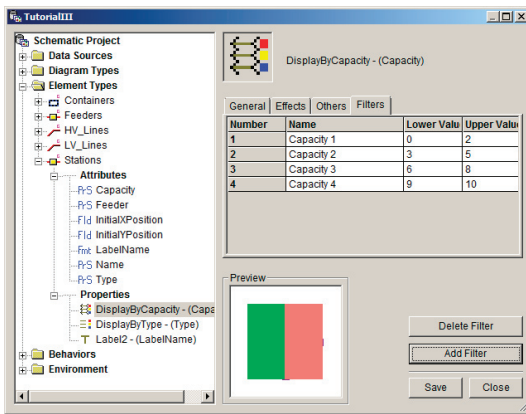
Next, you need to create the bounded filters.

7. Click Add Filter in the lower right corner of the Designer Editor window. The Add Bounded Filter dialog box opens.
8. Type a name for the first bounded filter (“Capacity 1”, for example) in the Name text box and type the lower and upper values (“0” and “2” in this exercise) that will define this first bounded range.



9. Click OK.
The Add Bounded Filter dialog box closes.
10. Repeat steps 7 through 9 to create three new filters corresponding to the following:
 - The 3 to 5 value range (Capacity 2)
 - The 6 to 8 value range (Capacity 3)
 - The 9 to 10 value range (Capacity 4)

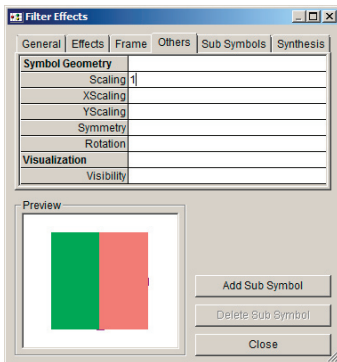
The Filters tab now appears like the following graphic:



Next, you will define the graphic effects that will represent each bounded filter.

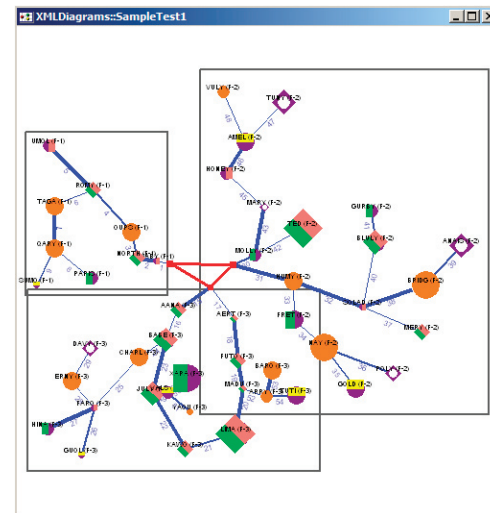
Start with the Capacity 1 bounded filter to set the station scaling factor to 1.

11. Click the Filters tab, click the Capacity 1 bounded filter, then click the ellipsis button.
12. Click the Others tab in the Filter Effects dialog box and type “1” for the Scaling parameter.



13. Click Close to close the Filter Effects dialog box.
14. To define the graphic effects related to Capacity 2, 3, and 4, repeat steps 14 through 16 and set the Scaling parameter with values of “2”, “3”, and “5”, respectively.
15. Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.

If you want to test the impact of the DisplayByCapacity property on the schematic diagram, click Close, switch to run mode, and open the diagram. It should appear as follows:

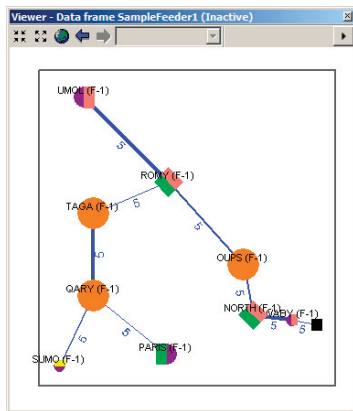


Exercise 4: Generating new XML diagrams and updating them

Generating new XML diagrams

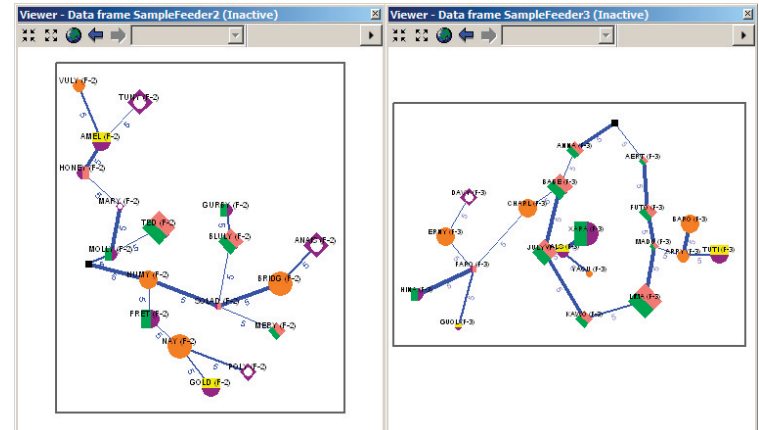
In this exercise, you will check this parameterization on new diagrams generated from other XML documents.

1. Exit Schematics Designer.
2. Start ArcMap and open the TutorialIII.mxd file you created in exercise 2.
3. Click Generate New Schematic Diagram.
4. browse and select the TutorialIII schematic dataset for the Schematic Dataset or Folder parameter.
5. For the Specify the input XML document parameter, browse and select the SampleNetworkFeeder1 XML document in your \ArcGIS\ArcTutor\Schematics\Schematics folder.
6. Type a name for the diagram that will be generated (for example, "SampleFeeder1") in the Schematic Diagram Name text box.
7. Click OK. The new generated diagram appears as follows:



The last specified parameters are taken into account for any new generated schematic diagram.

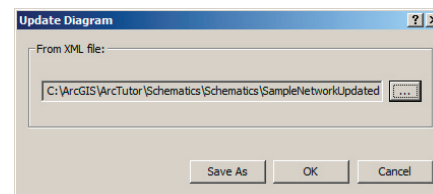
8. Repeat steps 3 through 7 to generate other XML diagrams based on the SampleNetworkFeeder2 and SampleNetworkFeeder3 XML files.



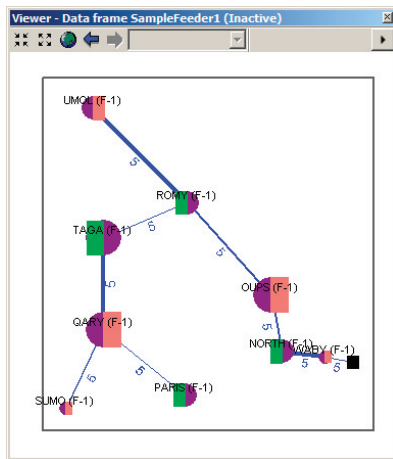
Updating XML diagrams

Updating an XML diagram also works from an input XML document; this XML document contains the updated version of the schematic elements contained in the diagram. The following steps describe how to update a diagram.

1. Remove the data frames related to the two last generated XML diagrams based on the SampleNetworkFeeder2 and SampleNetworkFeeder3 XML documents.
2. Make sure the SampleFeeder1 schematic diagram is set as the active diagram in the Active Diagram drop down list.
3. Click Schematic and click Update Diagram. The Update Diagram dialog box opens.
4. Click the ellipsis button, browse and select the SampleNetworkUpdatedFeeder1.xml stored in ArcGIS\ArcTutor\Schematics\Schematics folder and click Add. The Update Diagram dialog box appears as follows:



5. Click OK.
When the update process ends, the SampleFeeder1 diagram should appear as follows:



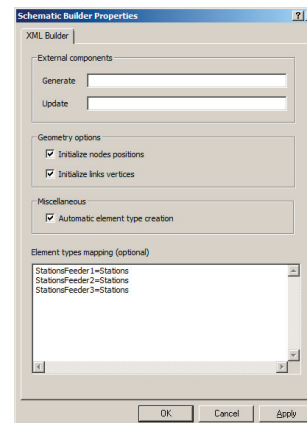
The SampleNetworkUpdatedFeeder1.xml document contains XML elements very similar to those contained in the SampleNetworkFeeder1 XML document initially used to generate the SampleFeeder1 diagram. Changes only impact the type of the Stations. In the initial XML document used to generate the diagram, there are Stations nodes for which the Type values are A, B, C, D, and E. In the updated XML document version, there are only Stations for which the Type value is A or B.

Set an end point to your schematic dataset definition

In the next paragraphs, you will learn about how to finalize your schematic dataset parameterization by activating the display of vertices along schematic links and deactivating the Automatic element type creation option.

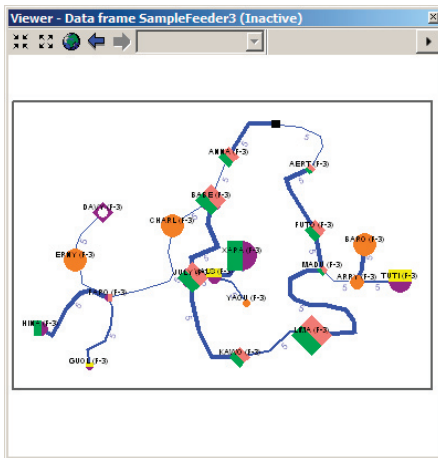
Until now, even if there are Vertices items among XML data related to link elements, this information has not been used in the generated diagrams because the XML Builder is not specified to manage them and pull them into the schematic database. To have the vertices managed by the builder and displayed along your links in the generated schematic diagrams, you need to activate the Initialize link vertices option from the Schematic Builder properties dialog box. The steps below detail how to proceed.

1. Start ArcCatalog and right-click the TutorialIII schematic dataset in the TutorialIII_SchematicDatabase.
2. Click Edit Project within Designer.
3. Click XMLDiagrams in the Designer tree.
4. Click the General tab, click the Schematic Builder parameter, then click the ellipsis button to edit the Schematic builder properties.
5. The Schematic Builders Properties dialog opens. Check the Initialize link vertices option.



6. Click OK.

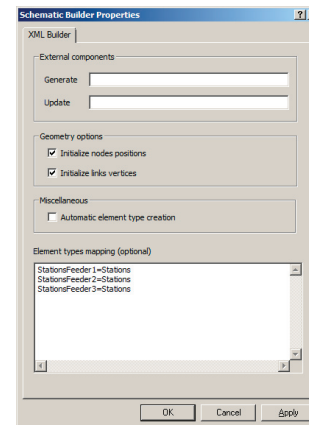
- You are now going to test this modification in your diagrams. For the vertices contained in the input XML files to be displayed along your schematic links, an update of the diagrams is required.
9. Start ArcMap and open the TutorialIII.mxd file.
 10. Click Open Schematic Diagrams, select the SampleFeeder3 diagram, then click Add.
 11. Click Schematic and click Update Diagram. The Update Diagram dialog box opens.
 12. Click OK to start the update process.
- At the end of the update operation, the diagram appears with vertices along their links.



13. Repeat steps 11 and 12 to update the other generated diagrams and to have vertices along the schematic links that they contain.

To set an end point to your schematic dataset definition, uncheck the Automatic element type creation option on the Schematic Builder Properties dialog box to be sure that no new parameter is added during the next diagram generations.

14. Close ArcMap without saving the TutorialIII.mxd file.
15. Edit the schematic dataset within Designer.
16. Click the XMLDiagrams entry in the Designer tree.
17. Click the Schematic Builder parameter on the General tab and click the ellipse button to edit the Schematic Builder Properties. The Schematic Builders Properties dialog box opens.
18. Uncheck the Automatic element type creation option.



19. Click OK.
20. Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.
21. Exit Schematics Designer.

Appendix: More about the Schematics XML Builder and its specifics

In this appendix, you will learn how an external component that builds XML data can be developed to automatically manage your XML diagram generation and update. You will also learn more about some specific points when using the XML builder: the associations with features and the use of dynamic attributes.

Specifying an external component to automatically build input XML data for XML diagrams generation/update

The XML documents used to generate XML Builder diagrams are probably not built manually. Generally, these documents are built by an external component that extracts all the required information from a database or several databases and pulls the data into an XML document. If a class of this component implements the `ISchematicXmlGenerate` interface available from the `SchematicUI` library, it is easy to specify this class so it is automatically launched with each request for an XML Builder diagram generation. In the same way, if the same class (or another class) of this component (or another component) implements the `ISchematicXmlUpdate` interface of the `SchematicUI` library, it can be specified to be automatically launched each time you request a XML Builder diagram update.

If you install the ArcGIS Desktop .NET Software Development Kit (SDK), you will get a sample of such a component code. The setup installs this sample project code in the `SchematicXMLBuilderComponent` subfolder on `ArcGIS\DeveloperKit\Samples\NET\Desktop\`. Two solutions are installed, a VBNet version and a CSharp project version. In both cases, the sample solution generates the `MyExtXMLComponent` Dynamic Link Library (DLL). The project code is composed of a class, `XMLDocImpl`, that implements the `ISchematicXmlGenerate` and `ISchematicXmlUpdate` interfaces. The program works from a set of features selected in a map document that comes from the `ElectricDataSet` feature dataset stored in the `Schematics_Designer_Tutorial.gdb` file. It allows you to automatically generate a XML document in memory with all the XML items associated with each selected feature. This XML document can then be used to generate/update XML diagrams.

The next steps are based on this sample component. To execute them, the following is required.

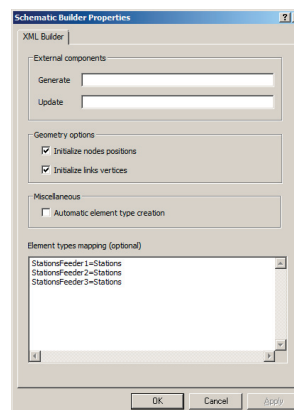
1. Install the ArcGIS Desktop .NET Software Development Kit.
2. Open one of the solution files available in the CSharp or VBNet directories installed on `ArcGIS\DeveloperKit\Samples\NET\Desktop\SchematicXMLBuilderComponent` folder within the Visual Studio 2005 environment.
3. Generate the solution. The `MyExtXMLComponent` DLL is generated.

This sample component has been used to generate the sample XML documents you have used during this tutorial.

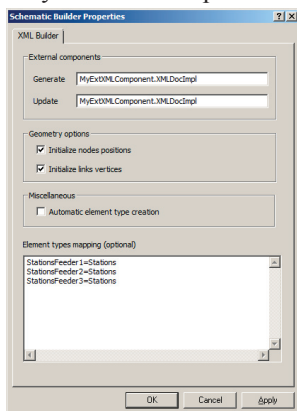
The next steps detail the modifications required for your TutorialIII schematic project so that this component is automatically launched each time you want to generate or update a diagram without browsing and selecting physical XML documents on your disk.

1. Start ArcCatalog.
2. Browse to the geodatabase that contains the TutorialIII schematic dataset, right-click this schematic dataset, then click `Edit Project`.
3. Click `XMLDiagrams` in the Designer tree.
4. Click the `General` tab, click the `Schematic Builder` parameter, then click the ellipsis button to open the `Schematic Builder Properties` dialog box.

The dialog box appears as follows:



5. Type the name of the applicative class that implements the ISchematicXMLGenerate interface in the Generate text box; that is, in that example, type “MyExtXMLComponent.XMLDocImpl”.
6. Type the name of the applicative class that implements the ISchematicXMLUpdate interface in the Update text box; that is, in that case, because the same class implements both interfaces, type “MyExtXMLComponent.XMLDocImpl” again.



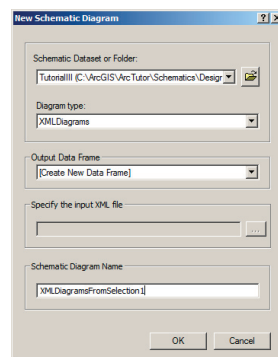
7. Click Apply and click OK.
8. Click Save. If the Schematic Project Verification Checklist dialog box opens, click Yes.

You can now test these modifications in ArcMap.

9. Exit Schematics Designer.
10. Start ArcMap and open the TutorialIII.mxd you created and saved at the end of exercise 2.
11. Click Selection and click Options to open the Selection Options dialog box.
12. Check the Select features completely within the box or graphic(s) option and click OK.
13. Save the .mxd file.
14. Click the Select Features tool on the Tools toolbar and select a set of features in the map.
15. Click Generate New Schematic Diagram.
The New Schematic Diagram dialog box opens.

- For the Schematic Dataset or Folder parameter, browse and select the TutorialIII schematic dataset in the TutorialIII_SchematicDatabase database you created in exercise 1.
- The XMLDiagrams type (the only diagram type in this schematic dataset) should be specified in the Diagram type text box.
- The ellipsis button at the right of the Specify the input XML file parameter appears dimmed. This is because the Generate parameter that implements the diagram type is specified on the XML Builder tab of the Schematic Builder Properties dialog box. In this case, no XML document must be specified for the diagram generation because the component automatically builds and returns XML data required for the diagram generation.

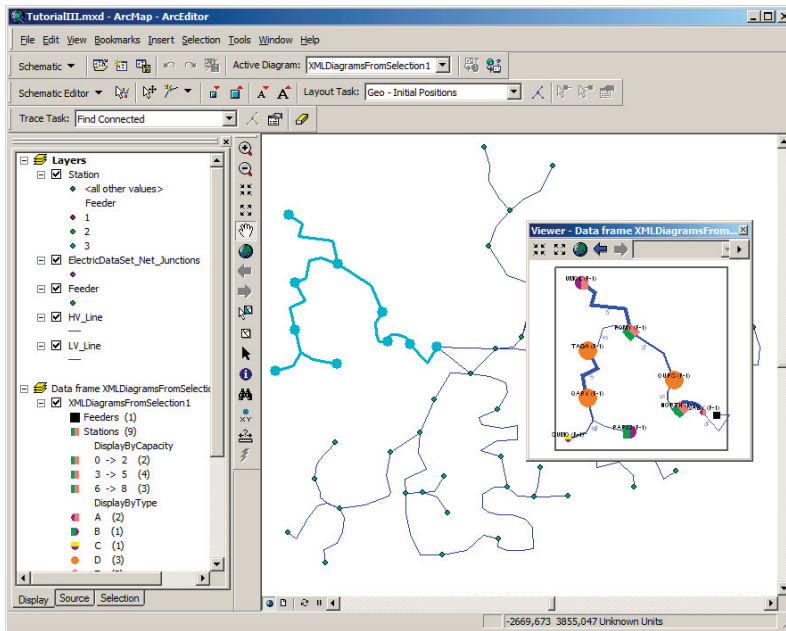
16. Type a name for the diagram that will be generated in the Schematic Diagram Name text box.
For example, type “XMLDiagramsFromSelection1”.



17. Click OK.

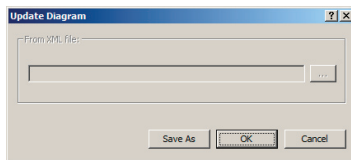
The Generating Diagram message box appears as follows with information about the state of each diagram generation phase:

- Phase 1: Executing external component.
During this phase, the external component operates and builds XML data.
- Phase 2: Processing XML elements
During this second phase, the XML data are read and pulled into the schematic tables.
Then the diagram should display in a viewer window.



Next, update your schematic diagram from a new selection set of features:

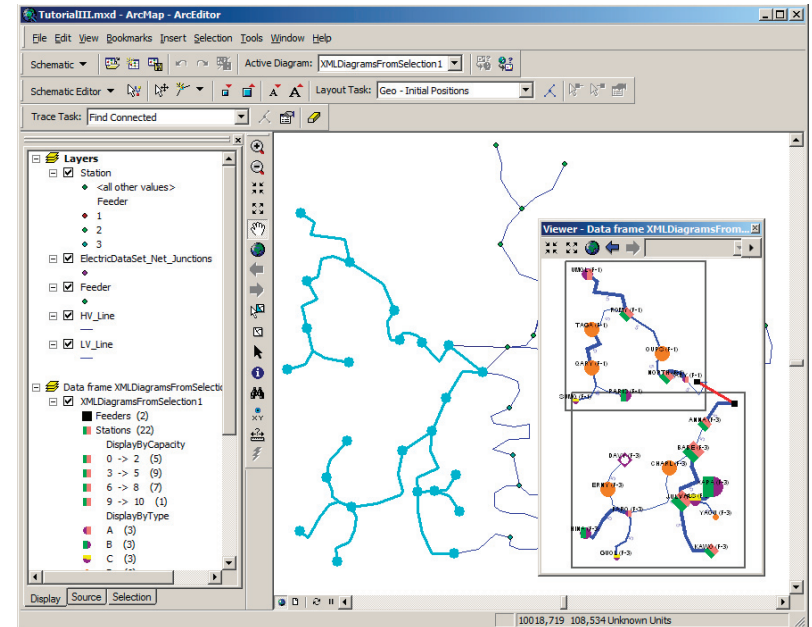
18. Click the Select Features tool and select another set of features in your map.
19. Click Schematic and click Update Diagram.
The Update Diagram dialog box opens:



The ellipsis button at the right of the From XML file parameter appears dimmed. This is because the Update parameter that implements the diagram type is specified on the XML Builder tab of the Schematic Builder Properties dialog box. In this case, no XML document must be specified for the diagram generation because the component automatically builds and returns the XML data required for the update operation.

20. Click OK.

The update process starts. At the end, the updated diagram appears. Click the Full Extent command on the schematic diagram viewer window to center the whole diagram content.



Associating features with schematic elements managed by the XML Builder

When elements contained in input XML documents come from features stored in a geodatabase, there are two ways to associate them with these features: by specifying particular items for each element in the input XML document or by creating predefined user data at the element type level in the schematic project.

Specifying the DataSourceName, UCID, and UOID for elements in the XML document

For any NodeElement, LinkElement, DrawingElement, NodeOnLinkElement, or SubLinkElement in the XML document, the DataSourceName, UCID, and UOID items can be specified.

- The DataSourceName must specify the name of the datasource to be used to connect to the geodatabase that contains the associated feature.
- The UCID gives the feature class ID of that associated feature.
- The UOID provides the associated feature ObjectID.

Note: The data source name must be previously declared in the <Datasources> section at the beginning of the XML document.

Following is a sample NodeElement where these items are specified:

```
<NodeElement>
  <ElementTypeName>StationsFeeder1</ElementTypeName>
  <ExternalUniqueID>Station-2</ExternalUniqueID>
  <DataSourceName>XMLDataSource</DataSourceName>
  <UCID>1</UCID>
  <UOID>2</UOID>
  ...
</NodeElement>
```

When these items are found for an element in the input XML document, the XML Builder pulls this data into the schematic table during the diagram generation/update, and the related schematic element in the diagram is automatically associated with its feature. When using the Identify tool on a schematic element, you will not only get schematic attributes related to the clicked element but also the attributes brought by the associated feature. Using the Propagate Schematic Selection To Map and Propagate Map Selection To Schematic commands will also work when the associated feature is displayed in a data frame.

Note: If the Associated Feature Class field on the General tab of the element type is specified in the schematic project, Schematics may also be able to retrieve the associated feature class even if the DataSourceName and/or UCID is missing in the input XML data. But the UOID in the input XML data is needed in all cases.

Creating predefined FDSN, FCN, and FOID user data at the element type level

Schematics will also try to manage the associations between features or objects contained in a geodatabase and schematic elements contained in XML Builder diagrams when three predefined user data values are specified for the element type that implements these schematic elements—FOID, FCN, and FDSN.

- The FOID user data value must specify the ObjectID of the feature or object you want to be associated with the schematic element.
- The FCN user data value indicates the feature class name or table name of the feature/object you want to be associated with the schematic element.
- The FDSN user data value specifies the name of the data source used to connect the geodatabase that contains this feature or object (this data source must use an ESRI connection—ESRI Access GeoDataBase (GDB), File GDB, ArcSDE GDB, and so on).

For elements contained in XML Builder diagrams, the FOID and FCN user data are required so the association is correctly managed. The FDSN is also often required unless the data source specified for the element type query connects the geodatabase containing the associated feature.

Note: To learn more about user data definitions, see the User Data help book in the ArcGIS Schematics Designer Help Online system.

More about attribute types on element types associated with the XMLBuilder diagram type

PropertySet attributes

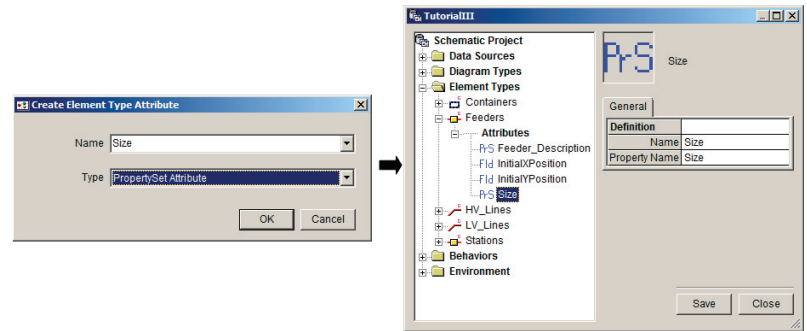
Because information you want to be available for your schematic elements will be mainly stored in your input XML document in the PropertySet sections, the PropertySet attributes are mainly those you will specify when using the XML Builder.

When you are generating XML diagrams while the Automatic element type creation option is checked, if a new PropertySet property is detected on an XML element for which the related element type has no PropertySet attribute associated with that PropertySet property, that PropertySet attribute is automatically created.

When the Automatic element type creation option is unchecked, the Key and Value of the PropertySet properties found in the XML document during the generation are stored in the schematic database even if no PropertySet attributes exist for the element type. But to get these values from the schematic tables and work with them, you will need to manually create a PropertySet attribute at the element type level. At its creation, the PropertySet attribute name must correspond exactly to the Key value of the PropertySet property found in the XML document.

For example, a new PropertySet property which Key value is Size is now available for Feeders NodeElement in the input XML document. To be able to work with this new information, you will need to create a new PropertySet attribute at the Feeders element type level in your schematic project as follows:

1. Edit your schematic dataset within Designer
2. Right-click the Feeders element type entry in the Designer tree and click Create Attribute. The Create Element Type Attribute dialog box opens.
3. Type “Size” in the Name text box.
4. Click PropertySet Attribute in the Type drop-down list
5. Click OK.

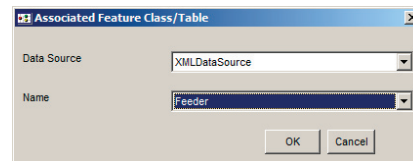


Dynamic attributes

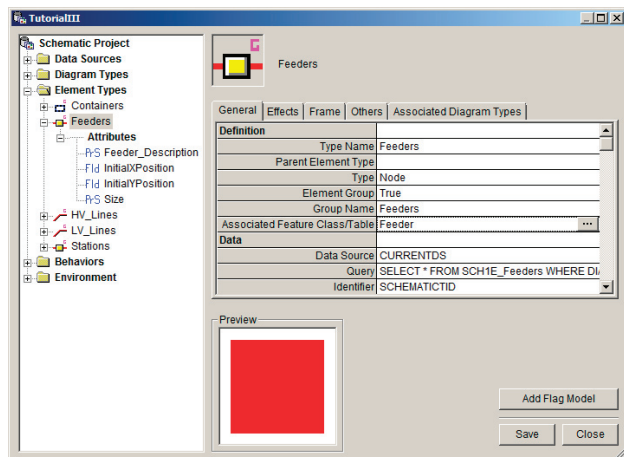
When your schematic elements can be associated with features, using Dynamic attributes can also be interesting.

Dynamic values are directly retrieved from a field in the associated feature class. To define such an attribute, even if the schematic element/feature association is defined from the DataSourceName, UCID and UUID items in the XML document, you need to specify the associated feature class at the element type level as described in the following steps:

1. Edit your schematic dataset within Designer.
2. Click the element type entry in the Designer tree for which you want to create a dynamic attribute.
3. Click the General tab and click the Associated Feature Class/Table parameter.
4. Click the ellipsis button that appears. The Associated Feature Class/Table dialog box opens.
5. Click the Data Source drop-down arrow and click the DataSource item that must be used to connect the database that contains the feature class you want to be associated with your element type.
6. Click the Name drop-down arrow and click the feature class.

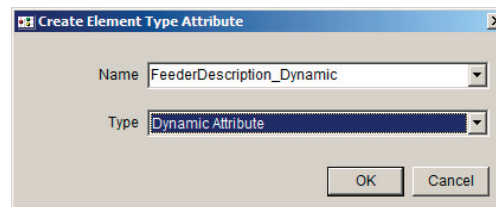


7. Click OK.
The specified feature class name appears as the Associated Feature Class/Table parameter.

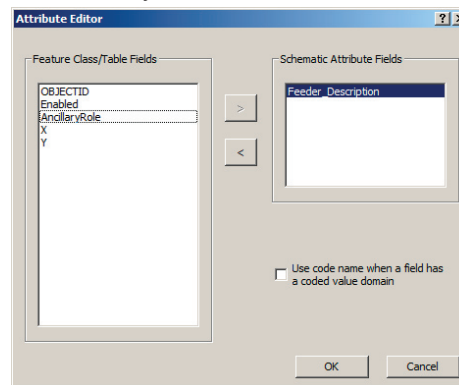


You can now create a Dynamic attribute related to any field contained in the associated feature class. For example, for the Feeder element type, the Feeder_Description PropertySet attribute already exists. Its values come from the Feeder_Description PropertySet property in the input XML document after they have been extracted from the Feeder_Description field of the Feeder feature class in the Schematics_Designer_Tutorial geodatabase. Suppose now that this PropertySet property is not available in the input XML document but you want to use it; one solution can be to create a Dynamic attribute directly associated with the feature class field.

8. Right-click Feeder element type in the Designer tree and click Create Attribute. The Create Element Type Attribute dialog box opens.
9. Type a name for your new attribute in the Name text box.
10. Select Dynamic Attribute in the Type drop-down list.
Note: This type of attribute was not available in the Type drop-down list before you specified the Associated Feature Class/Table parameter.



11. Click OK.
12. The Attribute Editor dialog box opens. Double-click the Feeder_Description field in the Feature Class/Table Fields list so it automatically moves into the Schematic Attribute Fields list.



12. Click OK.

Now, using the Feeder_Description PropertySet attribute or this newly created Dynamic attribute should be the same, except in some particular cases such as the following:

- The XML diagram has been generated and the Feeder_Description field values in the feature class have changed. In this case, the Dynamic attribute that is directly linked to the field will return the new values, while those coming from the schematic table will be the older values. It's better to update the diagram from an XML document that includes these changes.
- When working with ArcSDE, the XML diagram has been generated from a specific version, and the Associated Feature Class parameter specifies another version of the database for the associated feature class. In this case, the Feeder_Description field associated with the newly created Dynamic attribute will return values coming from that other version.

Static attribute

For the Standard Builder and Network Dataset Builder, which directly operate from a feature set during diagram generations, Static and Dynamic attributes are very near concept. A Dynamic attribute is directly linked to a feature class field, and its values can be retrieved from the associated features at any time. Static attribute values also come from values found in a feature class field, but that attribute is directly linked to a PropertySet property stored in the schematic table, this is a PropertySet attribute. These values are populated during the diagram generation and pulled into the schematic table when the input features are analyzed. For the XML Builder, even if the generated schematic elements are associated with features, the diagram generation never operates directly from those features; it only works from an input XML document. This explains why Static attributes may have no real meaning for that builder even if there is nothing to stop their creation within Designer. A Static attribute can be created as dynamic attribute at the element type level, but if its name doesn't correspond to a Key specified for one of the PropertySet properties defined on the input XML elements, this attribute will not return expected values. If a PropertySet property found in the input XML document has a Key that doesn't match any defined Static attribute name, that Static attribute will have no meaning.

Using the Import From Feature Layers command

If the schematic elements contained in your XML diagrams mainly come from feature/object classes, you probably have a map, map template, or layer file that contains the feature layers that comprise the features with which you want to work. The symbology used to represent the feature layers in your ArcMap project is often already defined, categories have been created, and you are familiar with the representation. In that case, when you start your schematic project conception, using the Import From Feature Layers command can be very helpful to quickly create the required element types, and import the symbology already defined in the map, map template, or layer file.

Please note that there are some points you need to verify first.

- If the names of the feature classes related to the feature layers don't match the names of the ElementTypeName found in the input XML, you will have to specify rules so they match in the Element types mapping section of the Schematic Builder Properties dialog box. If not, the XML diagrams will be empty at the end of the generation.
- The Static attributes that will be automatically created on the element types according to the categories specified for the imported layers (and the Schematics properties with which they are associated) will have meaning only if there are PropertySet properties in the input XML documents used to generate the XML diagrams that have Key values that exactly match those Static attribute names. You must change the attribute names so they match if needed.
- For the element types automatically created during the Import From Feature Layers process, the Associated Feature Class/Table field is populated in the schematic project. If the UUID is not specified in the input XML documents during the diagram generations, this parameter will have no effect.
If the UCID and/or DataSourceName are also specified in the input XML documents, those XML parameters will always be those used for the schematic element/feature associations.